

AD-A214 025

4

DTIC FILE COPY

**A Formal Approach to Planning With
Concurrent Actions and External Events**

Richard N. Pelavin

Technical Report 254
May 1988

DTIC
ELECTE
NOV 03 1989
S B D
Co

**UNIVERSITY OF
ROCHESTER
COMPUTER SCIENCE**

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

89 11 01 053

A FORMAL APPROACH TO PLANNING WITH
CONCURRENT ACTIONS AND EXTERNAL EVENTS

by

Richard N. Pelavin

Submitted in Partial Fulfillment
of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY

Supervised by James Allen

Department of Computer Science

University of Rochester
Rochester, New York

1988

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 254	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Formal Approach to Planning With Concurrent Actions and External Events		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard N. Pelavin		8. CONTRACT OR GRANT NUMBER(s) N00014-82-K-0193
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Department 734 Computer Studies Bldg. University of Rochester, Rochester, NY 14627		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS D. Adv. Res. Proj. Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE May 1988
		13. NUMBER OF PAGES 309
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Distribution of this document is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES none		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Planning, temporal logic, external events, concurrent interaction,		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) see reverse		

20. ABSTRACT

Planning was originally formulated in the state-based framework where actions are modeled as functions from instantaneous state to state. This framework provides a simple basis for describing the different ways the agent can affect the world, but is inadequate for describing or reasoning about planning problems that involve either concurrent actions or external events, i.e., events initiated by forces and agents other than the planning agent, that may occur while the planning agent is acting. In response to these deficiencies, Allen [1984] and McDermott [1982] put forth temporal logics that can describe simultaneous events. These formalisms, however, cannot capture the ways in which the planning agent can affect the world by executing different actions.)

This dissertation presents a deductive logic to describe and reason about planning problems that may involve concurrent actions and external events. A semantic theory and axiomatization are provided. We exploit the complementary strengths of the state-based model and those of Allen and McDermott by extending Allen's model with a structure similar to the result function found in the state-based model. This structure captures the result produced by executing different actions at specified times with respect to a context that includes external events that may be simultaneously occurring. This provides a framework for modeling concurrent interactions between the agent's actions and external events. It also provides a simple basis for composing actions, both concurrent and sequential, to form more complex ones (i.e., plans). (KR) ←

Using our framework, we analyze planning problems by showing how they can be expressed in our logic and casting planning as deduction in our logic. We pay particular attention to concurrent interactions and the relation between composite plan instances and their constituent parts. We also discuss the *STRIPS assumption* and its analog, the *persistence assumption* [McDermott 1982], which have been employed by most planning systems to determine which properties an action does affect. We demonstrate that these assumptions are inappropriate when planning with concurrent actions or external events, and propose an alternative approach.

Lastly, we develop a planning algorithm that is applicable in domains with concurrent actions and external events, which we prove is sound with respect to our semantic theory.

Curriculum Vitae

Richard Pelavin [REDACTED] He spent his childhood and teenage years in the Westchester NY area and attended high school in Chappaqua, NY. In 1977 he entered the State University of New York at Albany to pursue his interests in Math and Science. Richard graduated Summa Cum Laude from Albany in 1981. He received a B.S. degree with a major in both Computer Science and Economics and a minor in Mathematics. Richard also became a member of Phi Beta Kappa while at Albany.

Richard entered the Computer Science Department at the University of Rochester in 1981 to explore his interest in Artificial Intelligence. During his stay at Rochester, he was both a teaching and research assistant. He also worked for three summers at the IBM Research Center in Yorktown Heights, NY.

Soon after arriving at Rochester, Richard became interested and involved in the work of James Allen, who later became his advisor. Richard's primary interests lie in the area of knowledge representation and reasoning. His particular focus was in the area of AI planning.

Since the end of 1986, Richard has been employed in the Artificial Intelligence group at Philips Laboratories in Briarcliff, NY where he is pursuing his interests in AI research.



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Acknowledgments

I would like to thank my advisor, James Allen, for his time, guidance, and encouragement throughout my graduate career. I have benefited greatly from the interaction. By working with James, I have learned how to organize a research project, focus on relevant issues, and express myself clearly and precisely. I also thank the members of my committee, Henry Kyburg and Jerry Feldman, for their support and feedback.

I would also like to thank Philips Laboratories, and in particular, Dick Wexelblat, for their support and willingness to allow me to finish my dissertation amidst my other responsibilities.

Lastly, I would like to thank all the people I encountered in the Computer Science Department at Rochester, both staff and students, who made my stay both enjoyable and stimulating. Special thanks goes to Jay Weber and Josh Tenenberg for their help in coordinating things at Rochester while I was at Philips.

Abstract

Planning was originally formulated in the state-based framework where actions are modeled as functions from instantaneous state to state. This framework provides a simple basis for describing the different ways the agent can affect the world, but is inadequate for describing or reasoning about planning problems that involve either concurrent actions or external events, i.e., events initiated by forces and agents other than the planning agent, that may occur while the planning agent is acting. In response to these deficiencies, Allen [Allen 84] and McDermott [McDermott 82] put forth temporal logics that can describe simultaneous events. These formalisms, however, cannot capture the ways in which the planning agent can affect the world by executing different actions.

This dissertation presents a deductive logic to describe and reason about planning problems that may involve concurrent actions and external events. A semantic theory and axiomatization are provided. We exploit the complementary strengths of the state-based model and those of Allen and McDermott by extending Allen's model with a structure similar to the result function found in the state-based model. This structure captures the result produced by executing different actions at specified times with respect to a context that includes external events that may be simultaneously occurring. This provides a framework for modeling concurrent interactions between the agent's actions and external events. It also provides a simple basis for composing actions, both concurrent and sequential, to form more complex ones (i.e., plans).

Using our framework, we analyze planning problems by showing how they can be expressed in our logic and casting planning as deduction in our logic. We pay particular attention to concurrent interactions and the relation between composite plan instances and their constituent parts. We also discuss the *STRIPS assumption* and its analog, the *persistence assumption* [McDermott 82], which have been employed by most planning systems to determine which properties an action does affect. We demonstrate that these assumptions are inappropriate when planning with concurrent actions or external events and propose an alternative approach.

Lastly, we develop a planning algorithm that is applicable in domains with concurrent actions and external events, which we prove is sound with respect to our semantic theory.

Table of Contents

1. Introduction	1
1.1. The Problem	1
1.2. Deficiencies of the State-Based Planning Paradigm	3
1.3. Overview of the Thesis	7
2. Bringing in the Components of the Logic	11
2.1. Allen's Interval Logic	11
2.2. Plan Instances	14
2.3. The Deficiencies of Interval Logic in Connection with Planning	16
2.4. The Inevitability Modal Operator	21
2.5. What the Branching Time Structure Does Not Capture	28
2.6. The IFTRIED Modal Operator	35
3. The Formal Specification of the Logic	40
3.1. The Language	40
3.2. The Semantics	44
3.2.1. The Interval Logic Fragment	45
3.2.2. The Interpretation of INEV and the R Accessibility Relation	49
3.2.3. Plan Instances and Goldman's Theory of Actions	52
3.2.4. Basic Actions and the Interpretation of IFTRIED	57
3.2.5. The Composition of Basic Action Instances and Plan Instances	69
3.2.6. Comparison with Semantic Theories of Conditionals	84
4. A Proof Theory	88
4.1. Axiomatization of the First Order Connectives	88
4.2. Axioms Describing the Interval Logic Predicates and Terms	91
4.3. The Axiomatization of the INEV Modal Operator	96
4.3.1. S5 Properties	96
4.3.2. Temporal Properties	105
4.4. The Axiomatization of IFTRIED	114
4.4.1. IFTRIED as a Subjunctive Conditional	114
4.4.2. The Relation Between IFTRIED and INEV	120
4.4.3. The Relation Between IFTRIED and Plan Instance Composition	125
5. Analyzing the Planning Problem	130
5.1. The Planning Environment	132
5.2. Executability Conditions	134
5.3. Plan Instance Interactions	141
5.4. Plan Instance Effects	150
5.5. Persistence and Simultaneous Events	155

5.6. Maintaining a Property	164
6. A Planning Algorithm	171
6.1. Overview	171
6.2. Specification of the Planning Algorithm	174
6.3. Proving that the Algorithm is Sound	181
6.4. Planning Examples	192
6.4.1. The Two Planning Operators and Interval Logic Reasoning	192
6.4.2. Sequential Interactions and Maintenance	195
6.4.3. Concurrent Interactions	199
7. Conclusion	202
7.1. Summary	202
7.2. Limitations and Future Directions	204
7.2.1. Limitations of the Planning Algorithm	204
7.2.2. Issues Outside the Scope of the Deductive Logic	209
Bibliography	213
Appendix A The Primitive Language	217
Appendix B Defined Symbols	219
Appendix C The Semantic Model	221
Appendix D The Axiomatics	227
Appendix E Auxiliary Theorems and Derived Rules	232
Appendix F Proof of Theorems in Chapter 4	235
Appendix G Proof of Theorems in Chapter 5	246
Appendix H Proof that the Algorithm is Sound	261

List of Figures

2.1-4 A Branching Time Structure	21
2.4-2 Interpreting the POS and INEV Modal Operators	23
2.5-1 Possibilities Out of the Agent's Control	29
2.5-2 Enablement Relation Between Plan Instances	30
2.5-3 Relation Between Branching Time Structure and Executability Conditions that Hold During Execution	34
3.2-1 Two World-Histories Differing on the Account of the Execution of a Basic Action Instance	62
3.2-2 Minimal Revision Links Corresponding to a Game-Tree	65
3.2-3 Relation Between the R Relation and the Basic Action Functions	68
3.2-4 The Definition of and Two Constraints Relating to the Composition of Basic Action Instances	70
3.2-5 Interfering Basic Action Instances	73
3.2-6 Non-Interfering Basic Action Instances	74
3.2-7 Truly Simultaneous Plan Instances	77
4.1-1 Axioms and Inference Rules Relating to the First Order Logical Symbols	89
4.2-1 Axioms Relating to the Interval Logic Fragment	92
4.2-2 Soundness Proofs for Axioms AX-IL1 - AX-IL4	94
4.2-3 Soundness Proofs for Axioms AX-IL5 and AX-IL6	95
4.3-1 Axioms and Inference Rule Relating to the S5 Properties of INEV	97
4.3-2 Soundness Proofs for Axioms AX-INV1 and AX-INV2	98
4.3-3 Soundness Proofs for Axioms AX-INV3 and AX-INV4	99
4.3-4 Axioms Relating to the Temporal Properties of INEV	106
4.3-5 Soundness Proofs for Axioms AX-INV5 and AX-INV6	107
4.3-6 Soundness Proofs for Axioms AX-INV6 (cont.) and AX-INV8	108
4.3-7 Soundness Proofs for Axioms AX-INV8 (cont.) and AX-INV9	109
4.4-1 Axioms and Inference Rule Relating to IFTRIED as a Subjunctive Conditional	114
4.4-2 Soundness Proofs for Axioms AX-IFTR1 and AX-IFTR2	116
4.4-3 Soundness Proofs for Axioms AX-IFTR3 and AX-IFTR4	117
4.4-4 Axioms Relating to the Relation Between IFTRIED and INEV	120
4.4-5 Soundness Proof for Axiom AX-IFTR5	121
4.4-6 Soundness Proof for Axiom AX-IFTR6	122
4.4-7 Axioms Relating to Plan Instance Composition	125
4.4-8 Soundness Proof for Axiom AX-IFTR7	127
4.4-9 Soundness Proof for Axiom AX-IFTR8	128

4.4-10 Soundness Proof for Axiom AX-IFTR9	129
5.3-1 Pollack's Interpretations of the Header-Body-Precondition Specification of Actions	137
6.2-1 Definition of the REMOVE and INTRO Planning Operators	176
6.4-1 The Logical Statements Corresponding to the State of the Planning Algorithm	188
6.4-2 Constraints Imposed on the World Model by the Input Specifications	189
6.4-3 Logical Transformation Corresponding to REMOVE	190
6.4-4 Logical Transformation Corresponding to INTRO	191

Chapter 1

Introduction

1.1. The Problem

This thesis presents a formal model of action and time to provide a basis for planning in temporally rich domains. This includes domains where the planning agent may concurrently execute a set of actions in a world where other agents and external forces are simultaneously producing changes. The planning agent may need to interact with these **external events** in order to prevent some undesirable occurrence, insure the successful completion of some event, or to perform some action enabled by an external occurrence. By treating concurrency, we can model a robot agent that has multiple effector devices that can be operating simultaneously. This same model can also be used to represent plans that are to be jointly executed by a group of cooperating agents. In both cases, we refer to the agent or set of agent's doing the planning as the planning agent.

In domains where external events may produce changes in the world, the time when a plan is to be executed must be considered and reasoning about temporal constraints becomes necessary. Consider a simple planning problem. Currently, it is 10:30 PM and the agent is in the library. The agent's goal is to get its books out of the office before 11:00 PM. The office may be locked anytime between 10:45 and 11:00 PM. Two simple actions that the agent can perform are to walk from the library to the office, which can be done anytime and takes ten minutes, and to enter the office, which can only be done if the office door is not locked. If we also assume that the agent cannot unlock the door, then the agent must plan to leave from the library before 10:35 PM, getting to the office before it is locked.

In domains where events may occur simultaneously, there are a number of interactions involving simultaneous actions and events, both constructive and destructive, to consider. We investigate two types of constructive relations: the interaction between two actions that can be executed together but not separately, and the interaction involving an external event that enables the agent to perform some action. An example of the first type of interaction is where an object is lifted by applying pressure to two ends of an object, one hand at each end. If pressure were applied to only one end, the result would be a pushing action, not part of a lifting action. An example of the second type of interaction is where the action "sailing across the lake" can only successfully occur if the wind is blowing while the sailing is taking place. It is a parallel interaction since the wind must be blowing while the sailing takes place, not before or after.

Destructive interactions refer to situations where the occurrence of an event (or set of events) precludes another event (or set of events) from occurring. We will refer

to this interaction as interference. Interference may result from a *resource conflict*. For example, there may be only one burner working on a stove, and two dishes to be heated. Either dish can be heated at any time, but clearly, both dishes cannot be heated simultaneously. Thus, we must be able to represent interactions where there are two concurrent actions that can be done separately, but cannot be done together. We must also model resource conflicts between three or more actions where only a subset of them can be done together. Other examples of interference involve actions that are alternative choices, only one of which can be performed at one time. An example is where the agent can move forward at any time, move backward at any time, but cannot do both simultaneously.

Two concurrent actions may only conditionally interfere. As an example, the agent may be in the airport and wish to carry two bags onto the plane. Suppose that each passenger always can carry one bag onto the plane, and can carry additional bags only if the plane is not going to be full. To model this situation, we must represent that "carry on *bag1*" and "carry on *bag2*" can always be done individually, but cannot be done together if the plane is going to be full. We must also be able to represent that the condition "the plane is full" is out of the agent's control. This case must be distinguished from situations where two actions conditionally interfere, but the agent can bring about the condition under which they do not interfere. Consider a simple example. If the agent has more than seven dollars but less than fourteen dollars on hand, it can buy a record that costs seven dollars or buy some tape that costs seven dollars, but cannot do both. The agent, however, can purchase both items by bringing along at least fourteen dollars.

The examples above concern the interference between two actions that the planning agent can perform. Other types of interference involve the relation between an action initiated by the planning agent and an event caused by another agent or external forces. These types of interactions must be treated differently. If two of the agent's actions interfere and the agent wants to perform one of them, it can simply choose not to perform the other. On the other hand, if the planning agent's action interferes with an event caused by another agent or external forces, the planning agent may not be able to prevent this conflicting event from occurring. Consider a scenario where the planning agent and another agent share a terminal. There are a number of different interactions corresponding to different priority schemes. If the planning agent has priority and can kick the other agent off the terminal at anytime, then the planning agent can use the terminal at anytime. It can also prevent the other agent from using the terminal. Conversely, if the other agent has priority, the planning agent can be sure that it can use the terminal during some time period t only if it is sure that the other agent will not be using the terminal anytime during t . A third interaction involves a first come first serve priority scheme.

In the following chapters, we develop an interpreted logic (i.e., a logic with a semantic model) to represent planning problems having concurrent actions and external events. This framework provides the basis for analyzing and constructing planning systems that handle concurrent actions and external events. In the next section, we describe situation calculus, the context in which Artificial Intelligence

planning was originally formulated. This has given rise to the **state-based planning** paradigm, exemplified by such systems as STRIPS [Fikes&Nilsson 71] and NOAH [Sacerdoti 77]. Many of the issues that we consider are in response to the inadequacies of these systems. The main limitation of situation calculus is that it does not model simultaneous events. Consequently, the few state-based planners that treat concurrent actions or external events have done so without an appropriate underlying model. We illustrate that these planning systems are limited and produce unacceptable results if certain restrictions are not imposed. In order to relax these restrictions, a more general model of actions and events is needed.

1.2. Deficiencies of the State-Based Planning Paradigm

One of the most successful approaches for representing events and their effects in Artificial Intelligence has been situation calculus [McCarthy&Hayes 69]. In this framework an event is represented by a function that takes a **situation**, which is an instantaneous snapshot of the world, and yields the situation that would result from applying the event to its argument. Events can be combined to form sequences. The result of applying the sequence $e_1; e_2; \dots; e_n$ to situation s is recursively defined as the result of applying $e_2; \dots; e_n$ to the situation that results from applying e_1 to s .

Situation calculus has given rise to the state-based planning paradigm which, without extension, has the following form: given a set of sentences describing conditions that are initially true and a set of sentences describing goal conditions to be achieved, a sequence of actions must be found that when applied to any situation where the initial conditions hold yields a situation where the goal conditions hold. In this framework, the description of the world in which planning is done, which we refer to as the **planning environment**, only describes the initial situation, i.e., the situation that holds just prior to plan execution. As a result, this representation is only adequate for planning problems where all changes in the world result from the planning agent's actions. If this is the case, then conditions that are true in the initial situation will remain true until the agent performs an action that negates it. Consequently, the future is uniquely determined by the initial situation and the sequence of actions that the planning agent chooses to perform from this situation.

This simple type of planning environment does not provide for planning problems where the world may be affected by external events that occur while the agent is executing a plan. Examples of such conditions include:

- i) The bank is going to open at 9:00 and will close at 3:00
- ii) It will possibly start raining any time between 3:00 and 4:00
- iii) If the agent is outside without an umbrella while it is raining, the agent will get wet
- iv) It is possible that the can of paint sitting in the doorway will be knocked over if the agent does nothing about it

The type of goals considered in state-based planning are also very limited. Goals are just conditions that must hold at the completion of plan execution. In this research, a goal may be any temporally qualified proposition describing conditions in the future of planning time. Thus, a goal may be to avoid some condition while performing some task, prevent an undesirable condition that possibly will happen, or achieve a collection of conditions in some specified order. Examples of these type of goals are:

- i) Do not damage the tape heads while repairing the tape deck
- ii) Prevent Tom from entering the room
- iii) Get to the gas station on the way to driving to school (ordered goals)

More precisely, goals are temporally qualified propositions that partition the set of possible futures into the set of possible futures where the goal holds and the set of possible futures where the goal does not hold. In this research, we will not consider goals such as "finding the best way to achieve ..." which presupposes some utility measure or precedence ordering relating the possible futures. Whether a goal holds in some possible future is true or false and not dependent on other possible futures.

Finally, in the state-based approach, the types of actions and plans that can be represented are limited. Since actions are transformations from state to state, there is no conception of conditions holding or changing while an action takes place. This precludes the treatment of an action whose execution depends on conditions that hold during execution. For example, a necessary condition to edit a document on a text editor is that the text editor is operational while the editing is taking place. This relation cannot be described in situation calculus if we treat "editing a document on a text editor" as an action. Only conditions that hold prior to execution, which have been called *preconditions*, can be used as necessary conditions.

Plans are limited because they are sequences of actions to be executed in the initial situation. Thus, one cannot treat plans containing concurrent actions and plans that have an execution time that starts later than the initial situation. The first restriction comes about because it is impossible to represent concurrent actions in situation calculus. The second restriction does not matter in state-based planning problems since all changes in the world are due to the planning agent. Thus, it makes no difference when a plan is initiated; the world will not change until the plan is started.

Automated Planning Systems

Most domain independent planners are state-based planners, or limited extensions of the state-based approach, such as NOAH [Sacerdoti 77], NONLIN [Tate 77], DEVISER [Vere 1981], and SIPE [Wilkins 1983]. All these systems model actions as functions that transform one instantaneous state into another. The ancestor of these systems is STRIPS [Fikes&Nilsson 71].

The type of planning problems handled by STRIPS is exactly what we have described as the state-based planning paradigm. The major contribution made by STRIPS is the so called *STRIPS assumption* to handle the frame problem, that is, the problem of representing that an action only affects a small part of the world, leaving the rest of the world unchanged. Thus, if situation s_1 is the result of applying an action in situation s_0 , typically, s_0 and s_1 will be very much alike. In STRIPS, an action is defined as ordered triplet consisting of a precondition list, add list, and delete list, each member of these lists being atomic formulas. An action may be applied in any state s where all its preconditions hold yielding a new state that is computed from s by adding the formulas on the add list and deleting all the formulas on the delete list. Implicit in this treatment is that any formula that is not explicitly asserted to be true in a state is taken to be untrue. This enables one to avoid explicitly specifying negated atomic formulas. This approach, however, precludes the use of partial descriptions of states where the truth values of some formulas are unknown.

Also in the class of state-based planners are the non-linear hierarchical planners descending from NOAH [Sacerdoti 77]. Describing a planner as being hierarchical means that it solves goals by first considering an abstract plan and then considering more detail at successive stages. The term "non-linear" refers to a search strategy where the planner solves a conjunction of two sub-goals by looking at them separately (in isolation). The planner then tries to merge the two plans to avoid conflicts between actions in these plans. The fact that they are non-linear and hierarchical does not make them any more expressive than STRIPS. These terms refer only to the control strategy, rather than to the representation of plans. The planning environment in these systems are just as we described for the state-based planning paradigm. The plans returned by these systems are partially ordered set of actions that are to be linearized upon execution. The use of the STRIPS assumption is implicit in the method that is used to detect harmful interactions between two actions entered into a plan. If an action a_1 is detected whose deletion list mentions a precondition of another action a_2 , either a_2 is ordered after a_1 (if possible), or a third action that restores a_2 's precondition is inserted into the plan and ordered after a_1 and before a_2 . A concise presentation describing action interactions in non-linear planners is given by Chapman [Chapman 1985].

There have been a number of domain independent planning systems that have handled a larger class of planning problems than STRIPS. Wilkins' SIPE [Wilkins 83] and Vere's DEVISER [Vere 81] are two of the most sophisticated types, both of which are extensions of non-linear hierarchical planners such as NOAH [Sacerdoti 77] and NONLIN [Tate 77]. Wilkins' system handles plans with concurrent actions, and Vere's system treats future events that may occur while a plan is taking place and actions that have durations.

Parallel Actions in a State-Based System

In Wilkins' system, a plan is a partially ordered set of actions. Any two actions where one is not ordered before the other are considered to be in parallel branches.

Wilkins introduces *resources* to reason about the interaction of parallel actions. A resource is defined as an object that an action uses during its execution. If two actions share the same resource, they cannot be executed in parallel. Ordering constraints are imposed (if possible) to insure that any two actions sharing a resource cannot be in parallel branches.

Wilkins points out that although his notion of resources is very useful, they are still quite limited. They do not allow one to treat such things as money or computational power as resources. In general, resources cannot be used to encode actions that conditionally interact, such as the interaction we mentioned where the agent can carry two bags onto the plane only if the plane is not going to be full. This limited treatment of concurrent interactions can be attributed to the fact that his system is an extension of a state-based system. Situation calculus, the model for these systems, only represents sequential interactions. This refers to an enablement interaction where an earlier action brings about a later one's preconditions and a harmful interaction where an earlier action ruins a later one's preconditions. These interactions are captured by the precondition-add-delete lists provided for each action in a state-based system. These lists, however, do not describe the parallel interaction between actions. Without a model supporting parallelism, it is not clear what the potential types of parallel interactions are, and there is no clear guide as to what a correct implementation would be.

Another potential problem is the inappropriate use of the STRIPS assumption in a representation that models concurrency. Problems arise if the STRIPS assumption is used naively. Consider two actions $a1$ and $a2$ that are executed in parallel starting from state $s0$ and completing in state $s1$. Assume that both $a1$ and $a2$ have the same precondition $P1$, which holds in state $s0$. Also assume that the result of executing $a1$ is that $P1$ is deleted and the result of $a2$ is that $P2$ is added. Using the STRIPS assumption to compute the result of executing $a1$ in $s0$, we get that $P1$ is false in state $s1$ since it is deleted by $a1$. Using the STRIPS assumption to compute the effects of executing $a2$ in $s0$, we get that both $P1$ and $P2$ are true in $s1$, $P1$ being true since it is not mentioned in $a2$'s delete list. This of course is a contradiction, $P1$ and its negation cannot both hold at the same state.

Future Events in a State-Based System

Vere's DEVISER system can represent a planning environment where there are assertions about future events. In this system, a time line is modeled by the non-negative real numbers, zero being the time of planning and the positive real numbers being future times. This system can represent that an event not under the planning agent's control will occur starting at some specified time and ending at another time. The time line is also used for describing goal conditions and durations associated with actions. One can specify that some goal condition must hold between two time points. Vere calls the time points in which a goal is to be achieved the goal's (time) window. A goal can be the conjunction of two or more conditions to be

achieved simultaneously within one window, or conditions to be achieved at different times, thus each condition has its own specified window.

There are number of deficiencies in Vere's system that will be addressed here. One deficiency is that times of actions, events, and goals must be specified using an absolute scale. He does not provide a general representation that allows relative temporal orderings. For example, the system does not handle goals such as: get to the gas station before getting to school. The treatment of future events is also limited. One cannot represent that some future event will start any time between 2:00 and 2:15; the exact start and finish time must be given. Secondly, one cannot describe a future event that is influenced by both the external world and the planning agent. All future events must be external events, ones that the agent can work with and plan around, but not prevent.

Vere's system implicitly uses the STRIPS assumption just like other state-based planners. This leads to some potential problems because he is modeling a world where other agents and external forces may produce changes. One must assume that the planning environment completely describes all the external events that will occur while a plan is to be executed. This also precludes the use of disjunctions to describe the effects of actions and conditions that hold while a plan may be executed. If one does not assume that a complete description is given, incorrect results may be produced. For example, suppose the property "the bank is open" can only be affected by external events. If it is asserted that "the bank is open" holds in the initial situation, it is necessary to include the external event that negates this property at the time when the bank is closing. If this is not done, the precondition "the bank is open" is always satisfied since no matter where an action with this precondition is positioned, there are no earlier events or actions that delete this condition.

1.3. Overview of the Thesis

In order to analyze and circumvent the problems we have just described, we develop a model of action and time that represents concurrent actions and external events. Our starting point is Allen's interval logic [Allen 84]. In this formalism, a global notion of time is developed that is independent from the agent's actions. Temporal intervals are introduced to refer to chunks of time in a global time line. For any event, one can describe the temporal intervals over which the change associated with the event takes place. Thus, there is a notion of what is happening while an event is occurring. One can describe simultaneous events by stating that two events occur over intervals that overlap in time. *Properties*, which refer to static conditions, are treated similarly to events; for any property, one can describe the temporal intervals over which it holds. Both relative and absolute specifications can be used to temporally relate events and properties. In [Allen 83a], Allen describes a system, based on interval logic, that makes inferences about temporal relations using constraint propagation.

Allen's logic can be used to describe what actually happens over time, but cannot be used to describe the different possibilities that the agent can bring about. It can

be characterized as a *linear time logic*. Lacking from this logic is a construct like the result function in situation calculus that describes all the possible effects that can be produced by executing different actions in different situations. To accommodate this deficiency, we extend Allen's logic with two modal operators. Both these extensions are introduced in chapter two after discussing Allen's logic in more detail. First, we add a modal operator that captures temporal possibility enabling us to describe the different possible futures at some specified time. This allows us to distinguish between conditions that are possibly true and conditions that are inevitably true at a specified time. This extended logic can be characterized as a *branching time logic*.

After extending Allen's logic with the inevitability operator, we show that this extension alone is not sufficient for our purposes. For example, it does not distinguish whether a possibility is caused by the agent, caused by the external world, or caused by both factors. Making this distinction is necessary if we want to formalize what it means to say that a plan executed at a specified time solves the goal. For this purpose, we introduce a second modality **IFTRIED** which takes a **plan instance** and a sentence as arguments. Roughly, plan instances refer to both single actions and plans to be executed at specified times in specified ways. Plan instances take the place of plans and actions in our theory. The **IFTRIED** modality represents statements that can be interpreted as saying "if plan instance *pi* were attempted then sentence *S* would be true". There are two ways to look at this modality, either as a subjunctive conditional or as an to the result function from situation calculus.

In the third chapter, a formal specification of the logic is given. The first section describes the formal language and the second section describes the semantic model. Our basic approach to the semantics can be characterized as *possible worlds semantics* which was developed by Hintikka [Hintikka 62] and refined by Kripke [Kripke 63]. In this framework, a set of objects called *possible worlds* are included as part of a model. In our system, we refer to possible worlds as **world-histories** to emphasize that they correspond to worlds over time, not instantaneous snapshots. Each sentence in the language is given a truth value with respect to each world-history within a model. The truth value of sentences formed using the modal operators are interpreted in terms of relations and functions that relate world-histories. In chapter three, we pay particular attention to the functions that are used to interpret **IFTRIED**. This treatment derives from the semantic theories of conditionals developed by Stalnaker [Stalnaker 68] and Lewis [Lewis 73].

In chapter four, we present a proof theory that is sound with respect to the semantics. The axiomatization of most of the system is standard. The interval logic fragment is a first order theory that is formulated using a standard first order axiomatization extended with axioms describing the properties of a small collection of predicates and function terms found in interval logic. The inevitability modal operator behaves like a S5 necessity operator for a fixed time argument. An axiomatization of these properties is taken from Hughes and Cresswell [Hughes&Cresswell 1968]. The properties that are unique to this operator capture the relations: conditions that hold earlier than or during time *i* are inevitable at *i*, and what is inevitable at some time is inevitable at later times. The axioms and

rules capturing *IFTRIED* can be divided into three categories: i) properties relating to a subjunctive conditional, ii) the relations between *IFTRIED* and the inevitability operator, and iii) the relations between a the properties of a plan instance composed out of simpler plan instances and the individual properties of its constituent parts.

In chapter five, we use the logic that we have developed to analyze and describe the components of a planning problem having concurrent actions and external events. These components include the goal description, the planning environment description, the conditions under which plan instances can be executed both alone and in conjunction with other ones, and the effects produced by both simple and composite plan instances. We pay particular attention to the interaction between plan instances, both sequential and concurrent, and to the *persistence problem* [McDermott 82], which is the problem of determining how long a property remains true in a formalism that allows simultaneous events. By looking at these problems using our framework, we are able to explicate some of the problems that we mentioned earlier in conjunction with state-based systems. We illustrate how to represent some types of parallel interactions using our logic and describe how these interactions are used to determine the conditions under which two actions can be executed together. In the last two sections of chapter five, we examine some issues relating to the persistence problem. We analyze the relation between the persistence problem and the STRIPS assumption. This brings to light some of the problems encountered when the STRIPS assumption is used in an inappropriate setting. We also demonstrate that *the persistence assumption*, as put forth by some authors [McDermott 82] [Hanks&McDermott 85] as a replacement to the STRIPS assumption can lead to problems when reasoning about planning. We then discuss plan instances that maintain properties over intervals, which we use in place of the persistence assumption when reasoning about planning.

In chapter six, we present a planning algorithm that exploits some of the properties investigated in chapter five. We relate the algorithm to our logic and then prove that it can be viewed as a process that forms *sound* conclusions with respect to our semantic model. Our algorithm is novel in the way we handle harmful action interactions and the use of plan instances to maintain properties over temporal intervals, which are used in place of a STRIPS assumption. The harmful interactions between two or more plan instances, concurrent or sequential, are computed from a specification, given by the user, indicating for each pair of overlapping plan instances, the circumstances under which they do not interfere. By using maintenance plan instances, rather than the inappropriate STRIPS assumption or persistence assumption, we are able to remove the restrictions that must be imposed when using these assumptions.

In the last chapter, we present a summary of this work and discuss future directions. The future directions section discusses extensions along two dimensions. First, we discuss implementing and extending the planning algorithm that we developed in chapter six. We then discuss some issues outside of the scope of "the deductive planning problem", the focus of this work. In particular, the work in this thesis provides a precise account as to what it means to have an *airtight* plan that solve a goal with respect to some description of the world. We have not, however,

addressed issues concerned with choosing the appropriate description to work from, such as deciding which possible future events to take into account and which ones to ignore.

Chapter 2

Bringing in the Components of the Logic

In this chapter, we introduce our logic by bringing in the pieces one by one, describing why each piece is needed. The starting point is Allen's interval logic [Allen 84] which we discuss in section 2.1. This is a linear time logic that models simultaneous events and can be used to describe what is taking place while an event is occurring. In section 2.2, we introduce **plan instances** which roughly refer to actions and plans at specified times to be executed in specified manners. Plan instances take the place of plans and actions in our theory. In section 2.3, we illustrate why a linear time logic must be extended to represent the different possibilities that the agent can bring about. To demonstrate this point, we describe Allen's and Koomen's [Allen&Koomen 83b] method for using interval logic to solve planning problems and show where their system must go outside the logic and the ensuing problems. In section 2.4, we introduce the *INEV* modal operator which captures temporal possibility. Interval logic extended with *INEV* is a branching time logic, a logic that can describe different possible futures. We discuss some properties of *INEV* and examine the specification of a planning environment using a branching time logic. In section 2.5, we illustrate why a further extension is needed in order to represent what it means to say that a plan instance solves a goal. This brings to light that a branching time logic cannot distinguish whether a possibility is caused by the agent, caused by the external world, or caused by both factors. To make this distinction, we introduce a second modality, *IFTRIED* which can be used to describe what the planning agent can and cannot affect. We briefly discuss some of its properties in section 2.6.

2.1. Allen's Interval Logic

The starting part for our development is Allen's interval logic. Allen describes a logic of action and time [Allen 84] that overcomes the aforementioned deficiencies in situation calculus. His logic is cast as a sorted first order theory with terms that denote *temporal intervals* (or simply "intervals"), *events*, and *properties* (among other types). Intervals refer to stretches of time in a linear time line. Properties refer to static conditions which have different truth values at different times, and events refer to changes in affairs that occur over stretches of time. We will also use the term **conditions** to refer to both events and properties. A scenario over time may be described by specifying the events that occur over various intervals and the properties that hold throughout various intervals. Concurrent events are easily handled by asserting that two events occur over the same interval or occur over intervals that overlap in time. In a similar manner, one can describe the properties that hold while an event is taking place.

Allen's treatment of events differs from situation calculus in two important ways. First of all, events in Allen's logic refer only to partial changes in the world, they do

not refer to all the changes that take place during their time of occurrence. Thus, one can model two events happening at the same time. This contrasts with situation calculus, where events are complete changes from one situation to the next. In section 5.5, we describe Georgeff's [Georgeff 86] treatment of actions and events in which situation calculus is modified so that events are only partial changes.

The second difference is that Allen's logic can describe what is happening while an event is taking place. This, as we will see, enables a simple treatment of actions with conditions needed for execution that must hold during execution. For example, we will be able to treat an action corresponding to the performance of "editing a document on a text editor" which succeeds only if the text editor does not go down while the typing is taking place.

The Interval Logic Fragment

Our formal logic is an extension of interval logic with two modal operators. Thus, a fragment of our language will consist of interval logic statements. In this subsection, we informally present this fragment, which will be referred to as the interval logic fragment.

Before presenting the interval logic fragment, we must first describe the notational conventions that we will be using throughout this text. Atomic symbols, including constant terms, function symbols, and proposition symbols, will be italicized when presented within the text. All other terms and sentences, when presented within the text, will be surrounded by Quine corners, which appear as " \ulcorner " and " \urcorner ". All variable terms will be prefixed with a "?" such as the variable term $?pr$.

When not explicitly noted otherwise, we will be using a LISP-like notation to specify both sentences and function terms. Function terms, other than constants, are formed by encasing a function symbol followed by its arguments with parenthesis. For example $\ulcorner (f\ a) \urcorner$ refers to a unary function f with the constant term a as its argument. Atomic formulas, other than proposition symbols, are formed by encasing a predicate symbol followed by its arguments in parenthesis. Similarly, the first order connectives are put in prefix form and encased by parenthesis. The logical connectives will be specified by: AND for conjunction, OR for disjunction, NOT for negation, IF for material implication, IFF for equivalence, \forall for the universal quantifier, and \exists for the existential quantifier. Unbound variables in a statement are interpreted as universal variables bound at the outermost layer. Since both conjunction and disjunction are associative and commutative we will allow them to take two or more arguments. The predicate symbol $=$ will be used to refer to equality. Lastly, we must note that with the exception of the following subsection, we will be cavalier in treating the "use-mention" distinction and, for example, will use "property pr " in place of "the property denoted by pr ", pr being a constant term in the language.

Allen introduces a number of binary predicates to specify the temporal relation between two intervals. For example, $\ulcorner (MEETS\ i1\ i2) \urcorner$ means that the interval denoted by $i1$ immediately precedes the interval denoted by $i2$, with no gaps in

between. The *MEETS* relation is considered to be a primitive interval relation. In all, there are thirteen primitive interval relations which are given in appendix B. The set of primitive relations are mutually exclusive, that is, two different primitive relations cannot hold between the same two intervals. All other interval relations can be defined as a disjunction of these thirteen primitive relations. Two examples of defined interval relations, that we will make frequent use of, are: $\lceil(\text{PRIOR } i1 \ i2)\rceil$ and $\lceil(\text{IN } i1 \ i2)\rceil$. The relation $\lceil(\text{PRIOR } i1 \ i2)\rceil$ means that the interval denoted by $i1$ is before the interval denoted by $i2$. *PRIOR* is defined as the disjunction of the primitive relations *MEETS* and *BEFORE*, the latter relating two intervals where one is properly before the other. The relation $\lceil(\text{IN } i1 \ i2)\rceil$ means that the interval denoted by $i1$ is properly contained in the interval denoted by $i2$. This relation is defined as the disjunction of *STARTS*, *DURING*, and *FINISHES* (see appendix B). We will introduce other interval relations as needed throughout the text.

The binary predicate *OCCURS* is used to specify when an event occurs over some interval. The formula $\lceil(\text{OCCURS } ev \ i)\rceil$ means that the event denoted by ev occurs during the interval denoted by i . Similarly, the binary predicate *HOLDS* is used to specify when a property holds throughout some interval. The formula $\lceil(\text{HOLDS } pr \ i)\rceil$ means that the property denoted by pr holds throughout the interval denoted by i . Since *HOLDS* specifies that a property holds throughout some interval, there is an axiom stating: if a property pr holds during some interval $i2$, pr also holds during any properly contained in i . In the language, this is given by:

$$\begin{aligned} &(\text{IF}(\text{AND}(\text{IN } ?i2 \ ?i) (\text{HOLDS } ?pr \ ?i)) \\ &\quad (\text{HOLDS } ?pr \ ?i2)) \end{aligned}$$

It is important to distinguish this treatment of *HOLDS* with a modal approach such as in [Rescher&Urquhart 71] where an operator like *HOLDS* takes a sentence as an argument, instead of a term that denotes a property. In the modal approach, the sentence supplied as an argument can stand for a property or may also be a *HOLDS* statement (or an *OCCURS* statements for that matter) forming a statement with nested modal operators. An example, of a nested statement is $\lceil(\text{HOLDS}(\text{HOLDS } P \ i1) \ i2)\rceil$. These nested statements prove to be useful if one is translating tensed statements with tenses such as past perfect where there are two implicit times of reference along with the time that the sentence is uttered. For our purposes, however, nested statements provide no benefits. Unless one allowed "indexical intervals", such as a special interval denoting "now", all but the innermost times in a nested statement are superfluous. For example, $\lceil(\text{HOLDS}(\text{HOLDS } P \ i1) \ i2)\rceil$ is equivalent to $\lceil(\text{HOLDS } P \ i1)\rceil$. There is also an advantage to using our first order approach. One can quantify over the set of properties, which cannot be done in a first order modal logic.

Past, Present, and Future

In interval logic, there is no notion of a current time, and consequently, there is no formal notion of the past, present, and future. These tense distinctions have been

used, however, when we described the planning problems that we wish to formalize. We stated that the planning environment consists of a description of past, present and future conditions and the goal refers to desired future conditions. Thus, we are assuming that the agent is situated at a particular time, the time when planning is taking place. The terms "past", "present" and "future" are with respect to this time of planning. For simplicity, we will treat these tense distinctions in an informal manner. We will require that the planning problem specification identifies one of the intervals as being the time of planning (we place no importance to the duration of this interval). When describing a planning scenario, we will use the convention that the future refers to times that are later than the time of planning, the present refers to the time of planning, and the past refers to times prior to the time of planning. Thus for example, "future events" refers to events that occur after the time of planning.

We must contrast this informal treatment of past, present, and future with formal treatments found in tense logics such as [Prior 67] and [Rescher&Urquhart 71]. In these logics, sentences are interpreted with respect to a particular state (related to earlier and later states). One can simply assert that some property is true. An assertion such as "the sky is blue" is taken to mean that "the sky is blue, now". This contrast with our system where sentences are interpreted with respect to a whole course of events, not a particular time or state laying within a course of events. Consequently, assertions about properties can only be made using the *HOLDS* predicate where a time is explicitly given along with a property.

In a tense logic, one can also make assertion such as "in the future, the building will be completed", "in the past, Carol was liberal minded", "in the past, it was the case that in the future Tom would be seeing John". All these sentences are formed by combining propositions standing for properties with modal tense operators. For example "in the future, the building will be completed" is formed by prefacing the proposition "the building is completed" with the future modal operator. The statement "in the past, it was the case that in the future Tom would be seeing John" is formed by prefacing the proposition "Tom sees John" with the past operator followed by the future operator.

2.2. Plan Instances

Along with identifying the time that planning takes place, It is also necessary to identify the time when a plan is to be executed. The planning agent must be able to interact with external events and thus perform the steps in a plan at the appropriate times. Clearly, whether or not a plan achieves a given goal depends on the time during which its steps are executed. For this reason, we will talk about **plan instances** which have times of execution associated with each step, instead of plans which are usually taken to be timeless entities. We extend Allen's ontology with these objects and introduce a sort referring to plan instance terms.

A plan instance is taken to be a set of events at specified times to be brought about by a particular execution. In situation calculus, one could get away with being

vague as to whether an action refers to an event done in a particular way or whether it refers to any performance that results in an event's occurrence. This is because there is no notion of what is happening during the time of execution and hence no way to describe two performances of the same event that differ in how the event is being executed. In a more general model, however, this distinction becomes conspicuous since one can represent what is happening while an event is taking place. Consequently, if the logic is to be used to describe the effects of an action (or plan instance in our case) one must be clear as to whether actions refer to a particular way of bringing about some event, or any way of bringing about some event.

As an example, consider a simple scenario where there are two paths going from location *A* to location *B*. Let us refer to these paths as *P1* and *P2*. When talking about the action "bringing about the event going from *A* to *B*", one must be clear as to whether it refers to a particular way of going from *A* to *B* (i.e. whether it refers to going down path *P1*, or going down path *P2*), or whether it refers to taking either path. If the action refers to a particular way of causing this event, one can simply talk about its effects. If on the other hand, the action refers to the whole set of performances that can cause "go from *A* to *B*", one must distinguish between saying "no matter how the action is done, proposition *EFF* will be true" and saying "there is a way that the action is done such that *EFF* will be true". In this example, it is correct to say that there is a way for an agent to cause "go from *A* to *B*" such that this agent is on path *P1* during execution, but incorrect to say that no matter how an agent does "go from *A* to *B*, this agent is on path *P1* during execution.

We say that a plan instance *pi* occurs iff the events associated with *pi* all occur during their specified times and these occurrences are brought about by the particular execution associated with *pi*. In our language, we use the atomic formula $\lceil (\text{OCC } pi) \rceil$ to mean that the plan instance denoted by *pi* occurs. We must note that although a plan instance has a set of events at specified times associated with it, a plan instance may be denoted by a simple term, such as a constant, that does not explicitly give this set. For example, we may use the term *buy-at-BP* to refer to a plan instance in which the agent buys a newspaper during time *i1* and then buys five gallons of gas during time *i2* while at the BP gas station.

Along with the times associated with the set of events to be caused by the execution of a plan instance, we refer to a time of occurrence for the plan instance as a whole. We take the time of occurrence for a plan instance *pi* to be the smallest interval that contains all the times (i.e. intervals) associated with its set of events. In the language, we use the function term $\lceil (\text{TIME-OF } pi) \rceil$ to denote the time of occurrence associated with the plan instance, as a whole, denoted by *pi*. For convenience, if there is only one way to perform the event *ev* during *i*, we will sometimes use the term *ev@i* (where @ is a binary infix function) to refer to the plan instance corresponding to the execution of the single event *ev* at time *i*. Thus, plan instance *ev@i*'s time of occurrence is *i* since the smallest interval that contains {*i*} is simply *i*.

Any two plan instances can be composed together to form a more complex plan instance which occurs iff both its parts occur. In our language, the function term $\lceil(\text{COMP } pi1 \text{ } pi2)\rceil$ is used to denote the plan instance formed by composing the plan instances denoted by $pi1$ and $pi2$. The set of events at specified times associated with a composite plan instance is the union of the sets associated with its component parts. Secondly, the particular execution associated with a composite plan instance is the executions associated with both its parts done together. Consequently, a composite plan instance occurs iff both its component parts occur. Thus, we find that the following statement is valid in our logic:

$$\begin{aligned} &(\text{IFF } (\text{AND } (\text{OCC } pi1) (\text{OCC } pi2)) \\ &(\text{OCC } (\text{COMP } pi1 \text{ } pi2))) \end{aligned}$$

We also find that the time of occurrence associated with a composite plan instance is the smallest interval that contains both its component time of occurrence. Thus, the following statement is valid in our logic:

$$\begin{aligned} & (= (\text{TIME-OF } (\text{COMP } pi1 \text{ } pi2)) \\ & (\text{COVER } (\text{TIME-OF } pi1) (\text{TIME-OF } pi2))) \end{aligned}$$

where $\lceil(\text{COVER } i1 \text{ } i2)\rceil$ refers to the smallest interval that contains both $i1$ and $i2$

Our approach is non-standard in that we are composing plan instances instead of composing events, actions, or plans (if we had such objects). In effect, this allows us to form plan instances that contain concurrent actions, sequential actions, and actions with gaps separating their execution times all using the same combinator function. The standard approach is to have one combinator function for forming event sequences and a second combinator to form simultaneous events, while typically the formation of event types with gaps are not treated. In section 3.2.5, it will be shown that the execution associated with the composition of two plan instances (in terms of objects that we call basic actions) is given without needing to treat the composition of sequential plan instances differently from the composition of overlapping plan instances. Consequently, it is simpler to treat only one type of combinator function. Furthermore, we consider the treatment of different combinator functions as being concerned with the description of plan instances and events types. In this work, we are concentrating on what a plan instance is, not on the description of plan instances.

2.3. The Deficiencies of Interval Logic in Connection with Planning

Allen's treatment of time can be characterized as a linear time logic; interval logic statements are about what is actually true, not about what is possibly true. A logic to be used for planning, however, must represent possibilities. In particular, it must allow us to describe the different possibilities that the agent can bring about. In solving some goal, the planner must determine whether the goal is one of the possible conditions that the agent can bring about. Moreover, if the goal is possible, the planner must find a plan instance that solves the goal. Thus, a logic to be used

for planning must be capable of representing sentences of the form "*C* is a condition that the agent can bring about", and allow us to adequately formalize "plan instance *pi* solves goal *G*". To achieve this end, we first extend Interval logic so that it can be used to describe the different courses of events that are possible, not just the actual course of events. We will see, however, that this extension alone is not sufficient and that a second modal operator is needed to formalize "plan instance *pi* solves goal *G*".

Before presenting the extensions to interval logic, it will be illustrative to describe Allen's and Koomen's method [Allen&Koomen 83b] for using Interval logic for solving planning problems. They describe how interval logic may be used to generalize the type of planning problems handled by non-linear planners such as NONLIN [Tate 77] and NOAH [Sacerdoti 77]. Their approach enables them to treat planning environments that include statements about the past, present, and future, not just statements about some initial state as can only be done in Sacerdoti's and Tate's systems.

Statements in interval logic are used to describe the planning environment, to describe desired future conditions (the goal), and to investigate the effects of executing the various actions at the planning agent's disposal. To determine the effects of executing some action, one simply adds an assertion stating that this action occurs to the set of sentences describing the planning environment and the effects of the actions. Although they were able to describe future conditions as part of the planning environment, they were unable to reason about interfering with future conditions. The planner could only work around these future conditions. Any logic or mechanism that is used to reason about interfering with the future must be able to distinguish between conditions that are possible, and thus possibly could be prevented, and conditions that are inevitable and thus cannot be prevented. This distinction cannot be made in interval logic since it cannot describe possibility.

Now, in Allen's and Koomen's system, the planner concludes that a goal is solved by some set of actions making up a plan if i) the actions taken together bring about the goal in the planning environment, and ii) the "preconditions" for each action that is part of the plan hold in the planning environment or are enabled by another action, or set of actions, that is part of the plan.¹

The concept of "precondition" is outside the scope of interval logic and is treated in an informal manner. Allen and Koomen do not give a precise account as to what preconditions are. They just describe how one might use preconditions in a small number of examples and allude to the use of preconditions in state-based systems. As we will see, this informal approach leads to unacceptable results if one is not careful when specifying a planning environment description. We will later show that the extensions we make to interval logic may be used to capture what Allen and Koomen were getting at by appealing to preconditions.

¹ We are simplifying this description so as to only mention what is pertinent to the current discussion. In particular, we are glossing over their method for making frame assumptions.

The action specifications in Allen's and Koomen's system are given by relating each action at a specified time to its preconditions and effects. This is given by asserting that if an action occurs at some specified time then both the preconditions and effects hold at appropriate times. As an example, consider the class of actions where one block is stacked on top of another one. Let the function term $\text{[(stack } b1 \text{ } b2)]$ refer to a stacking action where block $b1$ is stacked on top of block $b2$. The preconditions for executing $\text{[(stack } b1 \text{ } b2)]$ during interval i is that both block $b1$ and $b2$ are clear immediately prior to i . The effects of executing $\text{[(stack } b1 \text{ } b2)]$ during i is that immediately after i , block $b1$ is stacked on top of $b2$ and $b2$ is no longer clear. In our language, this is given by:

```
(IF (OCCURS (stack ?b1 ?b2) ?i)
  (AND (∃ ?i1 (AND (MEETS ?i1 ?i) (HOLDS (clear ?b1) ?i1)))
    (∃ ?i2 (AND (MEETS ?i2 ?i) (HOLDS (clear ?b2) ?i2)))
    (∃ ?i3 (AND (MEETS ?i ?i3) (HOLDS (on ?b1 ?b2) ?i3)))
    (NOT (∃ ?i4 (AND (MEETS ?i ?i4) (HOLDS (clear ?b2) ?i4))))))
```

The above relation does not distinguish between preconditions and effects, and hence the preconditions have to be annotated outside the logic in order to distinguish them from effects. One might suggest that the distinction between preconditions and effects can in fact be made in interval logic since preconditions hold before the plan instance's time of execution and the effects hold afterwards. This, however, is not necessarily the case. Allen and Koomen stated that they were providing for "preconditions" that may hold while an action is being executed. For example, the preconditions for sailing across the lake may be that the wind is blowing while the sailing is take place. Thus, it is a misnomer to call these conditions preconditions.

One can look at preconditions as describing what the planning agent can possibly bring about and capturing what it means to say that a plan instance solves some goal². Any condition C is possible if there is a plan instance at a specified time whose effects bring about C and whose preconditions hold in the planning environment. This conception of preconditions is adapted from state-based planners and situation calculus, the context in which preconditions were originally introduced. Preconditions in interval logic, however, are not on the same firm ground as preconditions in the setting of state-based systems. This is because, in interval logic, there is no analogue to the underlying structure found in situation calculus in which alternative actions are applied to a situation yielding the different possible situations that the agent may bring about. That is, in interval logic, there is no notion of a context in which we apply alternative plan instances yielding the different possible effects that the agent may bring about. Thus, we cannot define preconditions in a similar manner as is done in situation calculus where preconditions are conditions describing the context (situation) in which the action is to be executed; an action's preconditions are the conditions under which application

² Although Allen and Koomen do not use the term "plan instance", we will use this term where it is appropriate, such as referring to a set of actions at specified times.

of the action in a context (situation) where the preconditions holds leads to a context (situation) in which the action's effects hold.

Having a clear conception of preconditions in state-based systems leads to an intuitive definition of action sequence preconditions. Without having an underlying structure in which to interpret preconditions in interval logic, it is difficult to give a satisfying account as to what a composite plan instance's preconditions are. While Allen and Koomen explicitly identify preconditions for simple actions at specified times, preconditions for composite plan instances are treated in an implicit manner. From the description of their planning algorithm we take liberties and try to give a more concise account as to what they intended composite plan instances preconditions to be.

Consider an example where a planning environment is given in which the preconditions for both $pi1$ and $pi2$ hold. In this case, Allen and Koomen would accept as a solution to some goal a plan instances that contained both $pi1$ and $pi2$ as long as there are no constraints, specified as part of the planning problem, that prohibit $pi1$ and $pi2$ from occurring together. Thus, we might suppose that they would subscribe to the following relation: if the precondition of $pi1$ is the temporally qualified proposition $PRE1$ and the precondition of $pi2$ is the temporally qualified proposition $PRE2$, then the precondition of their composition is $\lceil (AND\ PRE1\ PRE2) \rceil$ under the conditions that $\lceil (AND\ (OCC\ pi1)\ (OCC\ pi2)) \rceil$ is consistent (in interval logic) with the statements specified as part of the planning problem

One cannot use consistency to define what preconditions are³ It seems that Allen and Koomen are using consistency to approximate possibility in the following manner. They assume that if $\lceil (AND\ (OCC\ pi1)\ (OCC\ pi2)) \rceil$ is consistent with the statements given as part of the planning problem then it is possible that both $pi1$ and $pi2$ occur together. Thus, a more accurate description of preconditions for composite plan instances is: the precondition of the composition of $pi1$ and $pi2$ is $\lceil (AND\ PRE1\ PRE2) \rceil$ as long as it is possible that $pi1$ and $pi2$ occur together.

Approximating possibility using consistency leads to problems if one is not careful. For example, it must be assumed that a complete description of all the plan instances that cannot occur together is given as part of the planning problem description. If this is not done, the system will erroneously conclude that $pi1$ and $pi2$ can be executed together when it is in fact the case that they cannot be executed together, although this fact was omitted from the description. Therefore, if we want to construct planning systems that may not have complete knowledge about the world, in particular about action interactions, Allen and Koomen's strategy must be modified. To handle these case, it would be necessary to recognize that consistency is being used as just an approximation. A distinction would have to be made between the case where it is unknown whether two plan instances possibly occur together and the case where it is possible that they can be done together.

³ This is under the assumption that preconditions are properties of plan instances, not properties concerning an agent's beliefs about plan instances, as captured by a set of sentences.

Another problem is that the relation "the precondition of the composition of $pi1$ and $pi2$ is $\lceil (AND\ PRE1\ PRE2) \rceil$ if it is possible that $pi1$ and $pi2$ occur together" does not hold in some simple cases. For example, suppose that two simultaneous plan instances $ev1@i$ and $ev2@i$ share the same type of resource and that both $ev1@i$ and $ev2@i$ have the identical precondition "at least one resource is available during time i ". Also assume that the planning environment description includes the statement "either one or two resources are available during i and both disjuncts are possible". Consequently, it is possible that $ev1@i$ and $ev2@i$ can be done together. Using the precondition relation above, we would derive that the precondition of $ev1@i$ and $ev2@i$ taken together is "at least one resource is available during time i " conjoined with itself which is equivalent to "at least one resource is available during time i ". Clearly, this is unsatisfactory since $ev1@i$ and $ev2@i$ can be done together only if two resources are available during i .

We will see that the extensions that we make to interval logic remedy the problems that we have just noted. By appropriately modifying the linear time model, we are able to describe the different possibilities that the agent can bring about and capture the concept that Allen and Koomen were getting at by introducing preconditions. We will be able to formulate the relation between two plan instance's preconditions and their composition's preconditions, avoiding the problem above.

2.4. The Inevitability Modal Operator

We extend interval logic to model possible courses of events. At each interval i , there may be many ways that the future can complete depending on which possible events happen after i . Thus, we arrive at a world model over time that can be pictured as a tree that branches into the future. The set of intervals are arranged in a linear global time line and are used to pick out stretches in different branches that are co-temporal (see diagram 2.4-1). This type of temporal model can be characterized as a future branching time model. We are specifically designing the logic this way to capture all the possible courses of actions that the agent can take. For every possible circumstance where plan instance p_i can occur, it is assumed that there exists a branch where it occurs under these circumstances and a branch where it does not occur under these circumstances. Events not caused by the agent may also lead to a branching in the future. While we are being exact as to possibility with respect to the agent's behavior, we are allowing different conceptions of possibility as related to the external events, i.e. events not caused by the planning agent. We will shortly elaborate on this point.

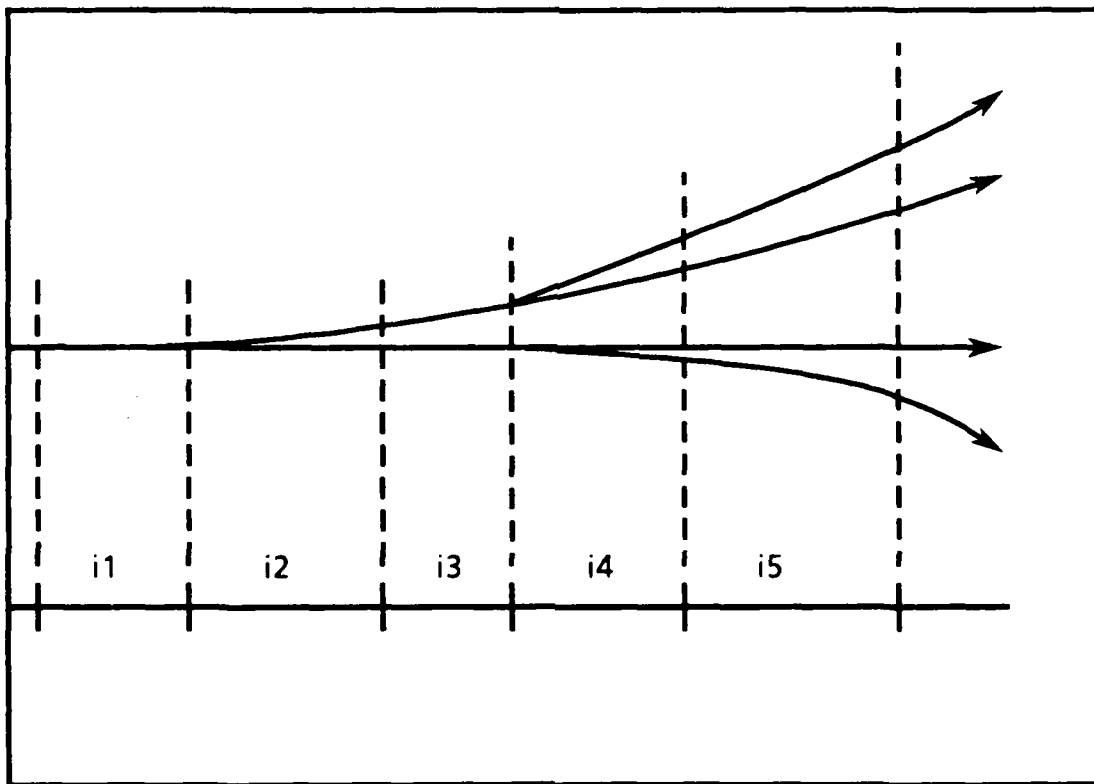


Diagram 2.4-1

To describe possible courses of events in our language, we introduce the "inevitability" modal operator which we designate by *INEV*. The *INEV* operator takes two arguments, an interval term and a sentence. We give the statement $\lceil (INEV\ i\ P) \rceil$ the english reading: "regardless of which possible events happen after i ,

P is true". The second argument P may be any sentence in our extended language (i.e. interval logic extended with *INEV*) Thus, we may form statements such as $\lceil(INEV\ i1\ (HOLDS\ pr\ i2))\rceil$ along statements containing a nested modal operator such as $\lceil(INEV\ i1\ (INEV\ i3\ (OCCURS\ ev\ i2)))\rceil$.

If intervals $i1$ and $i2$ finish at the same time, then $\lceil(INEV\ i1\ P)\rceil$ is true iff $\lceil(INEV\ i2\ P)\rceil$ is true. The reason for this is because the same events that are in the future of $i1$ are in the future of $i2$; it is only the end of the interval i that is relevant to the truth value of $\lceil(INEV\ i\ P)\rceil$. Secondly, if interval $i1$ ends before interval $i2$, then whatever is inevitable at time $i1$ is also inevitable at time $i2$. These two relations taken together can be succinctly stated using the following schema, which is valid for any sentence P in our language:

$$\begin{aligned} & (IF\ (ENDS \leq i1\ i2) \\ & \quad (IF\ (INEV\ i1\ P)\ (INEV\ i2\ P))) \end{aligned}$$

where $\lceil(END \leq i1\ i2)\rceil$ means that interval $i1$ ends before or at the same time as interval $i2$

A "possibility" modal operator is defined in terms of *INEV*. We will use *POS* to designate this operator, which like *INEV*, takes a term denoting an interval as its first argument and a sentence as its second argument. The statement $\lceil(POS\ i\ P)\rceil$ is given the english reading: "there is a course of events possible at time i in which P is true". *POS* is simply defined as the dual of *INEV* for a fixed time. This definition is given by:

$$(POS\ i\ P) =_{def} (NOT\ (INEV\ i\ (NOT\ P)))$$

Our treatment of the modal operators *POS* and *INEV* is compatible with interval logic in that the modal operators are temporally qualified. That is, one does not simply assert that a sentence is inevitable or possible, but instead that a sentence is inevitable or possible at a particular time. Statements in our logic are interpreted with respect to a branch within a tree of possible futures spanning all times, not with respect to a particular state situated in a tree of possible futures. This pertains to both modal and nonmodal (i.e. interval logic) statements. For example, consider diagram 2.4-2. With respect to branch $b3$, the statements $\lceil(HOLDS\ pr\ i2)\rceil$ and $\lceil(INEV\ i1\ (OCCURS\ ev\ i3))\rceil$ are true. The latter statement is true with respect to branch $b3$ since $\lceil(OCCURS\ ev\ i3)\rceil$ is true in branches $b3$, $b4$, and $b5$, these being all the branches that are possible with respect to $b3$ at time $i1$.

This treatment contrasts with a standard treatment of branching time logics found in the philosophy literature, such as in Prior [Prior 67] and Thomason [Thomason 70], where a one place inevitability operator is introduced that takes only a sentence as its argument. The reason for this difference is that these logics are cast as tense logics; statements are evaluated with respect to an instantaneous state that is arranged in a tree that branches into the future.

The interval and sentence supplied as arguments to an *INEV* or *POS* statement may have any temporal relation. There is a difference, however, in how $\lceil(INEV\ i1$

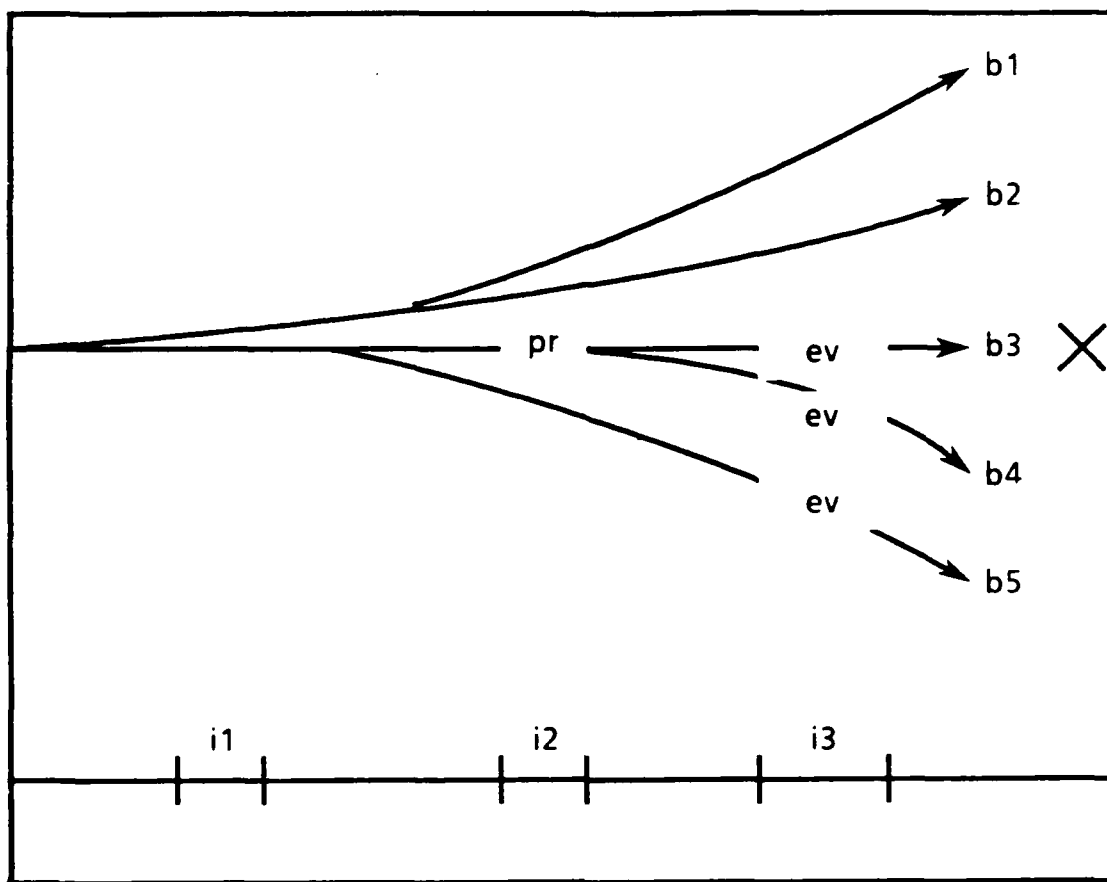


Diagram 2.4-2

$(\text{HOLDS } pr \ i2))^1$ is treated depending on the relation between the endings of $i1$ and $i2$. This is because if interval $i2$ ends before or at the same time as $i1$, then $\lceil (\text{HOLDS } pr \ i2) \rceil$ must be inevitably true at $i1$, or inevitably false at $i2$; a difference in later events has no bearing on earlier conditions. Thus, we find that the following statements are valid in our logic:

(IF(ENDS \leq ?i1 ?i2)
 (OR (INEV ?i2 (HOLDS ?pr ?i1))
 (INEV ?i2 (NOT (HOLDS ?pr ?i1)))))

(IF(ENDS \leq ?i1 ?i2)
 (OR (INEV ?i2 (OCCURS ?ev ?i1))
 (INEV ?i2 (NOT (OCCURS ?ev ?i1)))))

(IF(ENDS \leq (TIME-OF ?pi) ?i2)
 (OR (INEV ?i2 (OCC ?pi))
 (INEV ?i2 (NOT (OCC ?pi)))))

In the Artificial Intelligence literature, McDermott [McDermott 82] has put forth a branching time model to be used for describing scenarios over time and to be used for planning. He describes his general approach by saying that his logic can be thought of as a first order extensional logic that describes the interpretation of a modal temporal logic. He casts his logic as a sorted first order logic with terms that denote properties and events, similar to ours, along with terms that denote instantaneous states. The set of states are arranged in a tree that branches into the future by a "future-of" relation. To describe a scenario over time, one specifies the properties that hold at various states and events that occur over stretches of world-states arranged in the tree. Stretches of world states, more precisely convex sets of world-states ordered by the future-of relation, are referred to as intervals.

The intervals found in McDermott's system differ from the objects that we have been calling intervals in two ways. First of all, intervals in McDermott's logic are constructed out of point-like objects, while in our system intervals are treated as primitive objects. Secondly, intervals in his system refer to stretches in the tree of possible futures. This contrasts with our treatment where each interval is arranged in a linear global time line and serves to pick out stretches in different branches that are co-temporal. In McDermott's system, an additional function is needed to specify the global time associated with states in the tree of possible futures.

Since intervals in our system refer to a global time line, the temporal relation between two intervals does not vary from branch to branch in the tree of possible futures. This is in contrast to statements such as $\lceil (\text{OCCURS } ev \ i) \rceil$ and $\lceil (\text{HOLDS } pr \ i) \rceil$ whose truth value might vary from branch to branch. Thus, for example, we find that with respect to any time i , $\lceil (\text{MEETS } i1 \ i2) \rceil$ is either inevitably true or inevitably false. In our language, this is given by the following statement which is valid in our logic:

$$(\text{OR } (\text{INEV } i \ (\text{MEETS } i1 \ i2)) \\ (\text{INEV } i \ (\text{NOT } (\text{MEETS } i1 \ i2))))$$

Describing the planning environment

At planning time, there are many possible ways that the future may complete. The future that is actually realized depends on the actions (plan instances) that the agent chooses and the possible external events that actually occur. By choosing a particular set of future actions the agent constrains the set of possible futures that can be realized. Therefore, given a goal, the planner tries to find a plan instance that constrains the set of possible futures to a subset of possible futures in which the goal holds.

When describing a planning problem we do not work directly with world models but instead with a partial description of the world model as described by some set of sentences. We say that the sentences partially describe the world model since there may be many world models that fit a description given by a set of sentences. As we have mentioned, state-based planning problems are specified in this manner. In

these systems, the world model consists of an initial situation and the planning environment is specified by a set of sentences that partially describe the initial situation. The goal may be given by a conjunctive sentence that describes a set of desired properties that must hold in the situation that results from applying a plan. Given a goal G , the planner looks for a sequence of actions that transforms any initial situation that fits the planning environment description into a situation where G holds.

In our formalism, we cast the planning problem as follows. A world model consists of the set of branches that are possible at the time of planning. The planning environment is a partial specification of these possible branches and is described by statements in our extended language, i.e. interval logic extended with *INEV*. The goal is specified by an interval logic statement describing desired future conditions. Given a goal G , the planner looks for a plan instance that transforms any world model WM (which is a set of possible branches) that fits the planning environment description into a subset of WM in which G holds (in each branch belonging to the subset).

In our extended logic, modal statements describe possibility and inevitability, while statements such as $\lceil \text{(HOLDS } i \text{ pr)} \rceil$, $\lceil \text{(OCCURS ev } i) \rceil$, and $\lceil \text{(OCC } pi) \rceil$, not embedded within a modal operator, describe what is actually true. Thus, our extended language can describe conditions that are inevitable at some specified time, conditions that are possible at some specified time, and also, conditions that are actually true. This poses a slight problem. What does it mean to say that some course of events will possibly happen while also asserting that another course of events will actually happen? One answer to this is that possibilities are what could have happened. If this is the case, why should a planner worry about something that is possibly true if it is asserted that it is actually false?

We get around this problem by restricting the description of the planning environment so that it only includes statements about inevitable and possible conditions, omitting non-modal statements describing what is actually true. Thus, there will be no assertions about a condition that is actually false but possibly true. In solving a planning problem, we will be looking for a plan instance pi such that at planning time, it is inevitable that pi brings about the goal, not simply that pi brings about the goal. (note: Thomason [Thomason 70] describes a formal technique for avoiding the above problem (although his motivation for developing the machinery is different from ours). This solution involves giving a non-standard treatment to the semantic consequence operator which results in the fact that non-modal statements supplied on the left side act as if they are inevitably true. For our purposes, however, our simple solution is adequate and is compatible with Thomason's approach.)

We now see that conditions specified as part of the planning environment can be classified into three categories: i) conditions inevitably true at planning time, ii) conditions that are inevitably false at planning time, and iii) conditions that are both possibly true and possibly false at the time of planning. Both inevitably true and inevitably false conditions are ones that the agent can plan around but cannot

interfere with. Allen and Koomen [Allen&Koomen 83b] treated all future conditions as if they could not be interfered with, this being the result of not treating possibility. This is not to suggest, however, that the future conditions specified as part of Allen's and Koomen's planning environment would be treated as inevitably true in our system. This stems from the fact they did not put any restrictions on the type of future conditions that can be included as part of the planning environment. In particular, they allowed future conditions that described the agent's actions. These type of conditions cannot be treated as being inevitable in our formalism as we explain below.

The following assumption made in our system restricts the type of conditions that can be treated as being inevitable: for all possible circumstances where plan instance pi can be executed, it is possible (at all times prior to pi 's execution time) that pi occurs under these circumstances and possible it does not occur. Thus, it is incorrect to assert that it is inevitable that some future plan instance pi occurs as long as pi is not a plan instance such as "stay at home or do not stay at home during time i ", one that necessarily cannot occur. The effect of this restriction is more subtle than this though. It pertains not only to conditions that explicitly mention the agent's future actions or the agent's future locations, but also to some conditions that mention only the outside world. An example (taken from McDermott) will help to clarify. Consider the condition "Little Nell is tied to the train tracks and will be crushed by an oncoming train". Assume that Little Nell would be saved, if the planning agent executed the plan instance, *save-plan* and this plan instance could be executed. Thus, by our restriction above, it is possibly true that *save-plan* occurs. Consequently, the statement "Little Nell is tied to the train tracks and will be crushed by an oncoming train" cannot be inevitably true since if it were true, "save-plan occurs" would have to be inevitably false.

Solving The Goal

INEV was introduced as the first step in formulating what it means to say that a plan instance solves a goal. As we will see, an additional extension must be made in order to capture our intuitions, but let us first examine how one might try use interval logic extended with just *INEV* to formulate what it means to say that a plan instance solves a goal. Consider a planning problem where at planning time, which we will denote by Ip , we want to solve a goal G , where G is an interval logic statement describing desired future conditions (i.e. conditions in the future of Ip). We let S refer to the set of sentences describing the planning environment and the effects of executing each plan instance. A necessary condition to conclude that some future plan instance pi solves the goal G with respect to S is that pi brings about G under all future conditions that are possible at time Ip as described by S . Equivalently, we can describe this necessary condition by saying: for any world model WM (which consists of a set of branches possible at the time of planning Ip) in which all sentences in S are satisfied, G must hold in the subset of the branches in WM in which pi occurs. Formally, this condition can be given by relating S to the

sentence $\lceil (\text{INEV } I_p (\text{IF } (\text{OCC } p_i) G)) \rceil$, this sentence being true iff G holds in all the branches that are possible at I_p in which plan instance p_i occurs:

C1)

$$S \models (\text{INEV } I_p (\text{IF } (\text{OCC } p_i) G))$$

where the symbol \models designates the semantic entailment in our system. For a set S_0 , consisting of sentences in our language, and a sentence P in our language, " $S_0 \models P$ " is true iff sentence P is satisfied in every model in which all the sentences in S_0 are satisfied in. The formal definition of \models is given in chapter 3.

It is important to point out that we are treating S , which includes a specification of what is possible at planning time, as a parameter in formalizing what it means to say that a plan instance solves a goal. That is, we are formalizing " p_i solves the goal G with respect to S ", not simply " p_i solves the goal G ". As we mentioned earlier, while we are treating all the actions that the planning agent can execute as being possible, we are putting no constraints on which external events must be counted as possible. Whether the planning environment describes a great number of external possibilities, or just a few of the very likely ones is an interesting issue, but not of immediate concern here. We want to factor this problem out from the issues we are considering and thus make it clear how a mechanism that decides which possibilities to take into account can be integrated with the mechanisms based on the work here.

Now, we do not want to conclude that a plan instance solves a goal if it is an **impossible plan instance**, that is, a plan instance that cannot be executed under any circumstance. This point is mentioned because any impossible plan instance p_i would vacuously satisfy C1 since it is inevitable at any time that an impossible plan instance does not occur. Thus, we stipulate that a second necessary condition for "plan instance p_i solves the goal G with respect to S " is that it is possible at planning time that this plan instance occurs. That is, the following must be true:

C2)

$$S \models (\text{POS } I_p (\text{OCC } p_i))$$

A third necessary condition results from the following observations. A condition that the agent can prevent must be both possibly true and possibly false. Clearly, one cannot prevent a condition that is inevitably false. Secondly, the condition must be possibly false since it does not make sense to say that an agent prevented a condition that was going to happen anyway.¹ For example, it conflicts with conventional usage to say "I prevented winter for lasting for twelve months". For similar reasons, it is only proper to say that a condition is brought about if the condition is both possibly true and possibly false.

We see that a necessary condition for both " G is prevented" and " G is achieved" is that the G is both possibly false and possibly true. Thus, " G is both possibly true and possibly false" is a necessary condition for " p_i solves the goal G with respect to S ".

Now, $C1$ and $C2$ together entail that the goal is possibly true, that is, the following is true:

$$\{\ulcorner(\text{INEV } Ip (\text{IF } (\text{OCC } pi) G))\urcorner, \ulcorner(\text{POS } Ip (\text{OCC } pi))\urcorner\} \models (\text{POS } Ip G)$$

On the other hand, $C1$ and $C2$ (taken together) do not entail that G is possibly false. In other words, the negation of $\ulcorner(\text{POS } Ip (\text{NOT } G))\urcorner$ is consistent (in our logic) with $C1$ and $C2$ taken together. Thus, in order to conclude that pi solves the goal G , it is also necessary that the following holds:

$C3)$

$$S \models (\text{POS } Ip (\text{NOT } G))$$

Although $C1$, $C2$, and $C3$ are necessary conditions for " pi solves the goal G with respect to S ", they are not sufficient. Other necessary conditions are that pi can be executed regardless of possible circumstances out of the agent's control and p' must contain all the steps needed for execution. As we will see, interval logic extended with *INEV* alone is inadequate to represent " pi solves the goal G with respect to S ".

2.5. What the Branching Time Structure Does Not Capture

If it is possible at planning time that plan instance pi occurs, then it is only possible that pi can be successfully executed. It does not necessarily follow that pi can be executed under all circumstances that are possible at planning time. For one, there may be possible external events whose occurrence would prevent pi from being successfully executed. We want to conclude that a plan instance pi solves a goal only if under all possible circumstances caused by external events, pi can be executed.

Consider the following planning problem. The goal is to get into a particular room sometime during the interval I_g , which is in the future of I_p , the time of planning. Let *walk-in-room@I_w* refer to a plan instance corresponding to the performance of the event "walk through the doorway into the room" during interval I_w , where we are assuming that I_w ends at a time during I_g . We assume that it is possible that this plan instance occurs iff the door is unlocked at a time just prior to I_w . Thus, the execution associated with *walk-in-room@I_w* does not have a provision for unlocking the door if it happens to be locked. We also assume that it is possible at I_p that the door is locked just prior to I_w and possible at I_p that the door is unlocked

¹ To be more exact, we would have to distinguish the reason why an event is possible. It is not proper to say that the agent prevented some event from occurring if the event can only be caused by the agent's actions alone. In [Pelavin&Allen 86] (section 4), we describe how the logic being developed here can be used to represent that a possible event is not just caused by the agent's actions. This is achieved by stating that it is possible that the event would occur even if the agent were to remain inactive.

just prior to I_w . Furthermore, the agent cannot affect whether or not the door is locked. A diagram of such a scenario is given in 2.5-1.

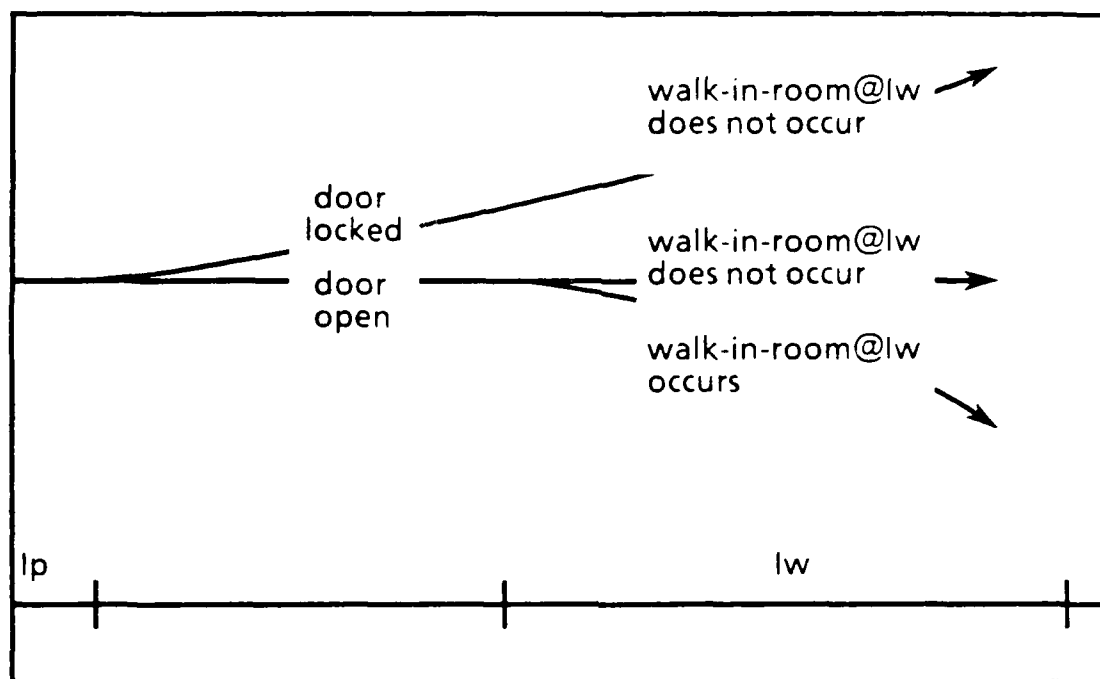


Diagram 2.5-1

In this scenario, it is possible at I_p that the plan instance *walk-in-room@I_w* occurs, but it is not appropriate to say that plan instance *walk-in-room@I_w* solves our goal. Whether or not it can be successfully executed depends on whether or not the door is locked, a condition out of the agent's control. A plan instance is desired that can be executed under all possible occurrences of external events as described by the planning environment. Thus, in the above example, a plan instance is sought that achieves the goal and can be executed regardless as to whether the door is locked or open.

Finding a plan instance that can be executed regardless of external conditions is still not sufficient. In order to correctly say that a plan instance solves a goal, the plan instance must contain all the steps needed for execution. This can be clarified by the following example.

During planning time I_p , the agent is standing by a locked safe and by a table on which the safe's key is resting. The goal is to open the safe at some time in the near future. Let *open-safe@I_o* refer to a plan instance corresponding to the performance of the event "open the safe with the key" during interval I_o . It is possible that this plan instance occurs iff the agent is grasping the key in its hand just prior to execution time I_o . The agent can also perform *grasp-key@I_g* which corresponds to grasping the safe's key at a time immediately after I_p and results in the key being grasped just prior to I_o . We assume, in this simple scenario, that *grasp-key@I_g* can be simply executed by itself. Thus, it is possible at I_p that *grasp-key@I_g* occurs

leading to a possible branch where the agent is grasping the key in its hand just prior to execution time, Io . Consequently, it is possible that $open-safe@Io$ occurs. A diagram of such a scenario is given in 2.5-2.

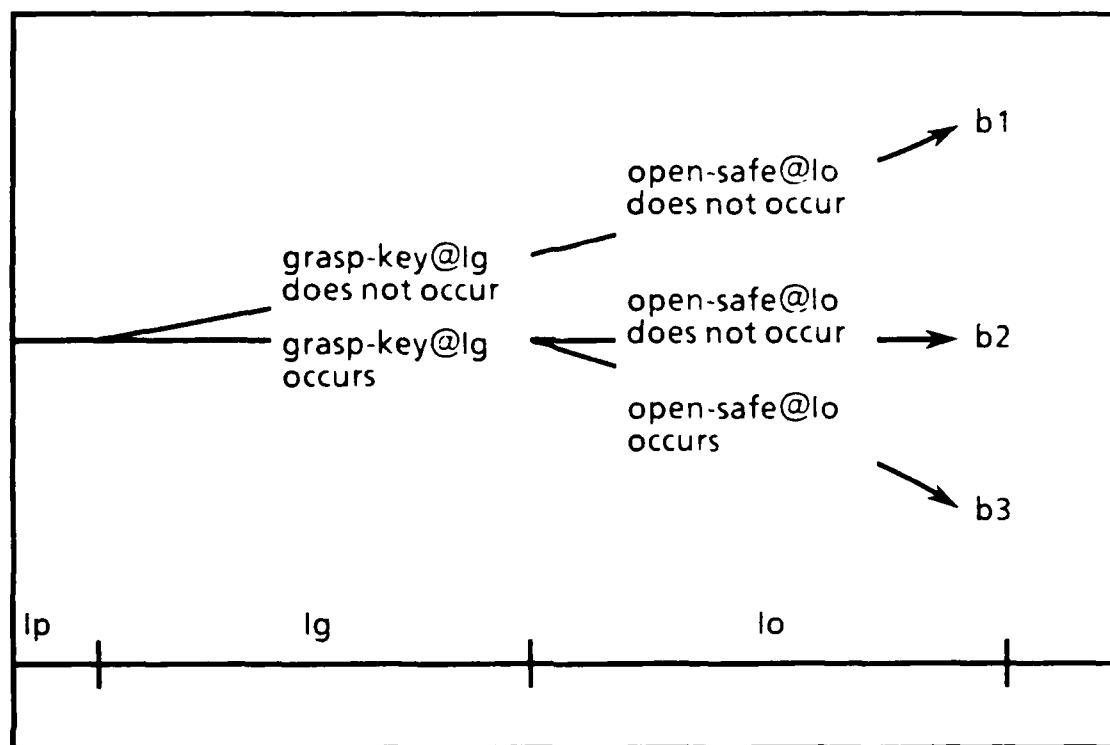


Diagram 2.5-2

In this scenario, it is under the agent's control to enable $open-safe@Io$ so that it can be executed. This is done by executing $grasp-key@lg$. We would not, however, want the planner to simply return that $open-safe@Io$ solves the goal, leaving out that it must be done in conjunction with $grasp-key@lg$. Instead, we would want the planner to return a plan instance such as $\lceil (COMP\ grasp-key@lg\ open-safe@Io) \rceil$, i.e. the plan instance corresponding to the joint execution of $grasp-key@lg$ and $open-safe@Io$.

In summary, we have shown that a necessary condition to conclude that "plan instance pi solves goal G with respect to S " is that pi can be simply executed alone under all possible circumstances at planning time as described by S . This insures that there is no possible external conditions that can prevent pi from being executed and that pi contains all the steps need for execution. Equivalently, we describe this necessary condition by saying that pi is **executable** in all possible futures at planning time as described by S . The property "executable" is described as follows:

pi is executable in branch b in the tree of futures iff either pi occurs in b or pi could have occurred in the circumstances holding in b without the aid of another plan instance to enable it

As an example, the plan instance *open-safe@I_o* is executable in branches *b2* and *b3*, but not in *b1* as pictured in diagram 2.5-2. As we describe in section 5.2, this term is also used by Pollack[Pollack 86] to refer to a similar concept.

Analyzing what we mean by executability is crucial to formalizing "plan instance *pi* solves the goal *G* with respect to *S*". We now dig deeper, asking the question: why do we say that a plan instance is executable under certain circumstances, but not under others? We then show that a branching time structure must be extended to capture executability.

Executability and Plan Instance Attempts

We arrive at an analysis of executability by making a distinction between plan instance attempts and plan instance occurrences. A similar distinction is made by Haas [Haas 85] where he distinguishes between the execution of a command to bring about an action and the occurrence of an action. When we first introduced plan instances, we stated that each one referred to a set of events, at particular times, to be brought about by a particular execution. Now, it is possible that the particular execution associated with a plan instance takes place although the set of events associated with the plan instance does not occur. In this case, we say that the plan instance is **attempted** but does not occur. A plan instance *pi* occurs iff the performance of the execution associated with *pi* brings about the occurrence of the events associated with *pi*. We equate "*pi* is executable" with "if *pi* were to be attempted, then *pi* would occur". This definition makes use of a subjunctive conditional, an important point which we will shortly get back to.

Consider the previous example where we were reasoning about opening a safe with a key. Implicitly, we had in mind that this plan instance is attempted by the agent moving it's arm in such a way that the key being grasped is twisted in the safe's lock. Now, these arm movements can be performed regardless as to whether or not the key is being grasped. These arm movements, however, will only bring about a safe opening event if the key is being grasped just prior to execution. In diagram 2.5-1, we omitted, for simplicity, a branch where the arm movements are performed while the key is not being grasped.

As a second example, consider a plan instance that corresponds to the performance of the event "editing a document on a text editor" during interval *i*. This plan instance is attempted by typing on a keyboard while sitting in front on a text editor. Now, typing on a keyboard can be done regardless as to whether the text editor is operational during interval *i*, but the event "editing a document on a text editors" is brought about during *i* only if the machine is operational during *i*.

The Branching Time Structure and Executability

We now examine the relation between the branching time structure and executability. Executability can be captured in a branching time structure only if for every plan instance pi , conditions prior to pi 's time of occurrence determine whether or not pi is executable. In this case we can encode " pi is executable by":

$(POS\ Ix\ (OCC\ pi))$

where Ix is any interval that immediately precedes (meets to the left) pi 's time of occurrence

The above relation between executability and the branching time structure is justified by showing the equivalence of " pi is executable in branch b " and " $\lceil (POS\ Ix\ (OCC\ pi)) \rceil$ is true at b ". This equivalence can only be established if conditions prior to pi determine whether or not it is executable.

The implication from " pi is executable in branch b " to " $\lceil (POS\ Ix\ (OCC\ pi)) \rceil$ is true at b " holds even in the case where conditions during execution, along with prior conditions, determine executability. This relation stems from our assumption that for each possible circumstance in which a plan instance can occur, it is possible that this plan instance occurs under these circumstances. From this we infer that if plan instance pi could have occurred in the circumstances holding in b without the aid of another plan instance to enable it, then there must be a branch in which pi occurs that shares a common past with b up until the beginning of pi 's execution time. Now, a plan instance is executable in branch b iff it occurs in b or could have occurred in the circumstances holding in b without the aid of another plan instance that enables it. Consequently, if pi is executable in b then pi occurs in b or there is a branch in which pi occurs that shares a common past with b up until the beginning of pi 's execution time. This consequent is exactly the condition described by " $\lceil (POS\ Ix\ (OCC\ pi)) \rceil$ ". That is, " $\lceil (POS\ Ix\ (OCC\ pi)) \rceil$ " is true at branch b iff pi occurs in b or there is a branch in which pi occurs that shares a common past with b through Ix , an interval that meets pi 's execution time.

The converse relation "if " $\lceil (POS\ Ix\ (OCC\ pi)) \rceil$ is true in b then pi is executable in b " can be established only if conditions prior to pi determine whether or not it is executable. If " $\lceil (POS\ Ix\ (OCC\ pi)) \rceil$ " holds in b , then it is clear that conditions prior to pi 's time of occurrence cannot preclude pi from being executable in b . Thus, the only way that " $\lceil (POS\ Ix\ (OCC\ pi)) \rceil$ " can be true at b , while pi is not executable at b , is if conditions during pi 's time of execution preclude pi from being executable. (note: we are working under assumption that conditions after pi 's time of occurrence have no bearing on whether or not pi is executable) This establishes the desired implication: "if only prior conditions determine executability, then if " $\lceil (POS\ Ix\ (OCC\ pi)) \rceil$ is true in b then pi is executable in b "

Let us now examine the situation where executability is determined (at least partly) by conditions that hold during execution. We have just shown that we cannot equate " pi is executable" with " $\lceil (POS\ Ix\ (OCC\ pi)) \rceil$ " if conditions during execution affect whether pi is executable. We now present an example to

demonstrate that the branching time model does not have all the structure needed to capture executability. Consequently, "pi is executable" cannot be reduced to any statement, not just $\lceil (\text{POS } Ix (\text{OCC } \text{pi})) \rceil$, given solely in terms of the language formed by extending interval logic with the modal operators *INEV* and *POS*.

Consider the following example. Let *sail@I* refer to a plan instance corresponding to the performance of the event "sail across the river" during interval *I*. We assume that this plan instance is executable in a branch in which the wind is blowing throughout interval *I*. In diagram 2.5-3, a simple model is depicted with four branches. In branch *b1*, the wind is blowing throughout interval *Iall* which contains intervals *Ix*, *I*, *I2*, and *I3*. In branch *b2*, the wind is blowing throughout interval *I2*, but does not blow during interval *I3*. Additionally, the agent does not try to execute *sail@I* in either *b1* or *b2*. Now, branch *b3* is like *b2* in that the wind blows during *I2*, but not during *I3*. In this branch the agent attempts to execute *sail@I* but fails since in the midst of execution the wind stops blowing. In branch *b4*, however, the agent attempts *sail@I* and succeeds (i.e. *sail@I* occurs) since the wind is blowing throughout interval *I-all* which contains *I*, the time of execution.

The implication from " $\lceil (\text{POS } Ix (\text{OCC } \text{sail@I})) \rceil$ is true in branch *b*" to "*sail@I* is executable in *b*" does not hold for all branches in this example. In particular, it does not hold in branch *b2*. This is because $\lceil (\text{POS } Ix (\text{OCC } \text{sail@I})) \rceil$ holds in branch *b2* since *sail@I* occurs in *b4*, but *sail@I* is not executable in *b2* since the wind does not blow throughout interval *I* in this branch.

The reason for describing this example is to show that the branching time structure does not capture executability when there are conditions during execution that determine whether or not a plan instance is executable. The branching time structure in this example just captures: i) *b1*, *b2*, *b3*, and *b4* all share a common past through interval *Ix*, ii) *b1* and *b2* share a common past through *I2*, and iii) *b3* and *b4* share a common past through *I2*. What is lacking from this structure are links such as one between *b1* and *b4* that capture our implicit intention that *b1* differs from *b4* on the account of attempting *sail@I*. Similarly, *b2* and *b3* should be related in this manner. If we had such a relation between *b2* and *b3*, we could determine that if the agent were to attempt *sail@I* in an environment given by *b2*, it would be in a branch (i.e. *b3*) in which the attempt does not lead to an occurrence of *sail@I*. This would allow us to conclude that *sail@I* is not executable in *b2*.

We will see in section 3.2.4 that our underlying model is an extension of a branching time model with links, as we have suggested, that relate branches that differ solely on the account of some plan instance being attempted. We now describe the extension we make to our language to describe branches that differ on the account of some plan instance attempt.

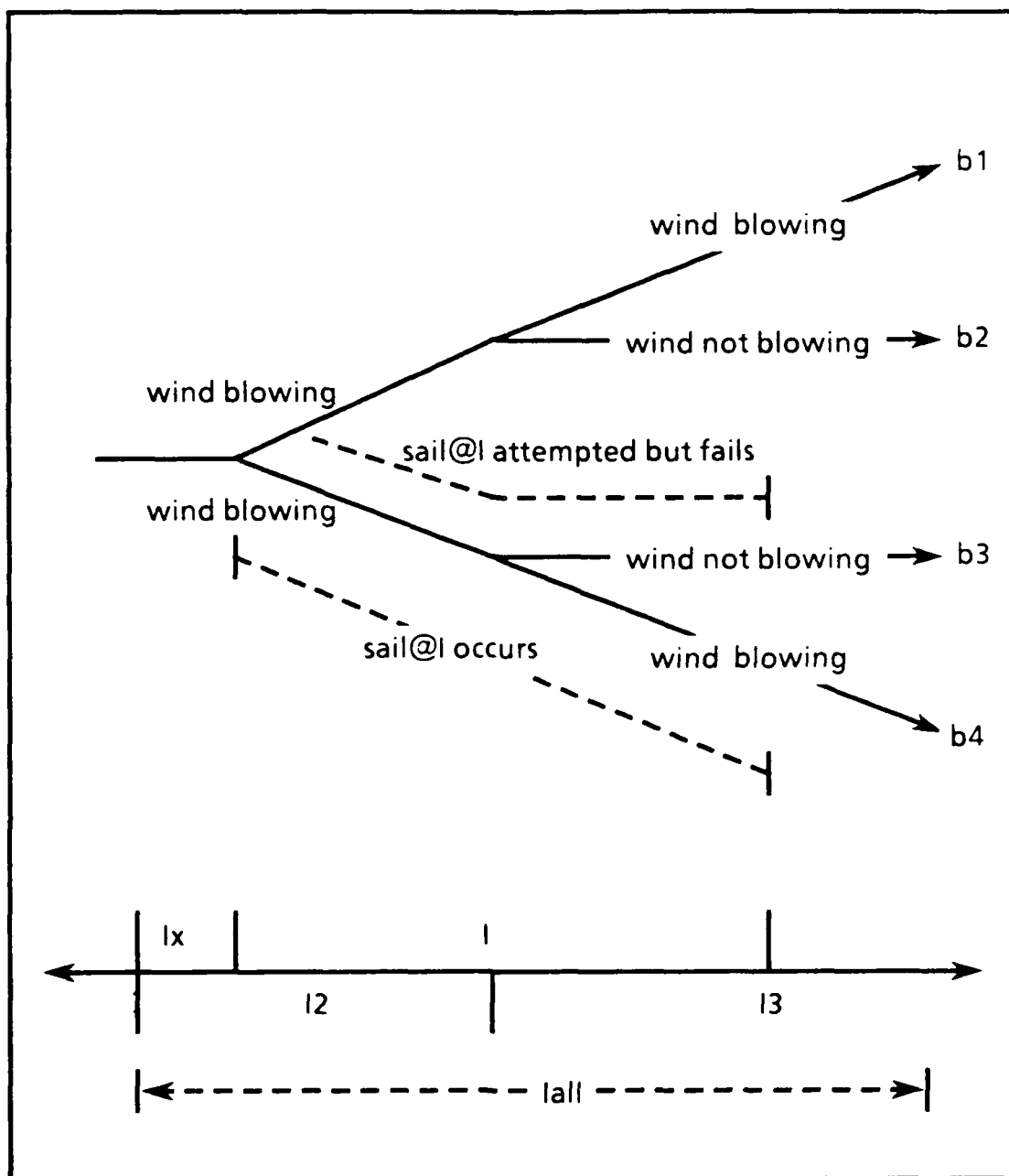


Diagram 2.5-3

2.6. The IFTRIED Modal Operator

We extend our language with the modal operator *IFTRIED* to describe the result of attempting a plan instance under different circumstances, this being the last extension we make. *IFTRIED* takes two arguments, a plan instance term and a sentence. Both arguments to *IFTRIED* are temporally qualified and may have any temporal relation. The first argument, being a plan instance, corresponds to a set of events to be executed at specified times. The second argument is a statement in the extended language that consists of interval logic extended with the *INEV* and *IFTRIED* modal operators. Therefore, the second argument to *IFTRIED* will be either a interval logic statement, and hence temporally qualified, or a modal statement whose innermost layers consists of interval logic statements.

We give $\lceil (IFTRIED\ pi\ P) \rceil$ the english reading: "if *pi* were to be attempted, then *P* would be true". As we will see in section 3.2.6, our treatment of *IFTRIED* closely resembles the semantic theories of subjunctive conditionals put forth by Stalnaker [Stalnaker 68] and Lewis [Lewis 73]. Just like a subjunctive conditional, an *IFTRIED* statement describes the result of minimally modifying some context to take into account the antecedent being true. In our case, the context is a branch in some tree of possible futures and the antecedent is "plan instance *pi* is attempted". The statement $\lceil (IFTRIED\ pi\ P) \rceil$ is true at branch *b*, if *P* is true in all the branches that are arrived at by minimally revising *b* to take into account "*pi* is attempted". In section 3.2.5, we discuss this in great detail and we explain why there may be multiple branches minimally differing from some branch. In this section we just briefly note some properties of *IFTRIED*. In this section, we just briefly note some properties of *IFTRIED*. A detailed presentation will be given in chapter 4 where we discuss a proof theory and in chapter 5 where we describe how our logic may be used to represent and solve planning problems.

Executability is defined in terms of *IFTRIED*. We will use the formula $\lceil (EXECUTABLE\ pi) \rceil$ to mean that plan instance *pi* is executable. The statement $\lceil (EXECUTABLE\ pi) \rceil$ is simply defined by:

$$(EXECUTABLE\ pi) =_{def} (IFTRIED\ pi\ (OCC\ pi))$$

The above definition can be read as saying: plan instance *pi* is executable iff if *pi* were to be attempted then *pi* would occur. This treatment of executability allows us to handle conditions that determine executability that hold during execution. For example, we can state that *sail@I* is executable in any branch *b* possible at *I_p* iff the wind is blowing during interval *I* in branch *b*. This can be given by:

$$(INEV\ I_p\ (IFF\ (HOLDS\ wind\ blowing\ I)\ (EXECUTABLE\ sail@I)))$$

Along with describing what the attempt of *pi* would affect, *IFTRIED* may also be used to describe what a plan instance does not affect. As we will see, describing what a plan instance does not affect is very important when determining whether two plan instances can be executed together. To state that the attempt of a plan instance *pi* does not affect the temporally qualified condition *C_t* under all conditions possible at

planning time I_p , we simply state that it is inevitable at I_p that if C_i is true then if p_i were to be attempted then C_i would (still) be true. For example, the following describes the situation where the attempt of plan instance *walk-outside* @ I does not affect whether or not it is raining outside during any interval.

```
(INEV Ip
  (AND (IF (HOLDS raining ?ir)
           (IFTRIED walk-outside@I (HOLDS raining ?ir) )))
  (IF (NOT (HOLDS raining ?inr))
       (IFTRIED walk-outside@I
        (NOT (HOLDS raining ?inr))))))
```

When describing a planning problem, it is not necessary to state that a plan instance does not affect any earlier condition since this result is a theorem in our system. For instance, the following statements are valid in our logic:

```
(IF (PRIOR ?i (TIME-OF ?pi))
     (IF (HOLDS ?pr ?i)
          (IFTRIED ?pi (HOLDS ?pr ?i))))

(IF (PRIOR ?i (TIME-OF ?pi))
     (IF (OCCURS ?ev ?i)
          (IFTRIED ?pi (OCCURS ?ev ?i))))

(IF (PRIOR (TIME-OF ?pi1) (TIME-OF ?pi2))
     (IF (OCC ?pi1)
          (IFTRIED ?pi2 (OCC ?pi1))))
```

As we have discussed, a necessary condition to conclude that plan instance pi solves any goal is that it is inevitable at planning time that pi is executable. This insures that there are no external events possible at planning time that prevent pi from occurring and that at planning time, pi contains all the steps needed for execution. We take this condition along with conditions $C1$, $C2$, and $C3$, described earlier, to be the necessary and sufficient conditions for concluding that "plan instance pi solves goal G with respect to S ". These four conditions are given by:

- C1) $S \models (\text{INEV } I_p (\text{IF } (\text{OCC } pi) G))$
- C2) $S \models (\text{POS } I_p (\text{OCC } pi))$
- C3) $S \models (\text{POS } I_p (\text{NOT } G))$
- C4) $S \models (\text{INEV } I_p (\text{IFTRIED } pi (\text{OCC } pi)))$

As we will see in chapter 5, condition $C2$ can be omitted since $C2$ is entailed by $C4$.

We conclude this chapter by briefly showing how action specifications are described and how a composite plan instance is related to its component parts. In chapter 5, we discuss these and other issues in detail after giving a formal

description of the language, the model theory, and the proof theory in chapters 3 and 4.

Action Specifications

Typically, the action specifications for a planning problem are given by specifying the "effects" and "preconditions" for each simple action. In our formalization of the planning problem, the specification for plan instance pi is given by describing the conditions that will inevitably hold at planning time if pi occurs, and conditions C under which it is inevitable at planning time that if C holds then pi is executable. For example, consider the plan instance $move-box@I$ which corresponds to bringing about the event "move the box against the wall" during interval I by sliding the box along the floor. The actions specifications for this plan instance may be given by:

$$\begin{aligned} & (INEV Ip \\ & \quad (IF(OCC move-box@I) \\ & \quad \quad (\exists ?i2 (AND (MEETS I ?i2) (HOLDS (against box wall) ?i2)))) \end{aligned}$$

$$\begin{aligned} & (INEV Ip \\ & \quad (IF(AND (\exists ?i0 (AND (MEETS ?i0 I) \\ & \quad \quad \quad (HOLDS (near agent box) ?i0))) \\ & \quad \quad (HOLDS no-obstructions-in-path I)) \\ & \quad (EXECUTABLE move-box@I))) \end{aligned}$$

The first expression says that it is inevitable at planning time that if $move-box@I$ occurs then the box will be against the wall immediately after the occurrence of $move-box@I$. This statement does not indicate how long the box will be against the wall since there may be other possible plan instances or external events that cause the box to be moved at some time in the future of I .

The second statement says that under all possible circumstances at planning time, $move-box@I$ is executable if the agent is by the box just prior to execution and there is no obstructions that will be in the way while the box is being moved.

In general, statements describing conditions for executability will be given in the form:

$$(INEV Ip (IF C_t (EXECUTABLE pi)))$$

where C_t is a sentence in the interval logic fragment

The conditions designated by C_t in the above expression closely resemble the conditions that Allen and Koomen [Allen&Koomen 83b] referred to by preconditions. In cases where Allen and Koomen would say that the temporally qualified statement C_t is pi 's preconditions, we would say that if C_t holds in any branch possible at planning time, then pi is executable in that branch. It is important to note, however, that we are formalizing executability, not preconditions. A formal treatment of preconditions would have to bring in the problem of deciding

what conditions to consider as being possible and what conditions to ignore when solving a planning problem. This problem has been called *the qualification problem* [McCarthy 80]. Implicit in the specification of an action's preconditions is a decision on what is considered possible and what is to be ignored. For example, suppose the preconditions for bringing about the event "starting the car" (by turning the key in the ignition and then stepping on the gas pedal) are "the agent is in the car, the ignition system is working and there is gas in the tank". Invariably there are conditions that can preclude the car from being started that are omitted from this precondition list. For example, if a potato is in the tail pipe the car cannot be started. This condition, however, would probably be omitted since its occurrence is highly unlikely.

Thus, we see that by omitting conditions that can ruin an occurrence, one is implicitly treating what is considered to be possible. Consequently, any formal treatment of preconditions would have to consider the problem of deciding which conditions should be taken into account and which conditions should be ignored. As we mentioned earlier, we want to factor this problem out from the formal analysis being presented here.

Composing plan instances

A central operation performed by traditional planning systems is the construction of a plan by composing simple actions together. Typically, a planning system is given the preconditions and effects for simple actions from which the system computes the preconditions and effects for action sequences, i.e. plans, constructed out of these simple actions. This process can be performed since action sequences are defined in terms of the individual actions and consequently the preconditions and effects for action sequences can be computed from the preconditions and effects of the individual actions making up the sequence.

Analogously, in our logic, we define composite plan instance in terms of the its components. This construction is given in section 3.2.5. Consequently, we will be able to give theorems that relate the conditions under which a composite plan instance is executable to the conditions under which its components are executable when taken alone. This formal treatment avoids the problems we mentioned in section 2.3 where we described the pitfalls of characterizing the preconditions for a composite plan instance.

A rough intuitive view of executability for composite plan instances is as follows. The composition of $pi1$ and $pi2$ is executable with respect to some branch b if i) they are both executable at b , and they do not "interfere" with each other at b , or ii) one of them, say $pi1$, is executable in b , the attempt of $pi1$ enables the conditions under which $pi2$ is executable and they do not interfere with each other at b . As we will see, the concept of "interference" plays a central role in formalizing plan instance composition. For example, consider two simultaneous plan instances $ev1@i$ and $ev2@i$ that share the same type of resource. In any branch b in which there is not at

least two resources available during i , we say that the two plan instances interfere at b .

In our language, we can characterize the situation where it is inevitable at planning time that $pi1$ and $pi2$ do not interfere with each other as follows:

```
(AND (INEV Ip (IF (OCC pi1) (IFTRIED pi2 (OCC pi1))))
      (INEV Ip (IF (OCC pi2) (IFTRIED pi1 (OCC pi2)))))
```

The first statement can be read as saying that in all branches possible at Ip where pi occurs, the attempt of $pi2$ would not ruin $pi1$'s occurrence. The second statement has a symmetrical reading. We get back to these issues, in detail, in section 5.3 where we describe how different types of plan instance interactions may be represented in our language.

Chapter 3

The Formal Specification of the Logic

In this chapter, a formal description of our logic is given. In section 3.1, we give a formal presentation of the language, and in section 3.2, we present the model theoretic semantics. In just this chapter, we will use symbols having the form " t_i " (i.e. "t" followed by a subscripted sequence of letters or numbers) to refer to terms in the object language. We adopt this convention because we are discussing both the object language and the model structure in this chapter. Thus, our convention is used to clearly distinguish between symbols referring to terms in the object language and symbols that refer to objects in the model structure.

3.1. The Language

The specification of the formal language is given in terms of a primitive language. All other operators and predicates that we use are defined in terms of the primitive language. These definitions are given in appendix B. In section 3.2, where we give the semantic interpretations to sentences in the language, we only specify interpretations for the primitive symbols. The defined symbols inherit their interpretations from the symbols they are defined in terms of. Thus, by minimizing the number of primitive symbols, we can give a more succinct interpretation. As an example, we treat negation and disjunction as being primitive, but do not treat conjunction as being primitive since this operator can be defined in terms of negation and disjunction. Alternatively, we could have introduced negation and conjunction and defined disjunction in terms of them. As a second example, the *MEETS* binary predicate is the only interval relation that is part of the primitive language. All other interval relations can be defined in terms of *MEETS* as described in Allen and Hayes [Allen&Hayes 85] and described here in appendix B. In the following presentation, informal text interspersed with the formal specification of the primitive language. An encapsulated description of the primitive language is given in Appendix A.

The Primitive Language

Our formal language is an extension of a sorted first order language with two modal operators, *INEV* and *IFTRIED*. The language is presented by first constructing *terms* which are linguistic entities that denote objects. Terms are formed from a collection of basic symbols which are classified as *individual variables*, *functions symbols* and special function symbols which correspond to functions that are given fixed interpretations in all models. Associated with each term is a type which corresponds to the class of the term's denotation. *Atomic formulas* are

constructed by prefixing a sequence of terms (of appropriate length) with a predicate symbol belonging to the set $\{=, \text{MEETS}, \text{HOLDS}, \text{OCCURS}, \text{OCC}\}$. The arguments to the predicates *MEETS*, *HOLDS*, *OCCURS*, and *OCC* are syntactically restricted so that they denote objects of the appropriate class. For example, the two arguments related by the *MEETS* predicate are restricted to be terms that denote temporal intervals. Finally, *well formed formulas* are formed by combining the atomic formulas with the standard first order connectives and the two modal operators. The set of well formed formulas are also referred to as the sentences in our logic.

The collection of basic symbols are classified as either individual variables or function symbols.

VAR_{01} refers to the set of individual variables $\{?v_1, ?v_2, \dots\}$

FN_{01} refers to the set of function symbols $\{f_1, f_2, \dots\}$

Associated with each function symbol is the number of arguments that it takes, i.e. the function's arity. The arity of each function symbol is specified by the function *DEG* which takes a function symbol as an argument and yields the non-negative integer associated with the function symbol's arity. IF $\text{DEG}(f_i) = 0$, we say that f_i is a *constant*.

A set of types refers to the different sorts that may be assigned to each function symbol and variable. The set of types is given by:

$\text{TYPES}_{01} = \{\text{OBJ}_{01}, \text{INT}_{01}, \text{PROP}_{01}, \text{EI}_{01}, \text{PI}_{01}\}$

These types correspond to:

OBJ_{01}	the class of physical objects in the world
INT_{01}	the class of temporal intervals
PROP_{01}	the class of properties
EI_{01}	the class of event instances
PI_{01}	the class of plan instances

The type associated with a variable corresponds to the class of objects over which the variable ranges, and the type associated with a function symbol corresponds to the range of the function.

TYPE-OF refers to the function that takes a variable or function symbol as an argument and yields the type (member of TYPES_{01}) associated with its argument

The set of terms are constructed from the function symbols, variables, and two special function symbols *TIME-OF* and *COMP*. Following is a recursive definition of the set of terms TERMS_{01} and the definition of the function *TYPE-OF** which specifies the type associated with each term.

Each variable $?v_i$ is a term and $\text{TYPE-OF}^*(?v_i)$ is defined as the type associated with $?v_i$.

For every variable $(?v_i)$, $?v_i \in \text{TERMS}_{01}$
and $\text{TYPE-OF}^*(?v_i) =_{\text{def}} \text{TYPE-OF}(?v_i)$

Each constant f_i , i.e. function symbol of arity 0, is a term and $\text{TYPE-OF}^*(f_i)$ is defined as the type associated with f_i .

For every function symbol (f_i) such that $\text{DEG}(f_i) = 0$, $f_i \in \text{TERMS}_{01}$
and $\text{TYPE-OF}^*(f_i) =_{\text{def}} \text{TYPE-OF}(f_i)$

A left parenthesis followed by a function symbol f_i of arity n , $n > 0$, followed by n terms, followed by a right parenthesis is a term, and TYPE-OF^* applied to this construct yields the type associated with f_i 's range.

For every function symbol (f_i) such that $\text{DEG}(f_i) = n > 0$
and terms (t_1, t_2, \dots, t_n) , $\lceil (f_i t_1 t_2 \dots t_n) \rceil \in \text{TERMS}_{01}$
and $\text{TYPE-OF}^*(\lceil (f_i t_1 t_2 \dots t_n) \rceil) =_{\text{def}} \text{TYPE-OF}(f_i)$

The construct $\lceil (\text{COMP } t_1 t_2) \rceil$ is a term, where t_1 and t_2 are terms of type plan instances, and the type of COMP is a plan instance.

For all terms $(t_1$ and $t_2)$ such that $\text{TYPE-OF}^*(t_1) = \text{PI}_{01}$
and $\text{TYPE-OF}^*(t_2) = \text{PI}_{01}$, $\lceil (\text{COMP } t_1 t_2) \rceil \in \text{TERMS}_{01}$
and $\text{TYPE-OF}^*(\lceil (\text{COMP } t_1 t_2) \rceil) =_{\text{def}} \text{PI}_{01}$

The construct $\lceil (\text{TIME-OF } t_1) \rceil$ is a term, where t_1 is a term of type plan instance, and the type of TIME-OF is a temporal interval.

For every term (t_1) such that $\text{TYPE-OF}^*(t_1) = \text{PI}_{01}$,
 $\lceil (\text{TIME-OF } t_1) \rceil \in \text{TERMS}_{01}$
and $\text{TYPE-OF}^*(\lceil (\text{TIME-OF } t_1) \rceil) =_{\text{def}} \text{INT}_{01}$

An atomic formulas is formed by prefixing a sequence of terms (of the appropriate length) with a predicate symbol belonging to $\{=, \text{MEETS}, \text{HOLDS}, \text{OCCURS}, \text{OCC}\}$ and encasing this construct in parenthesis. Following is a definition of the set of atomic formulas AF_{01} .

All constructs of the form $\lceil (= t_i t_j) \rceil$ are atomic formulas, if both t_i and t_j are terms of any type.

For all terms $(t_i$ and $t_j)$, $\lceil (= t_i t_j) \rceil \in \text{AF}_{01}$

All constructs of the form $\lceil (\text{MEETS } t_i t_j) \rceil$ are atomic formulas, if both t_i and t_j are temporal interval types.

For all terms (t_i and t_j) such that $\text{TYPE-OF}^*(t_i) = \text{INT}_{\text{typ}}$
and $\text{TYPE-OF}^*(t_j) = \text{INT}_{\text{ol}}$, $\lceil (\text{MEETS } t_i t_j) \rceil \in \text{AF}_{\text{ol}}$

All constructs of the form $\lceil (\text{HOLDS } t_i t_j) \rceil$ are atomic formulas, if t_i is a property type and t_j is temporal interval type.

For all terms (t_i and t_j) such that $\text{TYPE-OF}^*(t_i) = \text{PROP}_{\text{ol}}$
and $\text{TYPE-OF}^*(t_j) = \text{INT}_{\text{ol}}$, $\lceil (\text{HOLDS } t_i t_j) \rceil \in \text{AF}_{\text{ol}}$

All constructs of the form $\lceil (\text{OCCURS } t_i t_j) \rceil$ are atomic formulas if t_i is an event type and t_j is temporal interval type.

For every term (t_i) such that $\text{TYPE-OF}^*(t_i) = \text{EI}_{\text{ol}}$
and $\text{TYPE-OF}^*(t_j) = \text{INT}_{\text{ol}}$, $\lceil (\text{OCCURS } t_i t_j) \rceil \in \text{AF}_{\text{ol}}$

All constructs of the form $\lceil (\text{OCC } t_i) \rceil$ are atomic formulas if t_i is a plan instance type.

For every term (t_i) such that $\text{TYPE-OF}^*(t_i) = \text{PI}_{\text{ol}}$
 $\lceil (\text{OCC } t_i) \rceil \in \text{AF}_{\text{ol}}$

The set of well formed formulas is constructed by combining the atomic formulas with the modal operators *INEV* and *IFTRIED* and the standard first order connectives. As is typically is done, we use capital letters, such as *P*, *Q*, and *R*, to refer to the well-formed formulas. The recursive definition of the set of well formed formulas WFF_{ol} is given by:

For every atomic formula (af_i),
 $af_i \in \text{WFF}_{\text{ol}}$

For every well formed formula (*P*),
 $\lceil (\text{NOT } P) \rceil \in \text{WFF}_{\text{ol}}$

For all well formed formulas (*P* and *Q*),
 $\lceil (\text{OR } P \text{ } Q) \rceil \in \text{WFF}_{\text{ol}}$

For every variable ($?v_i$) and well formed formula (*P*),
 $\lceil (\forall ?v_i P) \rceil \in \text{WFF}_{\text{ol}}$

For every term (t_i) such that $\text{TYPE-OF}^*(t_i) = \text{INT}_{\text{ol}}$
and well formed formula (*P*), $\lceil (\text{INEV } t_i P) \rceil \in \text{WFF}_{\text{ol}}$

For every term (t_i) such that $\text{TYPE-OF}^*(t_i) = \text{PI}_{\text{ol}}$
and well formed formula (*P*), $\lceil (\text{IFTRIED } t_i P) \rceil \in \text{WFF}_{\text{ol}}$

3.2. The Semantics

We present the semantics for our logic by grafting the non-modal fragment of our logic, i.e. the fragment corresponding to Allen's interval logic, into a model structure that also interprets the modal operators *INEV* and *IFTRIED*. This approach is similar to that of Hughes and Cresswell [Hughes&Cresswell 68] who present the semantics of propositional modal logic and quantified modal logic by grafting propositional logic and first order logic, respectively, into a model structure that also interprets the necessity operator. Their basic approach can be characterized as *possible worlds semantics* which was developed by Hintikka [Hintikka 62] and refined by Kripke [Kripke 63].

In the possible worlds framework, a set of objects called *possible worlds* (or simply *worlds*) is identified as part of a model. Each sentence in the language is given a truth value with respect to each possible world within a model. Thus, a model in the possible world framework is a specification of what is true at a set of worlds, not just the specification of one world.

Relations on possible worlds, called *accessibility relations*, are introduced in the model to give interpretations to the modal statements. The truth value of a modal statement at a world w depends on worlds related to w by an accessibility relation. For example, to interpret the necessity modality, one introduces an accessibility relations that relates worlds that are possible with respect to each other. The statement, "necessary P " is interpreted as true at possible world w iff the statement P is true at all worlds that are accessible from w . The statement, "possibly P " is interpreted as true at possible world w iff there exists a world accessible from w at which P is true.

The truth value of a non-modal statement is only dependent on the possible world at which it is evaluated, not other worlds linked by accessibility relations. At a fixed possible world, the interpretation function is isomorphic to the interpretation function for the underlying non-modal fragment. For example, in propositional logic, the interpretation for conjunction would be given by:

For all sentences (P and Q)
 $V(\ulcorner \text{AND } P \text{ } Q \urcorner) = \text{TRUE}$ iff $V(P) = \text{TRUE}$ and $V(Q) = \text{TRUE}$

In the possible worlds framework, the interpretation for conjunction would be given by:

For all sentences (P and Q), and possible worlds (w)
 $V(\ulcorner \text{AND } P \text{ } Q \urcorner, w) = \text{TRUE}$ iff $V(P, w) = \text{TRUE}$ and $V(Q, w) = \text{TRUE}$

A sentence is *valid* if it is assigned the value of true at every possible world in every model. As a consequence every non-modal statement that is valid in the underlying non-modal fragment is also valid when the fragment is extended with modal operators. In the following chapters, we will use the notation " $\models P$ " to mean that sentence P is valid with respect to our semantic theory. We will also use " \models " in binary form. The relation " $S \models P$ " means that sentence P is true in every possible

world in any model in which all the sentences in S are true in. Consequently, " $\emptyset \models P$ " is equivalent to " $\models P$ ".

In section 3.2.1, we describe the interpretation of terms and the interpretation of interval logic statements. In the next section, we present the accessibility relation that is used to interpret *INEV* statements. In section 3.2.3, we describe the mathematical objects that model plan instances and relate our treatment to Goldman's theory of action [Goldman 70]. We also describe the interpretation of the predicate *OCC* and functions *TIME-OF* and *COMP* which take plan instance terms as arguments (and which are also considered part of the non-modal fragment). In section 3.2.4, we describe **basic action (functions)**, the structures in terms of which *IFTRIED* statements are interpreted. In the next section, we discuss the combination of these functions which serves to define the composition of two plan instances in terms of its components. Finally, in section 3.2.6, we discuss the relation between the interpretation of *IFTRIED* and the semantic theories of conditionals put forth by such authors as Stalnaker [Stalnaker 68] and Lewis [Lewis 73].

In this section, the formal specifications of the model are interspersed with informal discussion. We also omit the interpretation of the first order order connectives, this treatment being straightforward. In Appendix C, an encapsulated description of the complete model structure is presented.

3.2.1. The Interval Logic Fragment

The basic components of a model corresponding to the interval logic fragment are a set of possible worlds, a set of domain individuals, and an interpretation function that maps terms to domain individuals and assigns interval logic statements truth values at each possible world. We now discuss these components.

In each model, a non-empty set of possible worlds is identified. We can think of the set of interval logic statements that are true with respect to a world w as a partial description of w . Since the set of interval logic statements that are true at a world may refer to properties that hold and events that occur at many different times, this set describes a world over time, not an instantaneous snapshot. For this reason, we will use the term **possible world-history** (or simply **world-history**) to refer to a possible world. In chapter 2, world-histories were informally referred to as branches in the tree of possible futures.

H refers to the set of possible world-histories

Domain Individuals and the Interpretation of Terms

In the language, there are terms that denote objects, temporal intervals, properties, events, and plan instances. Our models must identify classes of domain

individuals that correspond to these objects and the interpretation function must map terms to the appropriate class. In the model, the following sets are identified:

OBJ	the non-empty set of physical objects that existed at any time in any world-history
INT	the non-empty set of temporal intervals
PROP	the non-empty set of properties
EV	the non-empty set of event types
PI	the non-empty set of plan instances

The sets *OBJ*, *INT*, *PROP*, *EV*, and *PI* are pair-wise disjoint. We define *D* as the union of *OBJ*, *INT*, *PROP*, *EV*, and *PI*. We refer to *D* as the set of domain individuals.

Each model identifies an interpretation function that maps variables and function symbols with arity 0 (i.e. constants) to the domain individuals that they denote and maps function symbols with arity greater than 0 to functions on *D* with the same arity. This interpretation function will be referred to as V_{vf} . In terms of V_{vf} , we define a function that maps terms in the object language into the domain individuals that they denote.

V_t refers to the function from $TERM_{ol}$ to *D* that maps a term in the object language to the domain individual that it denotes

The construction of V_t in terms of V_{vf} is given in appendix C. Constraints are placed on V_{vf} so that variables and constants are mapped to domain individuals belonging to the appropriate class (e.g., if f_i is a constant of type OBJ_{ol} , then $V_{vf}(f_i)$ is constrained to belong to the set *OBJ*), and functions of arity greater than 0 are mapped to functions on *D* whose range is restricted to the appropriate class.

Our treatment of the domain of individuals and the denotation of terms is a simplification in two respects. First of all, we are assuming that the set of domain individuals and the subsets making up the different classes is constant over world-histories. This is in contrast to a model where there is a set of domain individuals for each world-history. In fact, by treating possible worlds as being world-histories, instead of being instantaneous snapshots, we are precluding a treatment where a domain is defined for each time in each history.

Our second simplification is that a term's denotation does not vary from world-history to world-history. We say that all terms in our language are *rigid designators*. This is in contrast to a model in which there is a two place interpretation function from $TERM_{ol} \times H$ to *D* that assigns each term a possibly different denotation at each world-history. Treating all terms as rigid designators is a strong restriction, but greatly simplifies matters.

The Interpretation of the Interval Logic Atomic Formulas

The model must specify the truth value for each sentence (i.e. well formed formula) at each world-history in the model.

V_s refers to the function from $WFF_{ol} \times H$ to $\{TRUE, FALSE\}$ that assigns each sentence a truth value at each world-history

In this section, we only present the truth values for the atomic formulas (the equality atomic formula, and the *MEETS*, *HOLDS*, and *OCCURS* atomic formulas), these being the basic building blocks for forming any interval logic statement. The interpretation of the first order connectives, which combine the atomic formulas with each other and with the modal statements (in the complete language, not the interval logic fragment), are given in appendix C.

The interpretation of atomic formulas having the form $\lceil (= t_1 t_2) \rceil$, i.e. the equality atomic formula, is straight forward. The formula $\lceil (= t_1 t_2) \rceil$ is interpreted as true at world-history h iff t_1 and t_2 denote the same domain individual. Formally, this is given by:

For all wffs of the form $\lceil (= t_1 t_2) \rceil$ and world-histories (h),
 $V_s(\lceil (= t_1 t_2) \rceil, h) = TRUE$ iff $V_t(t_1) = V_t(t_2)$

Notice that the interpretation of $\lceil (= t_1 t_2) \rceil$ is independent of the world-history at which it is interpreted. This is a direct consequence of our decision to treat terms as rigid designators.

Interval Relations and the Interpretation of MEETS

The interpretation of the atomic formula $\lceil (MEETS t_{int1} t_{int2}) \rceil$ is also independent of the world-history at which it is evaluated. We explicitly design our models so that if $\lceil (MEETS t_{int1} t_{int2}) \rceil$ is true at one world-history, it is true at all world-histories. The reason for treating interval relations this way is to provide for the comparison of two different world-histories at some particular time. If each world-history had its own private time line, we would not be able to compare world-histories. This implies that we need a global time line that picks out common times across all the world-histories. To achieve this end, each model specifies a relation on intervals that arranges the intervals to form a global time line.

Allen and Hayes [Allen&Hayes 85] describe a construction in which all the interval relations are defined in terms of the *meets* relation (i.e. the relation between two intervals where one interval immediately precedes the other) as long as we make a few assumptions about the existence of intervals (which is given in appendix C). Thus, the arrangement of the intervals in a time line can be completely characterized by the *meets* relation.

$MTS(i1, i2)$ refers to the relation defined over intervals that is true iff interval $i1$ meets interval $i2$ to the left

The interpretation of the atomic formula $\lceil (MEETS\ t_{int1}\ t_{int2}) \rceil$ is given directly in terms of MTS :

For all wffs of the form $\lceil (MEETS\ t_{int1}\ t_{int2}) \rceil$ and world-histories (h),
 $V_s(\lceil (MEETS\ t_{int1}\ t_{int2}) \rceil, h) = \text{TRUE}$ iff $MTS(V_t(t_{int1}), V_t(t_{int2}))$ is true

Since the right hand side of the interpretation above (i.e. " $MTS(V_t(t_{int1}), V_t(t_{int2}))$ is true") does not mention the world-history h , the truth value of $\lceil (MEETS\ t_{int1}\ t_{int2}) \rceil$ does not vary from world-history to world-history.

The model must place restrictions on MTS so it is indeed an arrangement of intervals in a linear time line. This characterization of a *meets* relation is given in Allen and Hayes [Allen&Hayes 85] and is adopted here as described in appendix C.

Properties and Events

In each model, two properties are equal if at every world-history they hold during the same intervals. This is essentially the same treatment as given by McDermott [McDermott 82] where properties are defined as the set of instantaneous states over which the properties hold. We therefore take each property (i.e. each element of the set $PROP$) to be a set of elements of the form $\langle i, h \rangle$ where i belongs to the set of intervals, and h belongs to the set of world-histories. Intuitively, if $\langle i, h \rangle$ belongs to property pr , then pr holds during interval i in world-history h .

We take "property pr holds over interval $i1$ " to be equivalent to "property pr holds throughout interval $i1$ ". Thus, if a property holds during interval $i1$, this property holds over all intervals contained in $i1$. This is given by the following constraint which is imposed on our models:

PROP1)

For all properties (pr), intervals ($i1$ and $i2$), and world-histories (h)

If $IN(i1, i2)$ and $\langle i2, h \rangle \in pr$ then $\langle i1, h \rangle \in pr$

where $IN(i1, i2)$ is true iff $i1$ is properly inside of $i2$. Its definition in terms of MTS is given in appendix B

The atomic formula $\lceil (HOLDS\ t_{pr}\ t_{int}) \rceil$ is interpreted as true at world-history h iff the property denoted by t_{pr} (i.e. $V_t(t_{pr})$) holds during the interval denoted by t_{int} (i.e. $V_t(t_{int})$) in world-history h , this being true if $\langle V_t(t_{int}), h \rangle$ belongs to the set $V_t(t_{pr})$. Thus, the interpretation of $\lceil (HOLDS\ t_{pr}\ t_{int}) \rceil$ is given by:

For all wffs of the form $\lceil (HOLDS\ t_{pr}\ t_{int}) \rceil$ and world-histories (h),

$V_s(\lceil (HOLDS\ t_{pr}\ t_{int}) \rceil, h) = \text{TRUE}$ iff $\langle V_t(t_{int}), h \rangle \in V_t(t_{pr})$

Events are treated in a similar fashion as properties. Each event (i.e. each element of the set EV) is taken to be a set of elements each having the form $\langle i, h \rangle$ where i belongs to the set of intervals and h belongs to the set of world-histories. Intuitively, if $\langle i, h \rangle$ belongs to event ev , then ev occurs during interval i in world-history h . The interval over which an event occurs is taken to be the smallest interval over which the event from beginning to end takes place; thus, we do not impose a constraint on events that is analogous to *PROP1*. We also do not impose the constraint described by Allen [Allen 84] that says that if an event ev occurs over i , it does not occur over any interval contained in i . The reason for not imposing this constraint is so that we can model a situation where there are two instances of the same event type occurring, one during the other. An example of this is where an instance of "the agent's hand is waved", referring to the right hand being waved, occurs during the time when another instance occurs, referring to the left hand being waved.

The atomic formula $\lceil (\text{OCCURS } t_{ev} \ t_{int}) \rceil$ is interpreted as true at world-history h iff the event denoted by t_{ev} occurs during the interval denoted by t_{int} in world-history h , this being true if $\langle V_t(t_{int}), h \rangle$ belongs to the set $V_t(t_{ev})$. Thus, the interpretation of $\lceil (\text{OCCURS } t_{ev} \ t_{int}) \rceil$ is given by:

For all wffs of the form $\lceil (\text{OCCURS } t_{ev} \ t_{int}) \rceil$ and world-histories (h),
 $V_s(\lceil (\text{OCCURS } t_{ev} \ t_{int}) \rceil, h) = \text{TRUE}$ iff $\langle V_t(t_{int}), h \rangle \in V_t(t_{ev})$

3.2.2. The Interpretation of *INEV* and the *R* Accessibility Relation

We take $\lceil (\text{INEV } t_{int} \ P) \rceil$ to mean that P is true no matter which possible events happen after the time denoted by t_{int} . Appropriately, the truth value of an *INEV* statement depends on a set of world-histories sharing common pasts that are alternatives to each other. These sets are identified by the *R* accessibility relation which takes an interval and two world-histories as arguments. The relation $R(i, h1, h2)$ is true iff world-histories $h1$ and $h2$ are possible with respect to each other and share a common past through the end of interval i .

The interpretation of the statement $\lceil (\text{INEV } t_{int} \ P) \rceil$ at world-history h is true iff P is true in all the world-histories that are possible with respect to h and share a common past with h through the end of the interval denoted by t_{int} . Formally, the interpretation of $\lceil (\text{INEV } t_{int} \ P) \rceil$ is given by:

For all wffs of the form $\lceil (\text{INEV } t_{int} \ P) \rceil$ and world-histories (h),
 $V_s(\lceil (\text{INEV } t_{int} \ P) \rceil, h) = \text{TRUE}$ iff
 for all world-histories ($h2$) if $R(V_t(t_{int}), h, h2)$ then $V_s(P, h2) = \text{TRUE}$

Restrictions are placed on the *R* accessibility relation to rule out models that conflict with the intuitive meaning given to *R*. To begin with, the truth value of the relation $R(i, h1, h2)$ depends on the "end of interval i ". That is, if two intervals $i1$ and $i2$ start at different times but end together, then $R(i1, h1, h2)$ and $R(i2, h1, h2)$ will have the same truth values for all world-histories $h1$ and $h2$.¹ Formally, this

relation is captured by the following constraint on R :

R0)

For all world-histories ($h1$ and $h2$) and intervals ($i1$ and $i2$),
if ENDS-SAME($i1, i2$) then $R(i1, h1, h2)$ iff $R(i2, h1, h2)$

where ENDS-SAME($i1, i2$) means that intervals $i1$ and $i2$ end at the same time.
Its definition in terms of MTS is given in appendix B

For a fixed interval, the R relation is an equivalence relation. Thus, we impose the following constraints on R :

R1) R is reflexive

For every world-histories (h) and interval (i),
 $R(i, h, h)$

R2) R is symmetric

For all world-histories ($h1$ and $h2$) and intervals (i),
if $R(i, h1, h2)$ then $R(i, h2, h1)$

R3) R is transitive

For all world-histories ($h1$, $h2$, and $h3$) and intervals (i),
if $R(i, h1, h2)$ and $R(i, h2, h3)$ then $R(i, h1, h3)$

For all world-histories, if two world-histories share a common past through the end of interval $i2$, then they share a common past through the end of any interval that ends before interval i . This is formally captured by the following constraint:

R4)

For all world-histories ($h1$ and $h2$) and intervals ($i1$ and $i2$),
if ENDS-BEF($i1, i2$) and $R(i2, h1, h2)$ then $R(i1, h1, h2)$

where ENDS-BEF($i1, i2$) means that $i1$ ends before $i2$. Its definition in terms of MTS is given in appendix B.

The R accessibility relation must also be compatible with the specifications in the model describing the properties that hold at different times and the event that occur at different times at each world-history. Any two world-histories $h1$ and $h2$ that are related by R at time i must agree on all properties holding over intervals that end before or at the same time as i . Similarly, $h1$ and $h2$ must agree on all event

¹ If we had objects in the model corresponding to points, the R relation would more naturally take a point as its first argument. This is precisely what Van Fraassen [Van Fraassen 80] and Haas [Haas 85] do in providing models for branching time logics with a time line composed of points.

occurring over intervals that end before or at the same time as i . These restrictions are captured by the following constraints:

R5)

For all world-histories ($h1$ and $h2$), properties (pr),
and intervals ($i1$ and $i2$),
if (ENDS-BEF($i1, i2$) or ENDS-SAME($i1, i2$)) and $R(i2, h1, h2)$ then
 $\langle i1, h1 \rangle \in pr$ iff $\langle i1, h2 \rangle \in pr$

R6)

For all world-histories ($h1$ and $h2$), events (ev),
and intervals ($i1$ and $i2$),
if (ENDS-BEF($i1, i2$) or ENDS-SAME($i1, i2$)) and $R(i2, h1, h2)$ then
 $\langle i1, h1 \rangle \in ev$ iff $\langle i1, h2 \rangle \in ev$

Constraints $R5$ and $R6$, put together, specify that if two world-histories are R related at time i , then they agree on all properties that hold and events that occur at all times up until the end of i .² They do not, however, specify the converse of this relation which is given by: "if two world-histories $h1$ and $h2$ agree on all properties and events up until the end of interval i , then $R(i, h1, h2)$ must hold". There are two reasons why we would not want to impose such a constraint. To begin with, two world-histories might share a common past but still not be possible with respect to each other; thus, they should not be R related.

The second reason stems from the fact that the converse relation only mentions events that end before or at the same time as i . To conclude that two world-histories share a common past through i , it might be necessary to determine whether they agree on events that are occurring during i , but complete at a later time. Consider such an event ev that occurs in $h1$ during $i2$. Assume that interval $i2$ starts at the same time as i but finishes at a later time (in Allen's terminology we would simply say that i starts $i2$). Now, in order to conclude that $h1$ and $h2$ agree through the end of i , $h2$ must agree on ev 's behavior during interval i . Thus, we would have to compute the "prefixes over i " for all events that are in progress during i . We have not, however, formally introduced the notion of "prefixes" in our model.

3.2.3. Plan Instances and Goldman's Theory of Actions

In section 2.2, we described a plan instance as being a set of events at specified times, to be brought about by a particular execution. Associated with each plan instance are two sets. The first set consists of ordered pairs formed from events and

² We also get the constraint that two world-histories that are R related at i , agree on all plan instance that occur before the end of i . This relation follows from $R6$ and a relation imposed on plan instances (see 3.2.3) saying that a plan instance occurrence is associated with an event occurrence

intervals indicating the events at specified times to be brought about by executing the plan instance. The second set consists of ordered pairs formed from objects, which we call **basic actions (functions)**, and intervals. This set of pairs indicates the steps constituting the particular execution associated with the plan instance. A plan instance is uniquely identified by these two sets. Therefore, in our model, a plan instance is taken to be an ordered pair of the form $\langle ei\text{-set}, bai\text{-set} \rangle$ where *ei-set* is a non-empty set containing ordered pairs formed from events and intervals, and *bai-set* is a non-empty finite set containing ordered pairs formed from basic actions and intervals. We will refer to ordered pairs of the form $\langle ev, i \rangle$ as **event instances** and ordered pairs of the form $\langle ba, i \rangle$ as **basic action instances**. We will also designate basic action instances using the syntax $ba@i$.

The term "basic action" is adopted from Goldman [Goldman 70] who presents a theory of human action in which basic actions play a central role. In a non-formal sense, our conception of basic actions is very similar to that of Goldman's conception. Formally, however, the treatments differ since we model basic actions as functions (to be explained in detail in section 3.2.4) while Goldman treats them simply as objects.

We will also refer to two other concepts examined by Goldman: the *generation* relation and *standard conditions*. In the Artificial Intelligence literature, Pollack [Pollack 86] has also made use of Goldman's theory in a way compatible with our approach (see section 5.2) in her theory of plan recognition with invalid queries. Before continuing the discussion on plan instances, we give a brief description of Goldman's analysis of generation, basic actions, and standard conditions.

The Generation Relation, Basic Actions, and Standard Conditions

A central notion in Goldman's theory of action is a relation on action tokens (an action type at a particular time performed by a particular agent) which he calls *generation*. Roughly put, the relation "act token $a1$ generates act token $a2$ " holds whenever it is appropriate to say that $a2$ can be done by doing $a1$. Associated with each generation relation is a set of conditions C_i^* which are the necessary and sufficient conditions under which the occurrence of the generator leads to the occurrence of the token being generated. A typical example of this relation is given by: "flipping the light switch at time $t1$ " generates "turning on the light at time $t1$ " under the conditions that the filament is not burnt out, the wiring is not faulty, etc. Generation is a transitive, irreflexive relation that arranges the set of action tokens into linear chains. For example, "letting Beth in the room at time $t1$ " is generated by "opening the door at time $t1$ " which is generated by "turning the doorknob and pushing at time $t1$ ", etc.

Goldman also introduces the concept of a *basic action* (types). The two essential features of basic actions are that they can be "done at will" and that they are primitive in the sense that every non-basic action token is generated by some basic action token (or some set of basic action tokens executed together) and there is no action token more primitive that generates a basic action token. Doing any action

boils down to performing a collection of basic action tokens under the appropriate conditions. Goldman equated basic actions with the performance of body movements such as "raising the left arm", "taking a step with the right leg", etc. He did not go to a finer analysis to say that "moving one's arm" is generated by "sending nerve impulses to the muscles in arm". He argued that events such as sending nerve impulses should not be considered acts since an agent does not have fine enough control to send nerve impulses to particular muscles and thus cannot do this at will.

One problem that Goldman had to get around was that there are always conditions that can preclude any basic action from occurring, and thus it is not technically correct to say that they can be done at will. For example, "lifting one's arm" cannot be performed if the arm is being held down by another (stronger) agent, or if the arm is being tied down, etc.. For this reason, he introduced the notion of **standard conditions** which are conditions that must hold in order for the basic action to take place. If basic action $a1$'s standard conditions hold at time $t1$ then $a1$ can be done at will at time $t1$. As an example, the standard conditions for "lifting one's arm" are that the arm is not being held by another agent, the arm is not being tied down, etc. He took standard conditions to be external conditions, that is, conditions out of the agent's control. Thus, "internal" conditions, such as the arm is not paralyzed, are not considered as part of the standard conditions for lifting one's arm. In this case, Goldman would say that one loses the basic action "lifting one's arm" during the times when the arm is paralyzed.

Back to Plan instances and Basic Action Instances

In our theory, our conception of basic actions and their standard conditions is more general than Goldman's formulization. We treat a basic action instance as an action that is at the finest level of detail that is appropriate for the domain under consideration. They are not necessarily collections of body movements at specified times. For example, in modeling a game of chess, we might take our basic action instances to be simple chess moves such as moving the queen from Q1 to Q3 at a particular time. The standard conditions would be that the move is legal by the rules of chess. When reasoning about winning the chess game, there is no benefit in looking more closely at a chess move and saying that it is generated by the arm movement that physically moves the piece.

Formally, a basic action is taken to be a function from $I \times H$ to 2^H . We wait until section 3.2.4 to explain why these objects are characterized this way. In section 3.2.4, we will also see that if the standard conditions for a basic action at a particular time (i.e. a basic action instance) does not hold at some world-history h , then the basic action instance is treated as a "no-op", an action that does nothing if it were to be executed in h .

Associated with each basic action is the event type that is exemplified by its execution. This is specified by the following function which is given as part of a model:

BAEV refers to a function from basic actions to event types that maps a basic action to the event type associated with it

By linking a basic action to an event type we indirectly specifying the intervals at each world-history over which the basic action occurs. In particular, basic action *ba* occurs over interval *i* in world-history *h* iff $\langle i, h \rangle$ belongs to $\text{BAEV}(ba)$.

Members of *PI* (i.e. the set of plan instances) implicitly encode a relation similar to Goldman's generation relation. In any world-history where all event instances contained in *ei-set* occur at their specified times and all basic action instances contained in *bai-set* occur at their specified times, we say that *bai-set* generates *ei-set*. Aside from some superficial differences, we can look at the generation relation captured by the plan instances as agreeing with Goldman's relation for the special case where the generator is basic. Goldman took generation to be a relation between two act instances (which are roughly equivalent to our event instances), while generation in our case is a relation between a set of event instances and a set of basic action instances. The fact that generators in our case are basic action instances instead of event tokens associated with basic actions should not be counted as a difference. This is because Goldman did not make the distinction, that we are making, between basic actions and event types associated with basic actions. Secondly, the fact that generation in our models relates sets of instances instead of single instances should not be counted as a difference. This is because Goldman provided for "compound acts" which correspond to a collection of act types done together. In effect this allows him to handle the cases that we can handle where a set consisting of a single event instance is generated by a collection of basic action instances, and the case where a collection of event instances is generated by a set consisting of a single basic action instance. Lastly, we must point out a slight difference in our usage of the term "generates" from that of Goldman. Goldman took generation to be irreflexive, while we essentially have the case where a basic action generates itself since we allow plan instances of the form: $\langle \{ \langle \text{BAEV}(ba), i \rangle \}, \{ \langle ba, i \rangle \} \rangle$.

In Goldman's theory, if action token *a1* generates token *a2*, then *a1* and *a2* must have the same time of occurrence. In our theory, we impose a less stringent relation between the times associated with a set of event instances *ei-set* and the set of basic action instances *bai-set* that generates it. If $\langle ei\text{-set}, bai\text{-set} \rangle$ is a plan instance, then *ei-set* and *bai-set* must start at the same time, and *ei-set* must end at the same time or latter than *bai-set*. By relaxing their temporal relation in this way, we can model plan instances such as "breaking the vase during *i1* " performed by "knocking the vase off the table during *i2*" To model this plan instance, we would want *i2* to end before *i1* because the action of knocking the vase off the table finishes before the vase actually breaks.

The times associated with the basic action instance set and event instance set refer to the smallest intervals in which all their components occur. We will use **COVER** to designate the function that takes a set of intervals *i-set* and yields the smallest interval that contains each interval belonging to *i-set*. In appendix B, we give a definition of **COVER** in terms of the *MTS* relation. The time associated with a

set of event instances $ei\text{-}set$ is given by: $COVER(\{i \mid \langle ev, i \rangle \in ei\text{-}set\})$ and the time associated with a set of basic action instances $bai\text{-}set$ is given by: $COVER(\{i \mid \langle ba, i \rangle \in bai\text{-}set\})$. For any plan instance $\langle ei\text{-}set, bai\text{-}set \rangle$, the temporal relation between $ei\text{-}set$ and $bai\text{-}set$ is given by:

PI1)

For all plan instances $\langle ei\text{-}set, bai\text{-}set \rangle$,
 $STARTS(bai\text{-}time, ei\text{-}time)$ or $EQUAL(bai\text{-}time, ei\text{-}time)$

where $ei\text{-}time =_{\text{def}} COVER(\{i \mid \langle ev, i \rangle \in ei\text{-}set\})$,
 $bai\text{-}time =_{\text{def}} COVER(\{i \mid \langle ba, i \rangle \in bai\text{-}set\})$

PI1 says that for all plan instances $\langle ei\text{-}set, bai\text{-}set \rangle$, either the time associated with $bai\text{-}set$ starts at the same time but finishes before the time associated with $ei\text{-}set$, or the two associated times are equals

In our language, the function term $\lceil (TIME\text{-}OF\ t_{pi}) \rceil$, denotes the time associated with the plan instance denoted by t_{pi} . This time is simply taken to be the time that is associated with the plan instance's event instance set. Thus, the interpretation of $\lceil (TIME\text{-}OF\ t_{pi}) \rceil$ is given by:

For all plan instance terms (t_{pi}) ,
 $V_t(\lceil (TIME\text{-}OF\ t_{pi}) \rceil) = COVER(\{i \mid \langle ev, i \rangle \in V_t(t_{pi})|_1\})$

where $V_t(t_{pi})|_1$ is the first element of the ordered pair $V_t(t_{pi})$

We say that the plan instance $\langle ei\text{-}set, bai\text{-}set \rangle$ occurs iff all the event instances belonging to $ei\text{-}set$ occur and they are brought about by the occurrence of all the basic action instance belonging to $bai\text{-}set$. This is equivalent to saying that plan instance occurs $\langle ei\text{-}set, bai\text{-}set \rangle$ iff all the event instances belonging to $ei\text{-}set$ occur and all the basic action instance belonging to $bai\text{-}set$ occur. The reason for this equivalence is that if the ordered pair $\langle ei\text{-}set, bai\text{-}set \rangle$ forms a plan instance, then $bai\text{-}set$ generates $ei\text{-}set$. Hence, if all the elements of both $ei\text{-}set$ and $bai\text{-}set$ occur, one could say that $bai\text{-}set$ brought about $ei\text{-}set$.

In our language, we are using the sentence $\lceil (OCC\ t_{pi}) \rceil$ to mean that the plan instance denoted by t_{pi} occurs. Therefore, the interpretation of $\lceil (OCC\ t_{pi}) \rceil$ is given by:

For all wffs of the form $\lceil (OCC\ t_{pi}) \rceil$ and all world-histories (h) ,

$V_s(\lceil (OCC\ t_{pi}) \rceil, h) = \text{TRUE}$ iff

for all events (ev) , intervals (i) and basic actions (ba)

if $\langle ev, i \rangle \in V_t(t_{pi})|_1$, then $\langle i, h \rangle \in ev$, and

if $\langle ba, i \rangle \in V_t(t_{pi})|_2$, then $\langle i, h \rangle \in BAEV(ba)$

where $V_t(t_{pi})|_1$ is the first element of the ordered pair $V_t(t_{pi})$, and $V_t(t_{pi})|_2$ is the second element of $V_t(t_{pi})$

In the above interpretation, "if $\langle ev, i \rangle \in V_t(t_{pi})|_1$, then $\langle i, h \rangle \in ev$ " is true iff all the event instances belonging to $V_t(t_{pi})|_1$ occur, where $V_t(t_{pi})|_1$ is the set of event

instances associated with the plan instance denoted by t_{pi} . Similarly, "if $\langle ba, i \rangle \in V_t(t_{pi})|_2$, then $\langle i, h \rangle \in BAEV(ba)$ " is true iff all the basic action instances belonging to $V_t(t_{pi})|_2$ occur, where $V_t(t_{pi})|_2$ is the set of basic action instances associated with the plan instance denoted by t_{pi} .

Finally, we give an interpretation to the function term $\lceil (COMP\ t_{pi1}\ t_{pi2}) \rceil$ which denotes the composition of the two plan instances denoted by t_{pi1} and t_{pi2} . The composition of plan instances $\langle ei-set1, bai-set1 \rangle$ and $\langle ei-set2, bai-set2 \rangle$ is taken to be $\langle ei-set1 \cup ei-set2, bai-set1 \cup bai-set2 \rangle$ and, thus, the interpretation of $\lceil (COMP\ t_{pi1}\ t_{pi2}) \rceil$ is given by:

For all plan instance terms (t_{pi1} and t_{pi2}),

$$V_t(\lceil (COMP\ t_{pi1}\ t_{pi2}) \rceil) = \langle V_t(t_{pi1})|_1 \cup V_t(t_{pi2})|_1, V_t(t_{pi1})|_2 \cup V_t(t_{pi2})|_2 \rangle$$

This formalization of composition gives the desirable result that a composition occurs iff both its parts occur, and a composition is generated by the generators of both its parts taken together. We also place the constraint that the set of plan instances are closed under composition:

PI2)

For all plan instances ($\langle ei-set1, bai-set1 \rangle$ and $\langle ei-set2, bai-set2 \rangle$),
 $\langle ei-set1 \cup ei-set2, bai-set1 \cup bai-set2 \rangle \in PI$

3.2.4. Basic Actions and the Interpretation of IFTRIED

The effects produced by executing some basic action at a specified time, i.e. a basic action instance, depends on the environment in which it is executed. In a given environment, there are properties and events that a basic action instance affects and ones that it does not affect. Thus, to model a basic action instance *bai*, we must specify for each environment, the conditions that the execution of *bai* would affect and the conditions that the execution would have not affect. In our theory, we equate "environment" with world-history. To determine what would happen if basic action *bai* were to be executed in world-history *h*, one could "minimally revise" *h* to provide for the execution of *bai*. If *bai* happens to occur in world-history *h*, then no revision is necessary. Otherwise, we get a revised world-history that differs from *h* only on conditions that are affected (directly or indirectly) by executing *bai*. Thus, *h* and a revised world-history will agree on all conditions that are not influenced by executing *bai*. This includes all conditions that are out of the agent's control along with properties that hold and events that occur prior to *bai*'s time of execution.

There may not be a unique way to minimally revise a world-history to provide for the execution of basic action instance. A case of this is where a basic action has non-deterministic outcomes, an example being the rolling of a fair die. In this case, there will be (at least) six ways that a world-history can be revised to account for the rolling of the die. We will therefore speak of the set of world-histories that minimally differ from a world-history on the account of executing some basic action at some specified time.

In agreement with the analysis above, a basic action is taken to be a functions from $I \times H$ to 2^H that serve to specify, for each interval *i* and world-history *h*, the effects that would be produced if the basic action were to be executed during *i* in world-history *h*. The members of $ba(i,h)$ constitute all the world-histories that minimally differ from *h* on the account of executing *ba* during interval *i*. As we will shortly discuss, $ba(i,h)$ may contain a world-history in which *ba* does not occur during *i*.

The Interpretation of IFTRIED and the Execution of a Set of Basic Action Instances

In our language, the *IFTRIED* modality is used to make statements about what the attempt of a plan instance affects and what it does not affect. As we have discussed in the last section, a set of basic action instances is associated with each plan instance. Saying that a plan instance is to be attempted is taken to mean that the set of basic action instances associated with it are all to be executed together. Thus, the interpretation of $\langle (IFTRIED t_{pi} P) \rangle$ depends on what would happen if the set of basic action instances associated with t_{pi} were all to be executed together. Therefore, not only must we be able to determine the effect of executing a basic action instance alone, but also the effect of executing a set of basic action instances together. In particular, for every world-history *h* and non-empty set *bai-set*

consisting of basic action instances, the model must specify the world-histories that minimally differ from h on the account of executing all the basic actions belonging to $bai\text{-}set$. As we will see, the effects of executing a collection of basic action instances together is computed from the individual basic action (functions). This will be discussed in detail in section 3.25. We will use F_{cl} to refer to the function that takes a set of basic action instances $bai\text{-}set$ and a world-history h as arguments and yields the world-histories that minimally differ from h on the account of executing all the basic action instances belonging to $bai\text{-}set$. In the case where $bai\text{-}set$ consists of only one member, F_{cl} is simply defined as follows:

$$F_{cl}(\{ \langle ba, i \rangle \}, h) =_{\text{def}} ba(i, h)$$

The interpretation of $\lceil (IFTRIED\ t_{pi}\ P) \rceil$ is succinctly given in terms of F_{cl} :

IFT-INT)

For all wffs of the form $\lceil (IFTRIED\ t_{pi}\ P) \rceil$ and world-histories (h) ,

$V_s(\lceil (IFTRIED\ t_{pi}\ P) \rceil, h) = \text{TRUE}$ iff

for all world-histories $(h2)$ if $h2 \in F_{cl}(V_t(t_{pi})|_2, h)$ then $V_s(P, h2) = \text{TRUE}$

where $V_t(t_{pi})|_2$ refers to the second element of the ordered pair $V_t(t_{pi})$, this being the set of basic action instances associated with the plan instance term t_{pi}

IFT-INT can be read as saying: the statement $\lceil (IFTRIED\ t_{pi}\ P) \rceil$ is true at world-history h iff P is true in all the world-histories that minimally differ from h on the account of executing all the basic action instances associated with the plan instance denoted by t_{pi} . This interpretation of $\lceil (IFTRIED\ t_{pi}\ P) \rceil$ would vacuously hold at h if there did not exist any world-histories that belonged to $F_{cl}(V_t(t_{pi})|_1, h)$, i.e. $F_{cl}(V_t(t_{pi})|_1, h)$ equaled the null set. To preclude this condition, the basic action functions will be constrained so that none of them yield the null set for any arguments. Consequently, F_{cl} , which is defined in terms of these functions, will not yield the null set. Formally, this constraint on basic actions is given by:

BA0)

For every basic action (ba) , interval (i) , and world-history (h) ,

$ba(i, h) = \emptyset$

The interpretation of $\lceil (IFTRIED\ t_{pi}\ P) \rceil$ bears a strong resemblance to the interpretation of conditionals developed by Stalnaker [Stalnaker 68] and Lewis [Lewis 73]. In fact, our approach was modeled after these treatments. The function F_{cl} (which is defined in terms of the basic action functions) can be seen as a variant of the accessibility relations that are found in Stalnaker's and Lewis' semantic models. After presenting our analysis of the basic action functions and their compositions (section 3.2.5), we describe both Stalnaker's and Lewis' theories in section 3.2.6.

Minimally Differing World-Histories

We now go into detail discussing what we mean by saying that two world-histories minimally differ on the account of the execution of a basic action instance. Constraints will be imposed on the basic action functions to rule out models that do not meet our intuitions. In this presentation, a term of the form $ba@i$ will be used to refer to the basic action instance corresponding to the basic action ba executed during time i . We will use the phrase "*differ solely on the account of $ba@i$* " synonymously with "minimally differ on the account of the execution of $ba@i$ ". We will also use the term "*closest world history*" to mean "minimally differing world-history", this terminology being a vestige of Lewis' and Stalnaker's theories.

We begin with the trivial case where the basic action instance under consideration occurs in the world-history that we want to revise. In this case, no revision is needed. If a basic action instance $ba@i$ occurs in world-history h , then there is only one world-history that differs solely on the account of executing $ba@i$, and this world-history is h . We capture this property with the following constraint which we impose on our models:

BA1)

For every basic action(ba), interval (i), and world-history (h),
if $\langle i, h \rangle \in \text{BAEV}(ba)$ then $ba(i, h) = \{h\}$

If $ba@i$'s standard conditions do not hold in h and thus necessarily $ba@i$ does not occur in h , we equate $ba(i, h)$ with $\{h\}$. In effect, we are treating $ba@i$ at a world-history where its standard conditions do not hold as a "no-op"; it maps a world-history back into itself. This situation, where $ba@i$'s standard conditions do not hold in h , is treated differently from the cases where either $ba@i$ occurs in h or $ba@i$'s standard conditions hold at h in that $ba@i$ does not occur in the world-history belonging to $ba(i, h)$.

The reason for treating the lack of standard conditions in this special manner is because we want to restrict the basic action functions so that if h_2 belongs to $ba(i, h)$, then h and h_2 agree on all conditions that are not affected (directly or indirectly) by the execution of ba at time i . This restriction would be violated, if we allowed models where the following conditions were all true: i) h_2 belongs to $ba(i, h)$, ii) $ba@i$'s standard conditions do not hold in h , and iii) $ba@i$ occurs in h_2 . The reasoning for this goes as follows. A basic action instance's standard conditions are conditions that are out of the agent's control and/or hold prior to the basic action instance's time of occurrence; they clearly are not conditions that are brought about by the basic action instance's execution. Secondly, if basic action instance $ba@i$ occurs in world-history h_2 during i , then $ba@i$'s standard conditions necessarily hold in h_2 . Thus, if we had ii) $ba@i$'s standard conditions do not hold in h , and iii) $ba@i$ occurs in h_2 as above, then h and h_2 would disagree on conditions that are not affected by the execution of $ba@i$, i.e. $ba@i$'s standard conditions. Thus, if i) h_2 belongs to $ba(i, h)$ were also true, we would contradict our assumption that relation i) entails that h_2 and h cannot differ on conditions that are not affected by executing $ba@i$.

The lack of standard conditions is intended to model situations where it is undefined what the performance of a basic action instance would do if executed. This refers to the execution of *impossible* basic action instances such as "move and do not move at the same time i " or illegal moves when modeling a board game where only legal moves are possible considerations¹. Thus, we treat the lack of standard conditions in this simple fashion. This treatment, however, does not preclude the modeling of failed actions. These can be modeled by a plan instance whose associated basic action instance (or set of basic action instances) is (successfully) executed but the set of event instances associated with the plan instance do not all occur.

We now turn to the interesting case where $ba@i$'s standard conditions hold in h , but $ba@i$ does not occur in h . In this circumstance, we refer to the relation " $h2 \in ba(i, h)$ " by saying $h2$ properly differs from h solely on the account of $ba@i$. We use the term "properly" to imply that it is proper to say that $h2$ and h differ.

To begin with, if $ba@i$'s standard conditions hold in h , we assume that if $ba@i$ were to be executed in h , we would arrive at world-histories in which $ba@i$ occurs. In other words, if $h2$ properly differs from h solely on the account of $ba@i$, then $ba@i$ occurs in $h2$. Thus, we have the following constraint on our models:

BA2)

For every basic action(ba), interval (i), and world-histories (h and $h2$),
if $h = h2$ and $h2 \in ba(i, h)$ then $\langle i, h2 \rangle \in BAEV(ba)$

Our intuitive notion of " $h2$ properly differs from h solely on the account of $ba@i$ " is best explained by describing a revision process from h to $h2$; this does not mean, however, that the revision process is explicitly given in the model; the model only gives the result of the revision process as captured by the basic action functions.

- 1) (In going from h to $h2$) revise the status of $ba@i$
- 2) Revise the status of basic action instances that occur in h but physically cannot be done in conjunction with $ba@i$
- 3) Revise the status of properties and events that hold (occur) in h that are directly dependent on the non-occurrence of $ba@i$ and/or the occurrence of the basic action instances revised in step 2
- 4) Revise the status of the properties, events, and basic action instances that are directly dependent on the conditions changed in the previous revision; repeat

No other properties, events, or basic action instances are revised

¹ Alternatively, "move and do not move at the same time i " may refer to the composition of the two basic action instances "move at time i " and "do not move at time i ", both of which can be done separately, but not together. As we will see in section 3.2.5, the composition of two basic action instances they cannot be done together is treated as if its standard conditions do not hold.

The revision process is meant to capture that in going from h to h_2 , the only conditions that are revised are ones that are directly affected by $ba@i$'s occurrence, ones that are affected by conditions that are directly affected by $ba@i$, etc. The two world-histories agree on all other conditions. This includes all conditions that are out of the agent's control such as whether or not it is raining, when the bank closes and opens, etc. We also stipulate that h and h_2 must agree on all conditions that hold prior to interval i , $ba@i$'s time of execution. We will shortly describe how a constraint relating the basic action functions and R captures this restriction.

An example will help to clarify the revision process (See diagram 3.2-1). Consider the plan instance $\langle \{ \langle \text{init-BOOT}, i1 \rangle \}, \{ \langle \text{type-BOOT}, i1 \rangle \} \rangle$ where init-BOOT refers to the event "initiating the program BOOT" and type-BOOT refers to the basic action where the atom "BOOT" followed by a carriage return is typed by the agent. The occurrence of the basic action instance $\text{type-BOOT}@i1$ generates the event instance $\text{init-BOOT}@i1$ under the conditions that the computer is operational during time $i1$. Diagram 3.2-1 depicts the world-history h where $\text{type-BOOT}@i1$ occurs and the computer is operational during $i1$, and thus necessarily $\text{init-BOOT}@i1$ occurs. Furthermore, in h , the computer is operational during the larger interval $i\text{-all}$ and the initiation of BOOT leads to the occurrence where BOOT runs during interval $i2$.

Now, consider the plan instance $\langle \{ \langle \text{init-SCAN}, i1 \rangle \}, \{ \langle \text{type-SCAN}, i1 \rangle \} \rangle$ which can be executed during interval $i1$ instead of the plan instance described above. The event init-BOOT refers to the event "initiating the program SCAN" and type-BOOT refers to the basic action where the atom "SCAN" followed by a carriage return is typed by the agent (and whose standard conditions hold in h). The occurrence of the basic action instance $\text{type-SCAN}@i1$ generates the event instance $\text{init-SCAN}@i1$ under the conditions that the computer is operational during time $i1$. We now consider world-history h_2 which properly differs from h solely on the account of $\text{type-SCAN}@i1$. To begin with, $\text{type-BOOT}@i1$ does not occur in h_2 since it physically conflicts with typing "SCAN" during $i2$ and thus $\text{init-BOOT}@i1$ does not occur in h_2 since it is generated by $\text{type-BOOT}@i1$. It is also the case that BOOT does not occur during $i2$ in h_2 since this occurrence was (solely) caused by $\text{init-BOOT}@i1$'s occurrence in h . On the other hand, h and h_2 agree on the condition "the computer is operational during $i\text{-all}$ " since $\text{type-SCAN}@i1$ has no effect on the computer's operational status. Because (in h_2) $\text{type-SCAN}@i1$ occurs under the conditions "the computer is operation during $i1$ ", $\text{init-SCAN}@i1$ occurs causing the program SCAN to run during interval $i2$.

We must emphasize that in computing a world-history that minimally differs on the account of some basic action instance, all changes are initiated by conditions that physically cannot be done together. We do not take into account "intention conflicts". What we mean by "intention conflicts" is described by the the following example. Consider a world-history h , where at time i , the agent performs two basic action instances, "buy a tape-deck during i " and "buy a cassette tape during i ". Now, consider a world-history h_2 that properly differs from h solely on the account of "buy a turntable during i ". If we assume that the agent only has enough money to buy either a tape deck or record player, but not both, "buy a turntable during i "

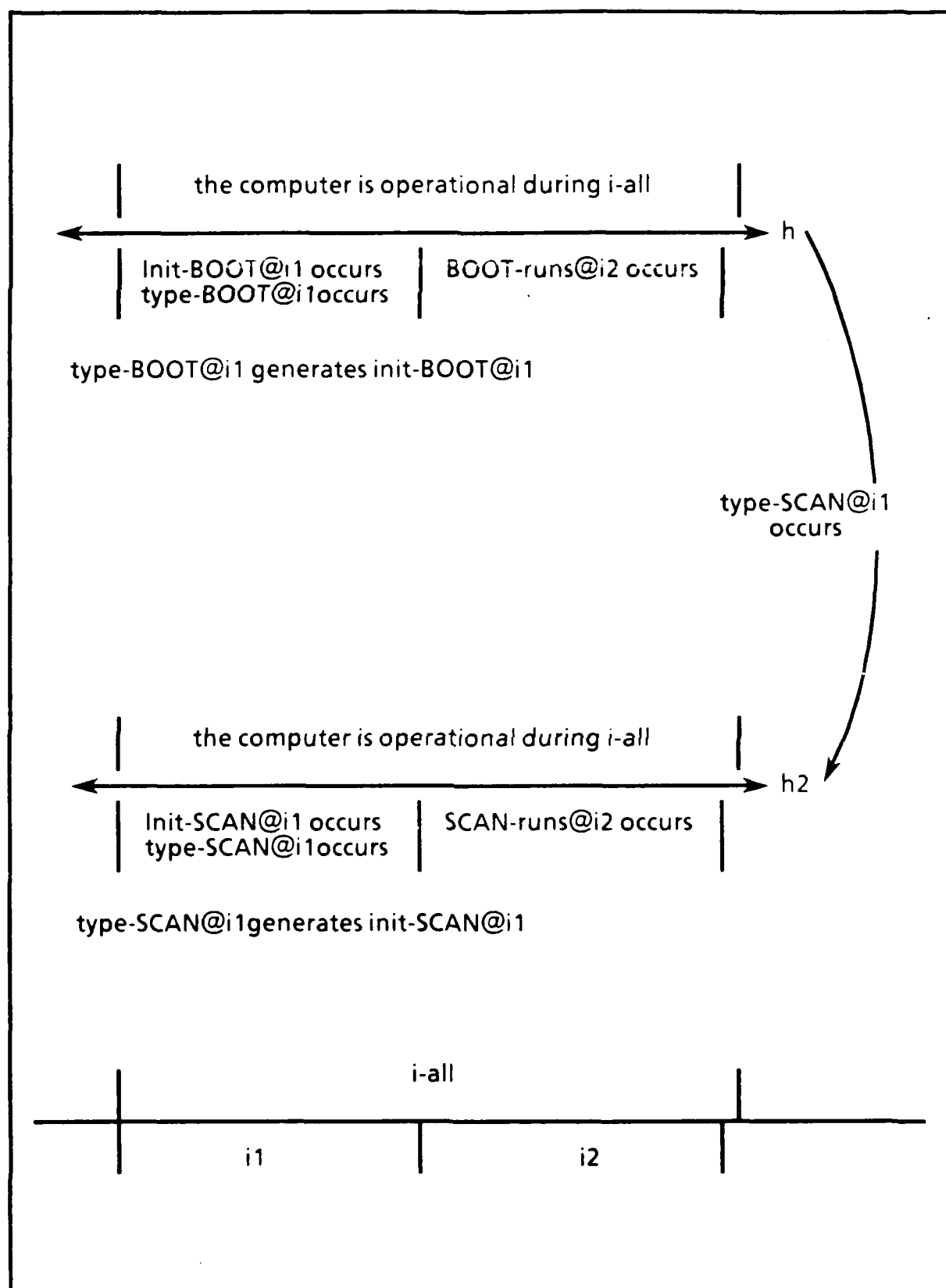


Diagram 3.2-1

physically conflicts with "buy a tape-deck during i ". Thus, in world-history $h2$, "buy a turntable during i " occurs, but "buy a tape-deck during i " does not. On the other hand, buying a turntable does not physically conflict with buying a cassette tape (assuming the cost of the turntable is less than or equal to that of a tape-deck). Therefore, according to our conception of minimal revision, one does not revise the status of "buy a cassette tape during i " in computing $h2$. Thus, $h2$ is a world-history in which a turntable is bought along with a cassette tape (apparently for no reason). If, however, our notion of minimal revision took into account intention conflicts, one would cast out "buy a cassette tape during i " since "buy a tape-deck during i " had to be cast out (assuming the agent did not have a second reason, such as another tape deck at home, for buying a cassette tape). In general, one would cast out any occurrence that was done solely for the reason or in preparation for another occurrence which had to be cast out.

We have implicitly ruled out the treatment of intention conflicts by adopting the following constraint: if h and $h2$ differ solely on the account of $ba@i$, then h and $h2$ share a common past up until the beginning of $ba@i$'s execution time. If we took into account intention conflicts, this constraint could not be imposed as the following example demonstrates.

Consider a world-history h where at time $i1$ the agent buys a present, and at a later time $i2$, the agent goes to a party bringing the present as a gift. Now, consider a world-history $h2$ that properly differs from h solely on the account of the basic action instance "the agent goes to the concert during time $i2$ ". It is clearly the case that "the agent goes to the party during $i2$ " does not occur in $h2$ since this physically conflicts with going to the concert during time $i2$. If we took into account intention conflicts, we would also have to cast out "the agent buys a present during $i1$ " since this is done solely for the reason of going to the party. Thus, we get two world-histories that solely differ on the account of "the agent goes to the concert during time $i2$ ", but differ on "the agent buys a present during $i1$ " which is a condition that holds before $i2$.

Treating intention conflicts would be an interesting project, although quite complicated. We have ignored this issue for simplification. Our hope is that one could build a theory of intentions and intention conflicts on top of the theory put forth here. Whether or not one treats intention conflicts, however, one must still treat physical conflicts. Physical conflicts are in sense prior to intention conflicts. Any intention conflict is at some stage initiated by some physical conflict.

Multiple Closest World-Histories

In the beginning of this section, we mentioned that there may not be a unique world-history that minimally differs on the account of some basic action instance. One case of this, which we previously mentioned, is where a basic action instance has non-deterministic outcomes. In this situation, there will be at least one world-history in $ba(i,h)$ corresponding to each non-deterministic outcome (in the case

where $ba@i$ does not occur in h while its standard conditions hold). There are other reasons why multiple closest world-histories may come about.

A second case is where there is not a unique resolution as to which basic action instances cannot be done in conjunction with a basic action instance that must be revised so that it occurs. The simplest example is where the agent can do at most two things at once. Consider three basic action instances $ba1@i$, $ba2@i$, and $ba3@i$ all of which have the same time of occurrence (i.e. i). Let h be a world-history where both $ba1@i$ and $ba2@i$ occur. There will be two closest world-histories to h where $ba3@i$ occurs, one where both $ba1@i$ and $ba3@i$ occur, and the other where $ba2@i$ and $ba3@i$ occur.

A third example of multiple closest world-histories is where the effects of a basic action instance enable some other event or (the agent's own actions) to be completed in two or more ways. A simple example crops up when modeling a two player game where the agent is one of the competitors (See diagram 3.2-2). Suppose that initially the agent has two possible moves $a1@i1$ and $a2@i1$ which are basic action instances with time of occurrence $i1$. At a subsequent time, say $i2$, the opponent is faced with the following options: If the agent plays $a1$ during $i1$, then the opponent is forced to make move $o1$ during $i2$. If the agent makes move $a2$ during $i1$, then the opponent has the choice of making move $o2$ or $o3$ during time $i2$. Let $h1$ be a world-history where the agent makes move $a1$ during $i1$ which is (necessarily) followed by the opponent's move $a1$. There are two candidates that are equally close to $h1$ where the basic action instance $a2@i1$ occurs. These are given in the diagram by $h2$ where $a2$ occurs followed by $o2$, and $h3$ where $a2$ occurs followed by $o3$.

Relating The Basic Action Functions to the R Relation

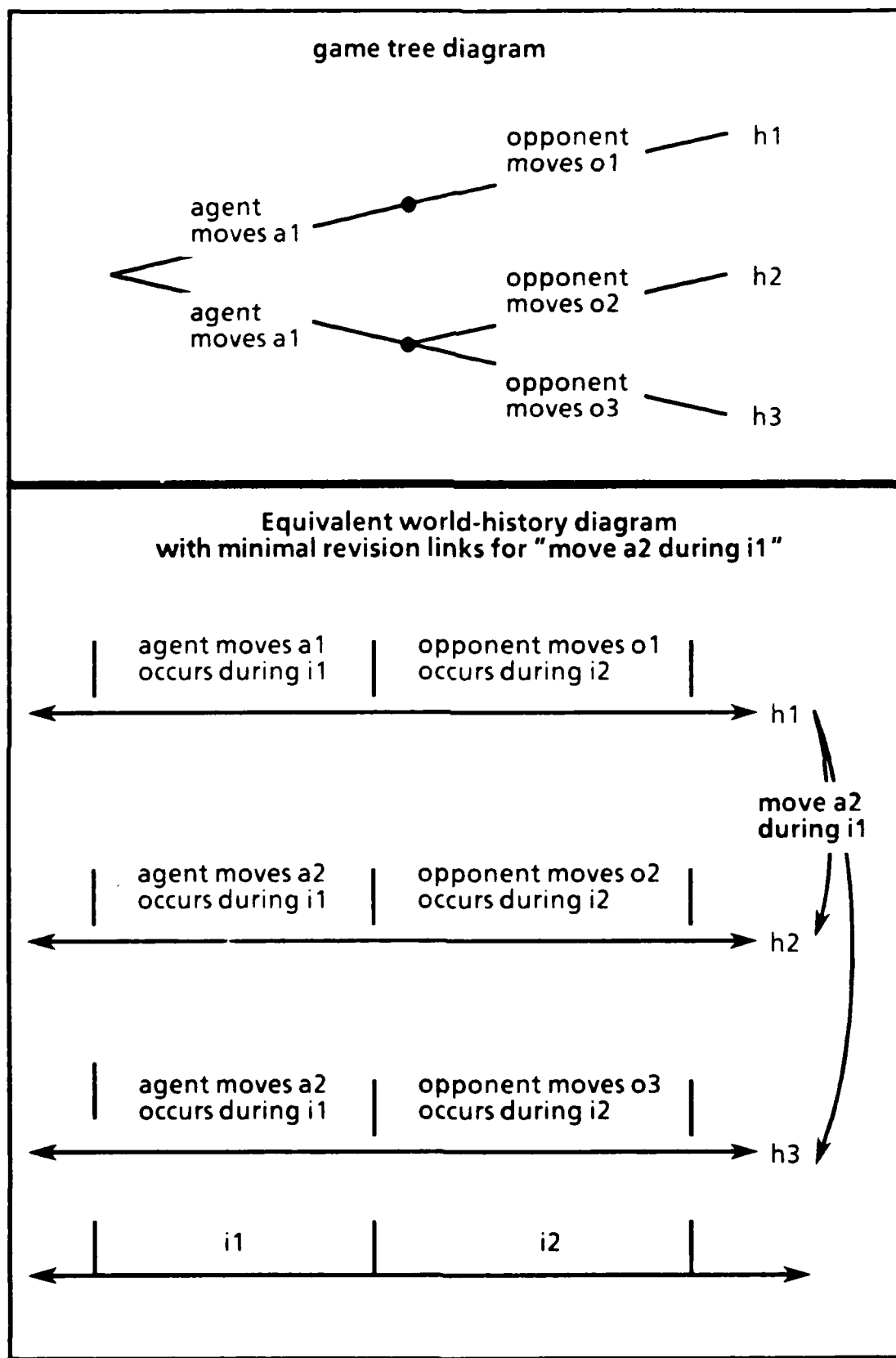
There are two constraints relating the basic action functions and R that we impose on our models. The first one entails the following restriction which we have previously mentioned: if $h2$ differs from h solely on the account of $ba@i$, then h and $h2$ share a common past up until the beginning interval i , $ba@i$'s execution time. Formally, this constraint is given by:

BA-R1)

For all world-histories (h and $h2$), basic actions (ba),
and intervals ($i0$ and i), if $h2 \in ba(i, h)$ and $MTS(i0, i)$ then $R(i0, h, h2)$

BA-R1 can be described as saying: If $h2$ differs from h solely on the account of $ba@i$ (i.e. $h2 \in ba(i, h)$), then for all intervals $i0$ that immediately precede (meet to the left) interval i ($ba@i$'s time of occurrence), $R(i0, h, h2)$ is true, entailing that h and $h2$ share a common past through the end of $i0$ and thus up until the beginning of i .

BA-R1, however, says something stronger than "if $h2$ differs from h solely on the account of $ba@i$, then h and $h2$ share a common past up until the beginning of i ". It can also be seen as a constraint on what world-histories must be possible with respect to each other as specified by R . As we have previously noted, the R



relationship only relates world-histories with common pasts that are possible with respect to each other. Thus, BA-R1 can be seen as narrowing the meaning of "possible" as captured by R by implying that under any circumstances (i.e. some world-history h), the result of applying any basic action instance leads to a circumstance that is possible with respect to h .

The second constraint relating the basic action functions and R stems from the observation: if $h2$ belongs to $ba(i,h)$, then the conditions that hold in $h2$ up until some time ir , are a function only of $ba@i$ and the conditions that hold in h up until ir . The conditions that hold in h after time ir , have no bearing on the conditions in $h2$ that hold up until ir . This principle is along the lines of "conditions at later times have no affect on condition at earlier times". The ramifications of this principle is that if two world-histories $h1$ and $h2$ are R related at time ir (and thus share a common past up until ir), the valuation of some basic action ba applied to $h1$ must be compatible with the valuation of ba applied to $h2$ in the following manner: for every member $hcl1$ of $ba(i,h1)$, there is at least one member of $ba(i,h2)$ that is R related to $hcl1$ at time ir . Formally, this is given by:

BA-R2)

For all world-histories ($h1$ and $h2$), intervals (i and ir) and basic actions (ba), if $R(ir,h1,h2)$ then for all world-histories ($hcl1$) such that $hcl1 \in ba(i,h1)$ there exists a world-history ($hcl2$) such that $hcl2 \in ba(i,h2)$ and $R(ir,hcl1,hcl2)$ are true

In the case where interval ir is prior to $ba@i$'s time of occurrence, BA-R2 logically follows from $\{BA-R1, R2, R3, R4\}$. Thus, we could have equivalently given a variant of BA-R2 where we restrict the relation between i and ir so that ir starts after the beginning of interval i .

We must now justify BA-R2 in the case where ir starts after the beginning of interval i . We will use the term the **prefix of $ba@i$ through ir** to refer to the portion of $ba@i$ that falls before the end of interval ir . If i ends before the end of ir , then "the prefix of $ba@i$ through ir " simply refers to $ba@i$ in its entirety. The justification of BA-R2 is given by arbitrarily picking two world-histories $h1$ and $h2$ that are R related at interval ir and then showing that the consequent in BA-R2 is true. We first consider the case where $ba@i$ has the property that for all world-histories h , there is a unique world-history that minimally differs from h on the account of $ba@i$. Thus, $ba@i$ acts in the same way given the same environment. We then consider the alternative case where there exists a world-history h in which $ba(i,h)$ has cardinality greater than 1.

CASE 1 $ba(i,h)$ has cardinality 1 for all world-histories h
and i starts before the end of interval ir

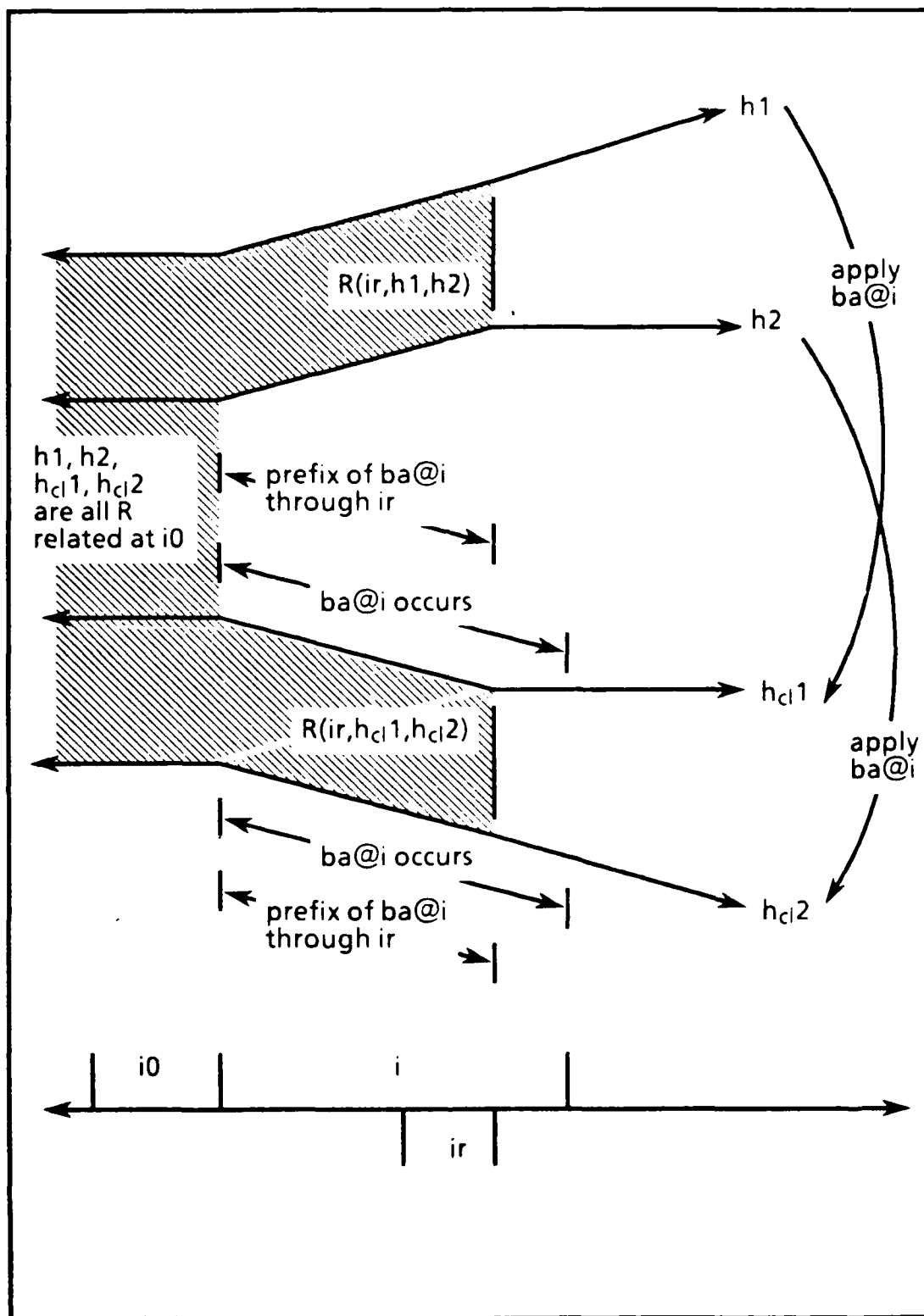
We assume that world-histories $h1$ and $h2$ are R related at interval ir . Let $hcl1$ be the world-history that differs from $h1$ solely on the account of $ba@i$, and let $hcl2$ be the world-history that differs from $h2$ solely on the account of $ba@i$ (See diagram 3.2-3). To justify BA-R2 in CASE 1, we need only show that $hcl1$ and $hcl2$ are R related at time ir .

The conditions that hold in $hcl1$ until the end of ir depend only on the prefix of $ba@i$ through ir and conditions that hold in $h1$ until the end of ir . Similarly, the conditions that hold in $hcl2$ until the end of ir depend only on the prefix of $ba@i$ through ir and conditions that hold in $h2$ until the end of ir . Since we are assuming that $ba@i$ acts in the same way in the same environment and that $h1$ and $h2$ share a common past through i , $hcl1$ and $hcl2$ must share a common past through ir . We assume without argument that $hcl1$ and $hcl2$ are possible with respect to each other at time ir . Thus, $hcl1$ and $hcl2$ are R related at time ir , thus justifying BA-R2 for CASE 1.

CASE 2 $ba@i$ may have multiple closest world-histories
and i starts before the end of interval ir

We work under the assumption that given the same environment, $ba@i$ results in the same set of possible behaviors. Once again we take world-histories $h1$ and $h2$ to be R related at interval ir . Let $p1$ refer to one of $ba@i$'s possible behaviors with respect to $h1$, and let $hcl1$ be the world-history that differs from $h1$ solely on the account of $p1$. Since we are assuming that for the same environment, $ba@i$ results in the same set of behaviors, one of $ba@i$'s behaviors with respect to $h2$ will coincide with $p1$'s behavior through time ir (Remember, $h1$ and $h2$ share a common past through ir). Let $p2$ refer to this behavior of $ba@i$ with respect to $h2$ and let $hcl2$ be the world-history that differs from $h2$ solely on the account of $p2$. To justify BA-R2 in CASE 2, we need only show that $hcl1$ and $hcl2$ are R related at time ir .

The conditions that hold in $hcl1$ until the end of ir depend only on the prefix of $p1$ through ir and conditions that hold in $h1$ until the end of ir . Similarly, the conditions that hold in $hcl2$ until the end of ir depend only on the prefix of $p2$ through ir and conditions that hold in $h2$ until the end of ir . Since $p1$'s behavior through time ir coincides with $p2$'s behavior through time ir , and world-histories $h1$ and $h2$ share a common past through ir , $hcl1$ and $hcl2$ must share a common past through ir . We assume without argument that $hcl1$ and $hcl2$ are possible with respect to each other at time ir . Thus, $hcl1$ and $hcl2$ are R related at time ir , thereby justifying BA-R2 for CASE 2.



3.2.5. The Composition of Basic Action Instances and Plan Instances

In this section, we examine the joint execution of a set of basic action instances. A "closeness" function is developed that takes a world-history and non-empty set of basic action instances as arguments and yields the set of closest world-histories differing from the world-history on the account of executing all the members in the basic action instance set. This function is defined in terms of the basic action functions. As we described in the last section, the interpretation of $\lceil (\text{IFTRIED } t_{pi} P) \rceil$ is based on such a function, which we have referred to by F_{cl} . $F_{cl}(\text{bai-set}, h)$ yields the set of closest world-histories to h in which all the elements in bai-set are executed. If the elements in bai-set cannot be jointly executed together in the environment given by h , $F_{cl}(\text{bai-set}, h)$ is set to $\{h\}$, treating bai-set as if its standard conditions do not hold at h . In the simple case where bai-set contains only one basic action instance, F_{cl} is equated with the closeness function associated with the basic action instance it contains. That is, for every basic action instance $ba@i$, $F_{cl}(\{ba@i\}, h)$ is defined to be $ba(i, h)$.

We are indirectly defining plan instances composition by defining basic action instance composition as given by the closest function applied to a set of basic action instances. Remember that the composition of plan instances $\langle \text{ei-set1}, \text{bai-set1} \rangle$ and $\langle \text{ei-set2}, \text{bai-set2} \rangle$ is defined as $\langle \text{ei-set1} \cup \text{ei-set2}, \text{bai-set1} \cup \text{bai-set2} \rangle$. Thus, the effect of attempting these two plan instances is given by F_{cl} applied to $\text{bai-set1} \cup \text{bai-set2}$.

We now discuss how F_{cl} applied to a set of basic action instances containing more than one element is defined in terms of the individual basic action functions. We begin by presenting the definition of $F_{cl}(\{ba@i\}, h)$ in the general case and presenting two related constraints that we place on our model. The remainder of the section is devoted to explaining and justifying this definition and the two constraints. To succinctly specify the definition of $F_{cl}(\{ba@i\}, h)$, we introduce the following notation and conventions. First of all, basic action instances will be designated by terms such as bai1 and bai2 , instead of using the $@$ -function notation that explicitly gives the basic action and interval associated with it. We will also use the notation $\text{bai}(h)$ to refer to the set of closest world-histories to h where basic action instance bai occurs.

For any basic action instance bai , $\text{bai}(h) =_{\text{def}} \text{ba}(i, h)$, where ba is the basic action associated with bai , and i is the corresponding interval

We will use $\text{OC}(\text{bai})$ to designate the set of world-histories in which basic action instance bai occurs in.

For any basic action instance bai , $\text{OC}(\text{ba}) =_{\text{def}} \{h \mid \langle i, h \rangle \text{BAEV}(\text{ba})\}$, where ba is the basic action associated with bai , and i is the interval associated with bai .

Lastly, we introduce a function combinator designated by ";" which combines any two functions f_x and f_y from H to 2^H to form $f_x;f_y$ which is also a function from H to 2^H . The definition of $f_x;f_y(h)$ for any world-history h is given by:

$$f_x;f_y(h) =_{\text{def}} \bigcup_{h_x \in f_x(h)} f_y(h_x)$$

This combinator function is associative, thus we can unambiguously use $f_1;f_2; \dots; f_n$ to designate the combination of two or more functions from H to 2^H . We call $f_1;f_2; \dots; f_n$ a **sequence function of FS** if set $FS = \{f_1, f_2, \dots, f_n\}$.

The definition of $F_{cl}(bai\text{-}set, h)$ and the two constraints we impose are given in figure 3.2-4. If the first constraint were not imposed, then the definition of *FCL-DEF* would be ill-formed. We explain and justify *FCL-DEF*, *BA-CMP1*, and *BA-*

FCL-DEF)

If there exists a sequence function of *bai-set* ($bai_1; bai_2; \dots; bai_n$) having the property:

for all world-histories (h_2) such that $h_2 \in bai_1; bai_2; \dots; bai_n(h)$ and each

basic action instance (bai) such that bai is in $bai_1; bai_2; \dots; bai_n$, $h_2 \in OC(bai)$

then $F_{cl}(bai\text{-}set, h) =_{\text{def}} bai_1@i_1; bai_2@i_2; \dots; bai_n@i_n(h)$

Otherwise, $F_{cl}(bai\text{-}set, h) =_{\text{def}} \{h\}$

BA-CMP1)

For all world-histories (h) and finite basic action instance sets (*bai-set*),

if there exists two sequence functions of *bai-set* (seq_1 and seq_2) such that they both meet the following property of seq :

for all world-histories (h_2) such that $h_2 \in seq(h)$ and each

basic action instance (bai) such that bai is in seq , $h_2 \in OC(bai)$

then $seq_1(h) = seq_2(h)$

BA-CMP2)

For all world-histories (h) and sequence functions (seq_1 and seq_2), if the

sequence functions seq_2 , and $seq_1; seq_2$ all meet the following property of seq :

for all world-histories (h_2) such that $h_2 \in seq(h)$, and each

basic action instance (bai) in the seq , $h_2 \in OC(bai)$

then $seq_2; seq_1$ also meets the property above

Figure 3.2-4

CMP2 by first considering the composition of two basic action instances, both of which have the following property: for all world-histories h , $bai(h)$ contains one world-history. We will refer to any basic action instances bai meeting this property as being **deterministic** since at every world-history h there is a unique closest world-history differing from h on the account of bai . After considering the composition of two deterministic basic action instances, we look at the composition of three or more deterministic basic action instances. Finally, we consider the general case where the basic action instances may have multiple closest world-histories.

The Composition of Two Deterministic Basic Action Instances

We examine the composition of two deterministic basic action instances in a case by case analysis, dividing the cases as follows:

- 1) i) Both $bai1$'s and $bai2$'s standard conditions hold at h , ii) $bai1$'s standard conditions hold in the world-history contained in $bai2(h)$, and iii) $bai2$'s standard conditions hold in the world-history contained in $bai1(h)$
- 2) i) Both $bai1$'s and $bai2$'s standard conditions hold at h , and ii) $bai1$'s standard conditions do not hold in the world-history contained in $bai2(h)$ or $bai2$'s standard conditions do not hold in the world-history contained in $bai1(h)$
- 3) i) $bai1$'s standard conditions hold at h , and ii) $bai2$'s standard conditions do not hold in h
- 4) i) $bai1$'s and $bai2$'s standard conditions do not hold at h

CASE 1

Intuitively, it seems that there are two alternative derivations that can be used to compute the closest world-history to world-history h where the composition of $bai1$ and $bai2$ occurs. One derivation is to minimally modify h so that $bai1$ occurs and then to minimally modify this world-history so that $bai2$ occurs. Symmetrically, we can modify h first by $bai2$ and then by $bai1$. The result of modifying h first by $bai1$ and then by $bai2$ is the world-history belonging to the singleton set $bai1;bai2(h)$, and the result of modifying h first by $bai2$ and then by $bai1$ is the world-history belonging to the singleton set $bai2;bai1(h)$. Both $bai1;bai2(h)$ and $bai2;bai1(h)$ are singleton sets because both $bai1$ and $bai2$ are deterministic basic action instances and hence $bai1(h)$ and $bai2(h)$ are singleton sets. For readability, if h -set is a singleton set, we will sometimes say " $bai1$ occurs in h -set" instead of the correct description " $bai1$ occurs in the world-history belonging to h -set".

The two revision processes, described above, may not yield the same world-history, i.e. $bai1;bai2(h)$ may be unequal to $bai2;bai1(h)$. Consider an example where $bai1$ is "move right hand up during i " and $bai2$ is "move right hand down during i ". These basic action instances cannot occur together under any circumstances. Now,

the closest world-history to h where $bai1$ occurs is a world-history $h1$ where "move hand up during i " occurs ($h1$ is not necessarily distinct from h since $bai1$ might occur in h , and thus $bai1(h) = \{h\}$). The closest world-history to $h1$, where $bai2$ occurs, is a world-history where "move right hand down during i " occurs. Thus, in the world-history belonging to $bai1;bai2(h)$, $bai2$ occurs but not $bai1$. Similarly, in the world-history belonging to $bai2;bai1(h)$, $bai1$ occurs but not $bai2$. Thus, the two revision processes give different results. Diagram 3.2-5 depicts the closest world-histories differing on the account of "move right hand up during i " and closest world-histories differing on the account of "move right hand down during i " at four world-histories in a model. As we will do in the rest of this section, we omit from diagram 3.2-5 the reflexive arcs that correspond to the application of a basic action instance to a world-history in which it occurs. So for example, the arc associated with "move right hand down during i " from $h2$ to $h2$ is omitted from the diagram since this basic action instance occurs in $h2$.

In the above example, we say that "move right hand up during i " and "move right hand down during i " interfere with each other at every world-history. In this case, we set $F_{cl}(\{bai1, bai2\}, h)$ equal to $\{h\}$. Thus, we are treating the composition of two basic action instances that always interfere with each other in the same way that we treat individual basic action instance's whose standard conditions do not hold. Now, as we will see, $F_{cl}(\{bai1, bai2\}, h)$ also equals $\{h\}$ if both $bai1$ and $bai2$ occur in h . This case, however, is distinguished from the case where they interfere with each other since if they interfere they do not both occur together in h .

Now consider two basic action instances that never interfere with each other (See diagram 3.2-6). Let $bai1$ be "move right hand up during i ", and let $bai2$ be "move left hand down during i ". In this case, whether we modify a world-history first by $bai1$ followed by $bai2$, or by $bai2$ followed by $bai1$, we arrive at the same world-history where both $bai1$ and $bai2$ occur. This resultant world-history is taken to be the closest world-history to h where the composition of $bai1$ and $bai2$ occurs. Thus, for all world-histories h , $F_{cl}(\{bai1, bai2\}, h)$ is defined as $bai1;bai2(h)$ which is equal to $bai2;bai1(h)$.

In the two above examples, either the two basic action instances interfered with each other at all world-histories, or they did not interfere at any of them. We may also have cases where two basic action instances interfere only under certain conditions. These situations typically arise when two basic action instances share the same type of resource. Only if a sufficient amount of the resource is present can the two basic action instances be done together.

Consider an example where a robot's power source fluctuates out of the robot's control. Only if enough power is being generated can the robot perform simultaneous actions. Let $bai1$ and $bai2$ be basic action instances with the same time of execution which we will denote by i . Let h be a world-history in which sufficient power is not being generated throughout interval i . We find that by modifying h first by $bai1$, then by $bai2$, we get to a world-history where $bai2$ occurs but not $bai1$. The reason for this is that the condition "sufficient power is not being generated throughout i " holds in the resultant world-history (i.e. $bai1;bai2(h)$)

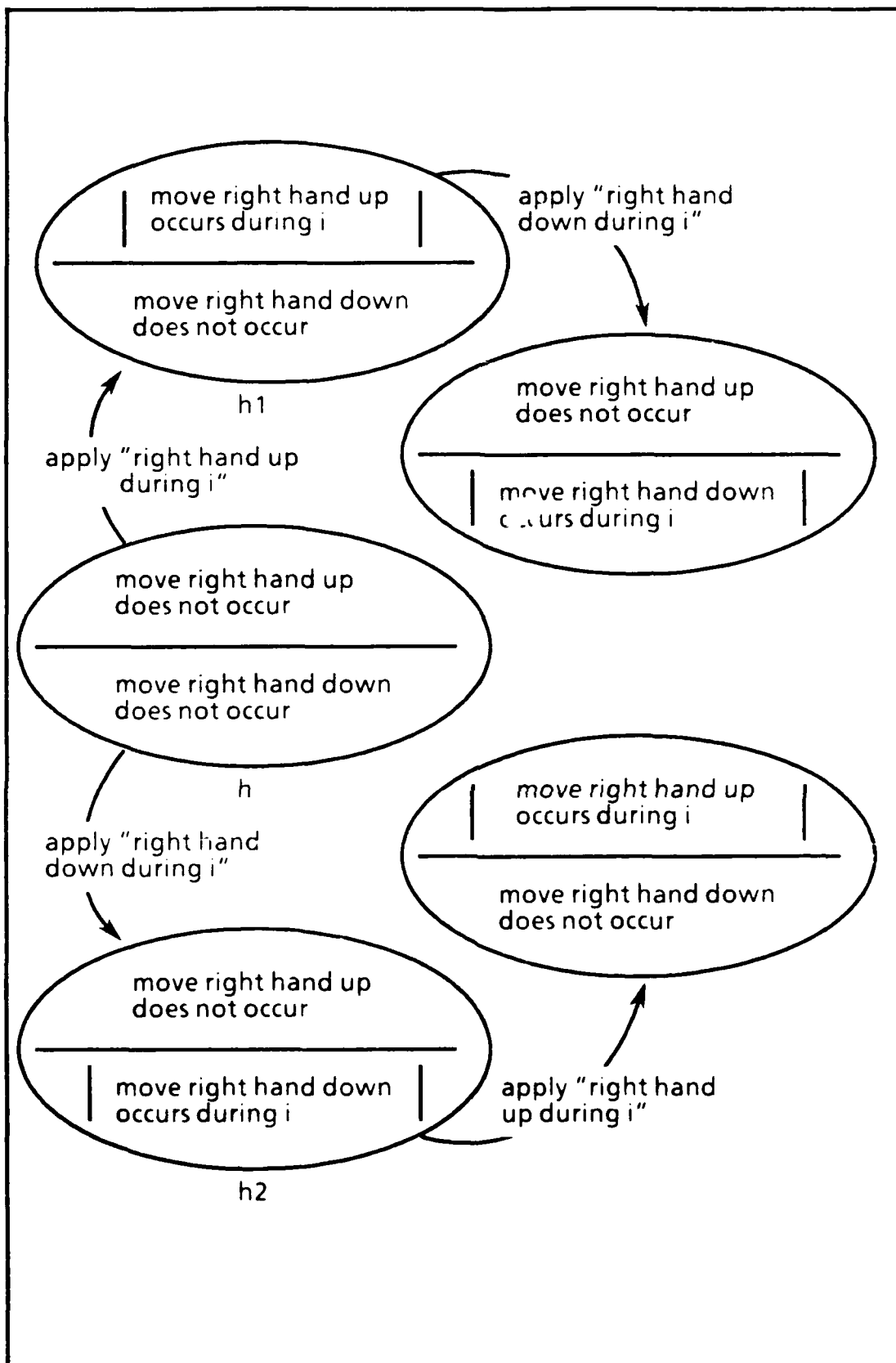


Diagram 3.2-5

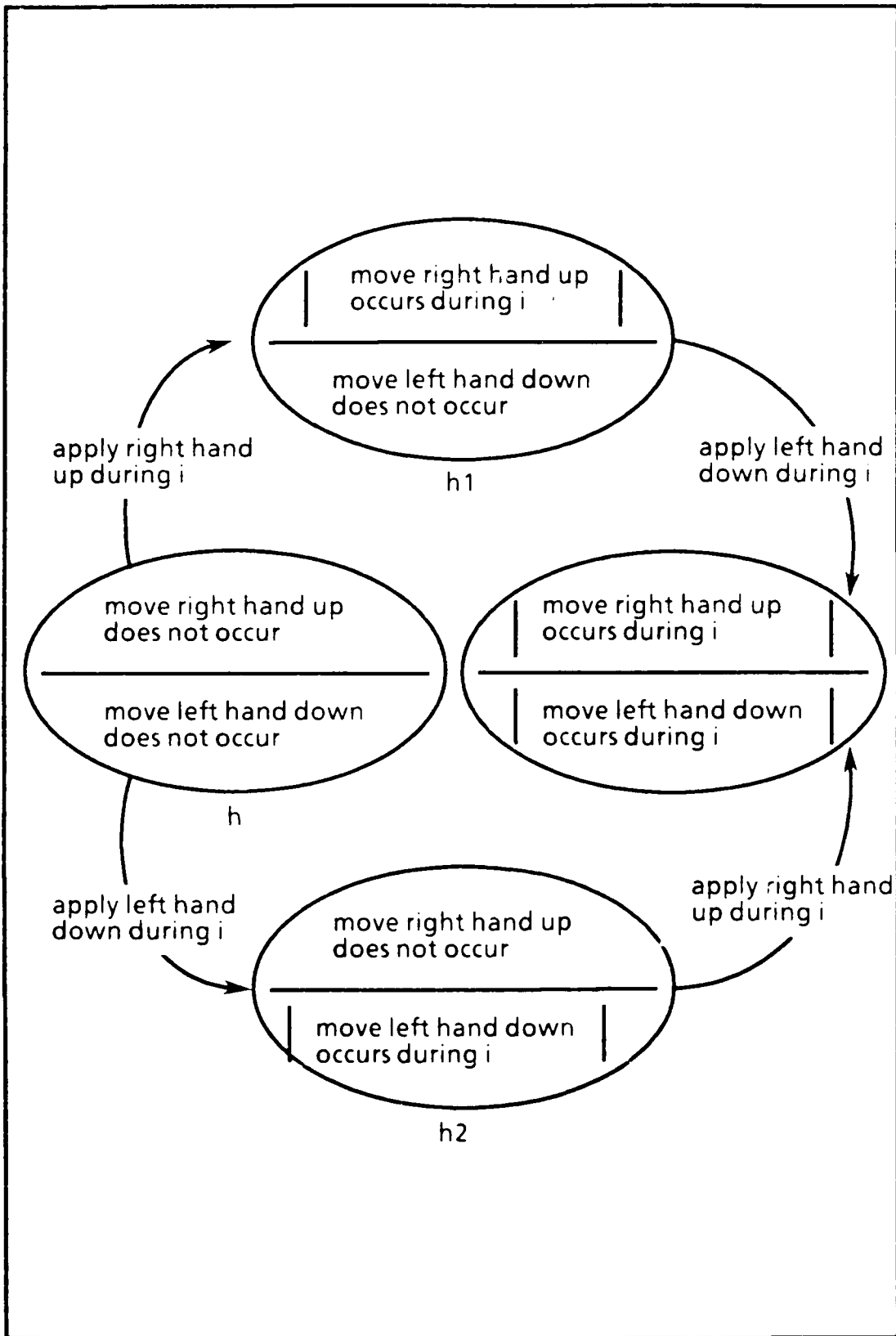


Diagram 3.2-5

because neither the execution of *bai1* or *bai2* affects this condition. Consequently, *bai1* does not occur in *bai1;bai2(h)*, since *bai2* occurs in this world-history along with the condition "sufficient power is not being generated throughout *i*". Thus, *bai1* and *bai2* interfere with each in the environment given by *h*. Appropriately, we set $F_{cl}(\{bai, bai2\}, h)$ equal to $\{h\}$ for any world-history *h* in which "sufficient power is not being generated throughout *i*" holds.

Now, let us look at a world-history *h2* where sufficient power is being generated throughout *i*. In this case, whether we modify *h2* by applying *bai1* followed by *bai2*, or instead, by applying *bai2* followed by *bai1*, we will arrive at the same world-history where both *bai1* and *bai2* occur and "sufficient power is being generated during *i*" holds. Thus, for any world-history *h2* where "sufficient power is being generated during *i*" holds, $F_{cl}(\{bai1, bai2\}, h2)$ is equated with *bai1;bai2(h2)* which is equivalent to *bai2;bai1(h2)*.

In summary, we can describe the composition of two basic action instances that meet *CASE1* as follows, this being a special case of *FCL-DEF* (see figure 3.2-4). For all basic actions instances, *baix* and *baiy*, such that *baix*'s standard conditions hold at *h* and *baiy(h)*, and *baiy*'s standard conditions hold at *h* and *baix(h)*:

FCL-DEF-SC1)

if both *baix* and *baiy* occur in *baix;baiy(h)*,
then $F_{cl}(\{baix, baiy\}, h) =_{def} baix;baiy(h)$

if both *baix* and *baiy* do not occur together in
baix;baiy(h) or *baiy;baix(h)*,
then $F_{cl}(\{baix, baiy\}, h) =_{def} \{h\}$

If *bai1* and *bai2* do not interfere with each other at *h*, then both *bai1* and *bai2* occur in *bai1;bai2(h)* and in *bai2;bai1(h)*. Consequently, the first definition clause in *FCL-DEF-SC1* is applicable for the two substitutions $\{bai1/baix, bai2/baiy\}$ and $\{bai2/baix, bai1/baiy\}$ leading to the result that $F_{cl}(\{bai1, bai2\}, h)$ is set to both *bai1;bai2(h)* and *bai2;bai1(h)*. This definition is well-formed because of the following constraint which is a special case of *BA-CMP1* (see figure 3.2-4), substituting *baix* for *seq1* and *baiy* for *seq2*:

CONSTRAINT1)

For all world-histories (*h*) and deterministic basic action instances (*baix* and *baiy*), if both *baix* and *baiy* occur in *baix;baiy(h)* and both *baix* and *baiy* occur in *baix;baiy(h)*, then *baix;baiy(h)* = *baiy;baix(h)*

If *bai1* and *bai2* interfere with each other at *h*, then both *bai1* and *bai2* do not occur together in either *bai1;bai2(h)* or *bai2;bai1(h)*. Consequently, the second definition clause in *FCL-DEF-SC1* is applicable for the two substitutions $\{bai1/baix, bai2/baiy\}$ and $\{bai2/baix, bai1/baiy\}$ leading to the result that $F_{cl}(\{bai1, bai2\}, h)$ is set to $\{h\}$.

The fact that either i) *bai1* and *bai2* occur together in both *bai1;bai2(h)* and *bai2;bai1(h)*, or ii) they do not occur together in either *bai1;bai2(h)* and *bai2;bai1(h)*,

is captured by the following constraint, which is a special case of *BA-CMP2*, substituting *baix* for *seq1* and *baiy* for *seq2*

CONSTRAINT2)

For all world-histories (*h*) and deterministic basic action instances (*baix* and *baiy*), both of whose standard conditions hold at *h*, if both *baix* and *baiy* occur in *baix;baiy(h)* then both *baix* and *baiy* occur in *baiy;baix(h)*

CONSTRAINT2 insures that the we do not get incompatible results from the two revision processes, *bai1;bai2(h)* and *bai2;bai1(h)*. If both *bai1* and *bai2* occurred together in *bai1;bai2(h)*, but did not occur together in *bai2;bai1(h)*, the former relation would be saying that *bai1* and *bai2* did not interfere at *h* while the latter would be saying they did interfere.

We must emphasize that we are formalizing basic action instance composition, not plan instance composition. Many interactions between plan instances mirror the interactions between basic actions instances. There are, however, some interactions that can only be captured at the plan instance level. One example is where two programs can be successfully executed separately, but if they are executed simultaneously they deadlock and neither occurs. Let the plan instances $\langle \{ \langle \text{prog1}, i \rangle \}, \{ \langle \text{com1}, i \rangle \} \rangle$ and $\langle \{ \langle \text{prog2}, i \rangle \}, \{ \langle \text{com2}, i \rangle \} \rangle$ refer to two processes that deadlock if executed together. The first plan instance corresponds to the event "program *prog1* (successfully) runs" that is performed during interval *i* by executing the basic action instance *com1@i*, similarly for the second plan instance. The basic action instances *com1@i* and *com2@i* are treated as if they do not interfere, while the plan instances that they generate are treated so that they do interfere. This is captured by a model in which the execution of *com1@i*, in any world-history in which $\langle \{ \langle \text{prog2}, i \rangle \}, \{ \langle \text{com2}, i \rangle \} \rangle$ occurs, leads to a closest world-history where neither *prog1* or *prog2* occur during *i*, similarly, for the execution of *com2@i* in a world-history where $\langle \{ \langle \text{prog1}, i \rangle \}, \{ \langle \text{com1}, i \rangle \} \rangle$ occurs (see diagram 3.2-7). Conversely, if *com1@i* is executed in a world-history where $\langle \{ \langle \text{prog2}, i \rangle \}, \{ \langle \text{com2}, i \rangle \} \rangle$ does not occur, then the execution of *com1@i* leads to a world-history where $\langle \{ \langle \text{prog1}, i \rangle \}, \{ \langle \text{com1}, i \rangle \} \rangle$ occurs, similarly, for the symmetrical case.

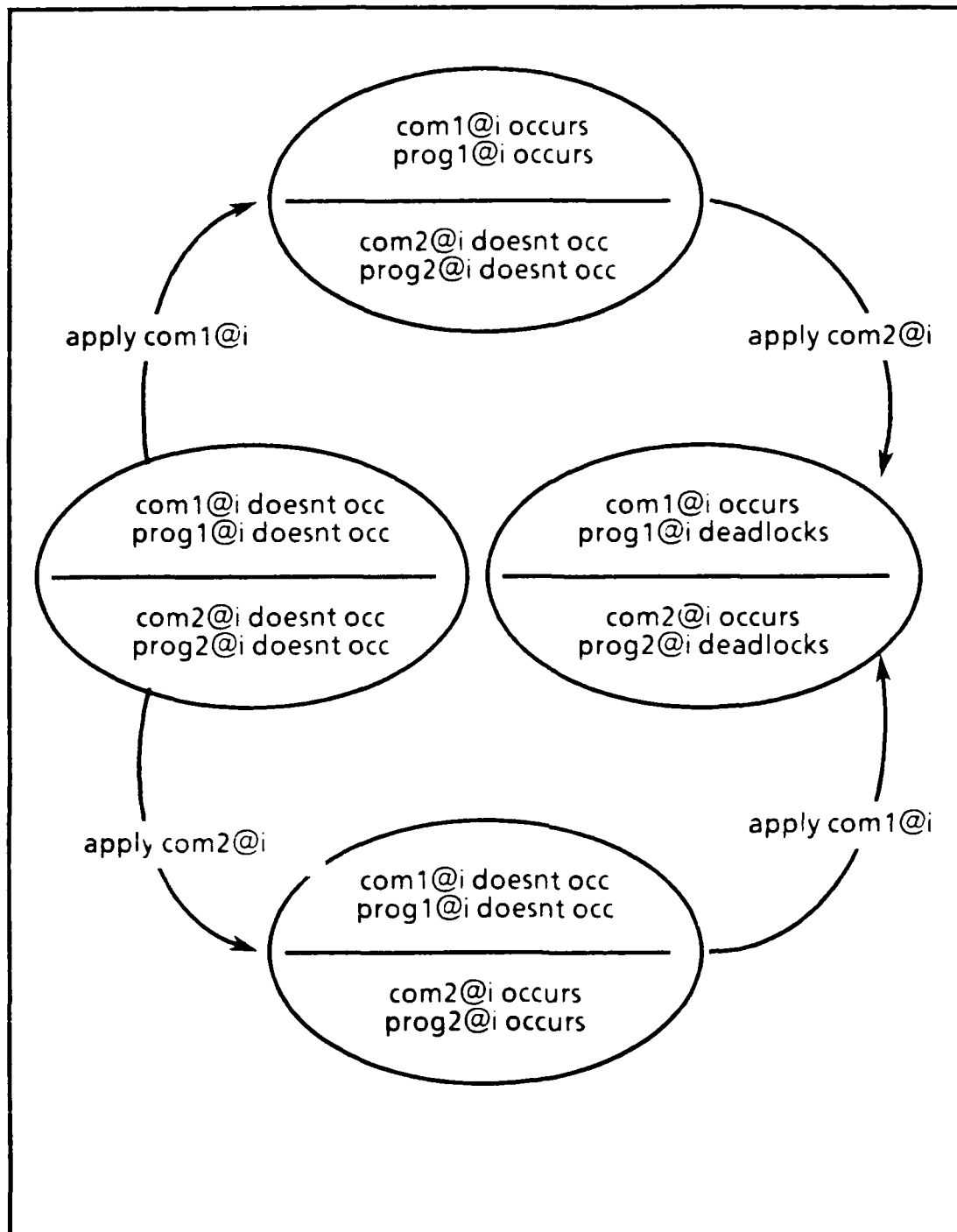


Diagram 3.2-7

CASE2

The next case to consider is where i) both *bail*'s and *bai2*'s standard conditions hold at *h*, and ii) *bail*'s standard conditions do not hold in the world-history contained in *bai2*(*h*) or *bai2*'s standard conditions do not hold in the world-history contained in *bail*(*h*). Let us suppose that the application of *bail* ruins *bai2*'s standard conditions. In this case, the desired result is that *bail* and *bai2* cannot be executed at *h*. Thus, we want to set $F_{cl}(\{bail, bai2\}, h)$ equal to $\{h\}$.

This relationship between *bail* and *bai2*, where the execution of *bail* with respect to *h*, ruins *bai2*'s standard conditions, can also be seen as interference at world-history *h*. In this case, both *bail* and *bai2* do not occur together in *bail*;*bai2*(*h*). More specifically, *bail* occurs, but not *bai2*, in *bail*;*bai2*(*h*). The reason for this is as follows. Since application of *bail* at *h* ruins *bai2*'s standard conditions, *bai2* does not occur in *bail*(*h*) and *bai2* applied to the world-history contained in *bail*(*h*) maps back to itself. Consequently, *bail*;*bai2*(*h*) equals *bail*(*h*), a set containing a world-history in which *bail* occurs, but not *bai2*. We also assume that both *bail* and *bai2* do not occur together in *bai2*;*bail*(*h*). Otherwise, we would get incompatible results from the two revision processes, *bail*;*bai2*(*h*) and *bai2*;*bail*(*h*), the first saying that *bail* and *bai2* interfere with each other at *h* and the other saying they do not interfere.

Thus, we see that if the application of *bail* at *h* ruins *bai2*'s standard conditions, then *bail* and *bai2* do not both occur together in either *bail*;*bai2*(*h*) or *bai2*;*bail*(*h*). Consequently, the definition given in *FCL-DEF-SC1* yields the correct result. In particular, the second definition clause in *FCL-DEF-SC1* is applicable for the two substitutions $\{bail/baix, bai2/baiy\}$ and $\{bai2/baix, bail/baiy\}$ leading to the result that $F_{cl}(\{bail, bai2\}, h)$ is set to $\{h\}$. It is also easily established that *CONSTRAINT1* and *CONSTRAINT2* are compatible with this case.

CASE3

We now consider the composition of two deterministic basic actions instances where one of the basic action instances's standard conditions holds and the other one does not hold at *h*. Without loss of generality assume that *bail*'s standard conditions hold at *h*, while *bai2*'s standard conditions do not hold at *h*. The desired result is that *bail* and *bai2* can be executed together only if the occurrence of *bail* brings about *bai2*'s standard conditions and the two basic action instance do not interfere at *h*. To determine whether the occurrence of *bail* brings about *bai2*'s standard conditions, we modify *h* by *bail* arriving at a world-history, which we will call *h1*, and see if *bai2*'s standard conditions hold at *h1*. If this is the case, then *bai2*(*h1*) which is equal to *bail*;*bai2*(*h*) is a singleton set in which *bai2* occurs. Secondly, if it is also the case that *bai2* does not interfere with *bail* at *h* (and thus also at *h1*), *bail* will also occur in *bail*;*bai2*(*h*). Therefore, if *bail*;*bai2*(*h*) contains a singleton set in which both *bail* and *bai2* occur, then the occurrence of *bail* brings about *bai2*'s standard conditions and the two basic action instances do not interfere at *h*. In this

case, we equate $F_{cl}(\{bail, bai2\}, h)$ with $bail; bai2(h)$. On the other hand, if both *bail* and *bai2* do not occur in $bail; bai2(h)$, we equate $F_{cl}(\{bail, bai2\}, h)$ with $\{h\}$, treating the composition as if its standard conditions do not hold.

The case where only one of *bail*'s and *bai2*'s standard conditions holds at *h* differs from the case where both basic action instances' standard conditions hold at *h* in that we must try out both orderings, "*bail* followed by *bai2*" and "*bai2* followed by *bail*", to determine whether they can be executed together in *h*. If execution of *bail* with respect to *h* brings about *bai2*'s standard conditions, then both *bail* and *bai2* occur in $bail; bai2(h)$, but not in $bai2; bail(h)$. Symmetrically, if execution of *bai2* with respect to *h* brings about *bail*'s standard conditions at *h*, then both *bail* and *bai2* occur in $bai2; bail(h)$, but not in $bail; bai2(h)$. Thus, we do not want to necessarily equate $bail; bai2(h)$ with $bai2; bail(h)$ when both *bail* and *bai2* occur in $bail; bai2(h)$. This is only done when additionally both *bail*'s and *bai2*'s standard conditions hold at *h*. *CONSTRAINT1*, which we put forth to equate $bail; bai2(h)$ and $bai2; bail(h)$ in the appropriate cases, does not apply here. This is because its antecedent (i.e. "*both bail and bai2 occur in bail; bai2(h)*") is false for both substitutions. *CONSTRAINT2* is also compatible since its provision "*both bail's and bai2's standard conditions hold at h*" is not met for either substitution.

Definition *FCL-DEF-SC1* also applies to this case. If the execution of *bail* in *h* brings about *bai2*'s standard conditions and the two basic action instances do not interfere at *h*, then the first definition clause in *FCL-DEF-SC1* is applicable for the substitution $\{bail/baix, bai2/baiy\}$ leading to the result that $F_{cl}(\{bail, bai2\}, h)$ is set to $bail; bai2(h)$. Conversely, if either the execution of *bail* in *h* does not bring about *bai2*'s standard conditions or the two basic action instances interfere at *h*, then the second definition clause in *FCL-DEF-SC1* is applicable for both substitutions leading to the result that $F_{cl}(\{bail, bai2\}, h)$ is set to $\{h\}$.

CASE4

The last case to consider is the composition of two deterministic basic action instances both of whose standard condition do not hold at *h*. In this case, neither of the basic action instances can be executed alone to bring about the other one's standard conditions. We therefore simply equate $F_{cl}(\{bail, bai2\}, h)$ with $\{h\}$. In this case, both $bail; bai2(h)$ and $bai2; bail(h)$ also equal $\{h\}$, a world-history in which neither basic action occurs. It is easily shown that *FCL-DEF-SC1* yields the correct result and that both *CONSTRAINT1* and *CONSTRAINT2* are compatible.

Summary of Cases 1 - 4

In summary, we have shown that F_{cl} applied to the composition of two deterministic basic action instances can be given by *FCL-DEF-SC1* which is a special case of *FCL-DEF*. We have also shown that all these cases are compatible

with *CONSTRAINT1* and *CONSTRAINT2* which are special cases of *BA-CMP1* and *BA-CMP2*.

Composing three or more deterministic basic action instances

The composition of three or more deterministic basic action instances is derived by generalizing from the case where two basic action instances are composed together. To begin with, in order to conclude that two basic action instances can be done together at h , it must be the case that they do not interfere with each other at h . Analogously, to conclude that three or more basic action instances can be done together, it must be the case that all the basic action instances when taken together do not interfere with each other. We must clarify by noting that a set of basic action instances taken pair-wise may not not interfere, but when all taken together they interfere. As an example, consider the case where there are three basic action instances $bai1$, $bai2$, and $bai3$ that consume the same type of resource. Assume that $bai1$'s execution time is $i1$ which is properly before $bai2$'s and $bai3$'s execution time which is $i2$. Also assume that in world-history h , none of these basic action instances occur and there are two units of the resource available during intervals $i1$ and $i2$. In this case, $bai2$ and $bai3$ do not interfere with each other at world-history h , but the three basic action instances interfere with each other at h . If $bai1$ were to be executed in h , it would use up one resource and therefore only one resource would be available during $i2$. Hence, both $bai2$ and $bai3$ could not both be executed together along with $bai1$ in h .

Conversely, a relation among basic action instances may arise where two basic actions interfere with each at some world-history h , but these basic action instances could be executed together if some additional basic action instance were to be executed along with them. Consider the example where two simultaneous basic action instances $bai1$ and $bai2$ share the same type of resource, and let h be a world-history in which only one unit of the resource is available. In this case, $bai1$ and $bai2$ interfere with each other at h . Now, consider a third basic action instance $bai3$ which produces additional resources. In this case, $bai1$, $bai2$, and $bai3$ all can be executed together, and we say that they do not interfere with each other when taken together at h .

A second factor that we took into account in deriving the composition of two basic action instances is the order in which we successively modify a world-history by the two basic action instances. It might be the case that $bai1$ and $bai2$ can be executed together because either $bai1$ brings about $bai2$'s standard conditions (and they do not interfere with each other), or $bai2$ brings about $bai1$'s standard condition. Thus, the order in which we successively modify a world-history by two basic actions may be crucial. If $bai1$ brings about $bai2$'s standard conditions and the two basic action instances do not interfere with each other at h , then both $bai1$ and $bai2$ occur in $bai1;bai2(h)$ while $bai2;bai1(h)$ equals the null set. Conversely, if $bai2$ brings about $bai1$'s standard conditions and $bai1$ and $bai2$ do not interfere with each other at h ,

then both *bai1* and *bai2* occur in *bai2;bai1(h)* while *bai1*, but not *bai2* occurs in *bai1;bai2(h)*.

This situation is easily generalized to the case where there are three or more basic action instances. To check if we can execute three or more basic action instances together with respect to *h*, we may have to check the modifications of *h* by all the possible orderings of the basic action instances. For example, consider the composition of *bai1*, *bai2* and *bai3*. To determine whether they can be executed together, we may have to check the six functions formed by the six orderings of the three basic action instances: *bai1;bai2;bai3*, *bai1;bai3;bai2*, *bai2;bai1;bai3*, *bai2;bai3;bai1*, *bai3;bai1;bai2*, and *bai3;bai2;bai1*. For example, it might be the case that *bai1*'s standard conditions hold everywhere, *bai2*'s standard conditions hold only if *bai1* occurs, and *bai3*'s standard conditions hold only if *bai2* occurs. In this case, if *h* is a world-history where none of three basic action instances occur, *bai1;bai2;bai3(h)* will be the only one out of the six functions that does not equal the null set. If additionally, *bai1*, *bai2*, and *bai3* all occur in the world-history belonging to *bai1;bai2;bai3(h)*, then we conclude they do not interfere with each at *h*, and we appropriately set $F_{cl}(\{bai1, bai2, bai3\}, h)$ equal to *bai1;bai2;bai3(h)*.

The definition of F_{cl} given by *FCL-DEF* provides for this generalization where there are two or more basic actions instances belonging to *bai-set*. The "otherwise" clause in *FCL-DEF*, which sets $F_{cl}(bai-set, h)$ to $\{h\}$, is only applicable if there are no sequences that yield a world-history in which all the elements in *bai-set* occur in. The constraint given by *BA-CMP1* insures that if there are two sequences that yield a world-history in which all the elements in *bai-set* occur in, then they must yield the same world-history. Without this constraint *FCL-DEF* would be ill-formed. Constraint *BA-CMP2* is a generalization of *CONSTRAINT2* and can be explained as follows.

Suppose that all the basic action instances in *bai-set* can be executed together in *h* as manifested by the fact that there is a function sequence of *bai-set* that when applied to *h* yields a world-history in which all the elements in *bai-set* occur in. We will say that any sequence function of *bai-set* that meets this property is a solution sequence. We consider a solution sequence which we break up into two parts represented by two sequence functions combined by ";". Let *seq1;seq2* refer to such a combined sequence function. We also assume that when *seq2* is applied to *h* it yields a world-history in which all its members occur in. Now, using *BA-CMP2* we can derive that *seq2;seq1* is also a solution sequence.

The justification for this conclusion stems from the assumption that the only reason that different sequence functions of *bai-set* must be tried is because there may be some set of basic action instances or some single one whose standard conditions do not hold unless other basic action instances in *bai-set* are sequenced before it. (Remember, we say that a set's standard conditions do not hold if the elements cannot be executed together) If, however, there is a subset of *bai-set* whose standard conditions hold in *h*, then we can reorder any sequence that is a solution sequence so that some sequence of this subset is in front. In our example, the elements of *seq2* are such a subset.

Basic Action Instances with Multiple Closest World-Histories

We now examine the composition of two or more basic action instances that may have multiple closest world-histories. We show that definition *FCL-DEF* provides for this case and that constraints *BA-CMP1* and *BA-CMP2* are applicable. The treatment of basic action instance with multiple closest world-histories is just like the treatment of basic action instances with unique closest world-histories with the following exception. Instead of simply having cases where either *bai1* and *bai2* interfere at *h* or do not interfere at *h*, we may also have the situation where only some of *bai1*'s behaviors interfere with some of *bai2*'s behaviors. Similarly, we encounter cases where some of *bai1*'s behaviors bring about *bai2*'s standard conditions while other behaviors do not.

If there is a behavior of *bai1* that interferes with a behavior of *bai2*, $F_{cl}(\{bai1, bai2\}, h)$ is set to $\{h\}$, treating this composition like two deterministic basic action instance that interfere. Similarly, if some but not all of *bai1*'s behaviors bring about *bai2*'s standard conditions, we set $F_{cl}(\{bai1, bai2\}, h)$ to $\{h\}$. This treatment of F_{cl} does not distinguish between the case where all of the behaviors of *bai1* and *bai2* interfere and the case where some of them interfere. Similarly, F_{cl} does not distinguish between the case where some of *bai1*'s behaviors bring about *bai2*'s standard conditions from the case where none of *bai1*'s behaviors bring about *bai2*'s standard conditions. For our purposes, however, making this distinction is not necessary. We are ultimately interested in whether the composition of two or more plan instance are executable so that they can be used a part of a solution for a planning goal. Now, two plan instances are executable only if the basic action instances associated with them do not interfere. Whether two basic action instances interfere under all behaviors or interfere only under some behaviors, we would want to reject, as a solution, the composition of any plan instances associated with these basic action instances. Making this distinction would only be useful if we wanted a more detailed model of harmful interactions.¹

FCL-DEF captures the relations described above. Since we are no longer assuming that the basic action instances are deterministic, *bai1*(*h*), *bai1*;*bai2*(*h*), *bai2*(*h*), etc. may contain more than one world-history. The world-histories that are in *bai1*;*bai2*(*h*) are those that are reached by first modifying *h* by one of *bai1*'s behaviors and then modifying by one *bai2*'s behaviors. Thus, the set *bai1*;*bai2*(*h*) is the result of pairing all *bai1*'s behaviors with all of *bai2*'s behaviors (modifying first by *bai1*). Consequently, if there are two or more pairings that interfere with each other, there will be at least one world-history belonging to *bai1*;*bai2*(*h*) in which both *bai1* and *bai2* do not occur together. Similarly, if only some of *bai1*'s behaviors bring about *bai2*'s standard conditions, there will be at least one world-histories belonging to *bai1*;*bai2*(*h*) in which both *bai1* and *bai2* do not occur together (more

¹ This is not to suggest that we cannot capture this distinction in the model. This difference can be determined by looking at the two basic action instance functions separately. We are just defining F_{cl} so as not to make this distinction

precisely. there will be at least one world-history in which *bai2* does not occur). Thus, we see that the first definition clause in *FCL-DEF* only holds if there is an ordering of $\{bai1, bai2, \dots, bain\}$ where no combination of behaviors interfere with each other. This is because this clause only holds if all the members in $\{bai1, bai2, \dots, bain\}$ occur in all the world-histories (*h2*) belonging to *bai1; bai2; ...; bain(h)*.

For the case of multiple world-histories, *BA-CMP1* says: if there are two sequence functions, *seq1* and *seq2*, of the same set *bai-set* in which all the members of *bai-set* occur in all the world-histories in *seq1(h)* and all the world-histories in *seq2(h)*, then *seq1(h)* equals *seq2(h)*. This makes sense because if all of the combinations of the different behaviors do not interfere with each other, then the order in which we modify the world-histories should produce the same set of combinations (as long as the sequences do not violate the orderings needed to insure that some basic action instance's standard condition hold). In a similar manner, we could justify *BA-CMP2* by adapting the argument we gave for *BA-CMP2* in the last section on deterministic basic action instances.

3.2.6. Comparison with Semantic Theories of Conditionals

The semantic interpretation of the IFTRIED modality derives from the semantic theories of conditionals developed by Stalnaker [Stalnaker 68] and Lewis [Lewis 73]. The connection between these theories and our treatment is brought out by looking at $\lceil \text{IFTRIED } t_{pi} P \rceil$ as a subjunctive conditional having the form "If t_{pi} were to be attempted then P would be true". In the following sections, we give a brief description of both these theories and then discuss their relation to our theory.

The Treatment of Counterfactuals in a Possible Worlds Framework

The approach of interpreting counterfactual conditionals within the possible worlds framework originates in the work of Stalnaker and Lewis. In both theories, counterfactuals are interpreted in terms of an accessibility relation (actually an accessibility function in Stalnaker's case) that measures the "closeness" (i.e. relative similarity) between possible worlds in the model. In Stalnaker's models, a *selection function* is introduced which takes a proposition P and a world w and yields the closest world to w where P is true. The counterfactual "If A then C " is interpreted as true at possible world w iff C is true at the closest world to w where A is true.¹ Stalnaker presented a number of constraints that are placed on all selection functions. For example, one constraint stipulated that the closest world to a world is closest to itself. The other constraints will be described shortly in conjunction with Lewis' theory.

Stalnaker motivates the use of a closeness measure to interpret counterfactuals by first describing a pragmatic principle that specifies how one evaluates a counterfactual. This principle, which is an adaptation of a test put forth by Frank Ramsey, is given as follows:

Suppose that you want to evaluate the counterfactual, "If A then C ". First you hypothetically add the antecedent A to your stock of beliefs and make the minimal revision required to make A consistent. You then consider the counterfactual to be true iff the consequent C follows from this revised stock of beliefs.

He then states that in going from a principle that describes the justified belief of a counterfactual to a theory that gives the truth conditions to counterfactuals, we make a transition from "a stock of beliefs" to "a possible world", and from the notion of "minimal belief revision" to the notion of "closest possible world".

¹ If proposition P is not possible with respect to w , as specified by a second accessibility relation in the model, then the selection function applied to P at w returns the "impossible world", a world where every proposition is true. The notion of an "impossible world" is introduced just for convenience so that the interpretation of "If A then C " at w comes out to be (vacuously) true if P is impossible at world w .

Lewis' Treatment of Conditionals

Lewis' treatment of conditionals differs from Stalnaker's theory in two primary ways. To begin with, he permits models where there may be many worlds equally close to a world where some proposition holds. If this is the case, then the counterfactual "If A then B" is interpreted as true at w iff B is true in all the closest worlds to w where A is true. One result of permitting models with multiple closest worlds is that the law of the conditional excluded middle, which is valid in Stalnaker's logic, is not valid in Lewis' logic. The law of the conditional excluded middle is given by:

"If A then B" or "If A then not B" is true

If the law of the excluded middle does not hold, one cannot negate a conditional by simply negating the consequent (i.e. the negation of "If A then B" is not "If A then not B"). To get around this, Lewis identifies two forms of conditionals, "If A then B would be true" and "If A then B might be true" which are duals for a fixed antecedent, that is the following holds:

the negation of "If A then B would be true" is equivalent to
"If A then not B might be true"

The second difference between Lewis' and Stalnaker's systems is that Lewis does not make what he calls *the limit assumption*. This is an assumption to the effect that for any world w and proposition P that is possible with respect to w , there exists at least one closest world to w where P holds. An example that contravenes the limit assumption is the case where we have an antecedent of the form "Tom is taller than 7 feet". We work under the assumption that for any positive rational number n (say under 10), there exists a world where Tom is n feet. In this case, given any world w , there is no closest world to w where Tom is greater than 7 feet. We can get closer and closer arriving at a world where Tom is $7 + \epsilon$ feet, but for any ϵ , we can always get to a closer world where Tom is $7 + (\epsilon/2)$ feet.

Lewis circumvents this problem by interpreting "If A then C" at world w by looking to see if the consequent C is true as we get to closer and closer worlds in which A is true. In particular, he introduces a comparative similarity relation that for each world w ordered the set of worlds in accordance with how close they are to w . He uses " $w_2 \leq_w w_1$ " to mean that with respect to w , w_2 is not closer than w_1 . The conditional "If A then C" is interpreted as true at w iff there does not exist a world (w_2) such that A is true and C is false at w_2 and w_2 is closer to w than every world where both A and B is true. In the case where there exists a set of closet worlds where A is true, the limit-type interpretation coincides with the interpretation where one checks if the consequent is true in all closet worlds where the antecedent is true.

Lewis shows that if the limit assumption is adopted in his models, then his formulization is essentially Stalnaker's formulization with the exception that multiple closest worlds are allowed. To illustrate this connection, Lewis defines a

selection function, defined in terms of his comparative similarity relation, which we will refer to by *SL*. $SL(P, w)$ yields the set of closest world histories to w where P is true (and equals the null set if P is impossible with respect to w). He demonstrates that the restrictions imposed on *SL* (as inherited from the restrictions imposed on \leq_{w0}) match the restrictions that Stalnaker imposes on his selection function. These restrictions are given as follows:

- SL1) If P is true in world w , then $SL(P, w) = \{w\}$
- SL2) P is true in all the worlds in $SL(P, w)$
- SL3) If Q is true in all the worlds in which P is true in and $SL(P, w)$ is non-empty, then $SL(Q, w)$ is non-empty
- SL4) If Q is true in all the worlds in which P is true in and there exist a world in $SL(Q, w)$ in which P is true in, then $SL(P, w)$ equals the subset of $SL(Q, w)$ in which P is true in

note: if P is not possible at w , then $SL(P, W)$ equals the null set

The *SL* selection function and the F_{cl} function

The statement $\lceil (IFTRIED\ t_{pi}\ P) \rceil$ is interpreted as true at h iff P is true in all world-histories belonging to $F_{cl}(bai\text{-}set, h)$, where *bai-set* is the basic action instance set associated with t_{pi} . $F_{cl}(bai\text{-}set, h)$ yields the set of closest world-histories to h where all basic action instances belonging to *bai-set* occur (if they can be executed together). If all the elements in *bai-set* occur together, we say that t_{pi} is attempted. Thus the elements of $F_{cl}(bai\text{-}set, h)$ can equivalently be described as the closest world-histories to h where (the plan instance denoted by) t_{pi} is attempted. Consequently, we can describe our interpretation of $\lceil (IFTRIED\ t_{pi}\ P) \rceil$ as: $\lceil (IFTRIED\ t_{pi}\ P) \rceil$ is true at h iff P is true in all the closest world-histories to h where t_{pi} is attempted. This is just like the interpretation of the conditional "if t_{pi} were to be attempted, then P would be true" using the framework described by Lewis and Stalnaker.

Since F_{cl} yields a set of closest world-histories that differ on the account of a specified set of basic action instances, our treatment is along the lines of a theory of conditionals that provides for multiple closest worlds and makes the limit assumption. Thus, it is natural to compare F_{cl} with the *SL* selection function.

To begin with, because of the presence of multiple closet world-histories, the analog to the law of the excluded middle does not hold in our system. Instead, we get two senses of *IFTRIED* corresponding to Lewis's "would" and "might" conditionals. *IFTRIED* is treated as a "would" conditional since $\lceil (IFTRIED\ t_{pi}\ P) \rceil$ is true at h iff P is true in all the closest world-histories to h where t_{pi} is attempted. The "might" counterpart to *IFTRIED* is designated by *P-IFTRIED*. The *P-IFTRIED* modality is simply defined as the dual of *IFTRIED* for a fixed plan instance term:

$$(P\text{-IFTRIED } t_{pi} P) =_{\text{def}} (\text{NOT} (\text{IFTRIED } t_{pi} (\text{NOT } P)))$$

Constraint *SL1* above, which says that if P is true in w then w is the closest world to itself where P holds, is analogous to the following constraint which we impose on the basic action functions:

BA1)

For every basic action(ba), interval (i), and world-history (h),
if $\langle i, h \rangle \in \text{BAEV}(ba)$ then $ba(i, h) = \{h\}$

Recall that $F_{cl}(bai\text{-}set, h)$ is defined in terms of the basic action functions that are in $bai\text{-}set$ (see figure 3.2-4). For the simple case when $bai\text{-}set$ consists of only one basic action instance, $F_{cl}(\{ba@i\}, h) = h$. Using *BA1* and the definition of F_{cl} given in 3.2-4, we can derive that if all the basic action instances in $bai\text{-}set$ occur in h , then $F_{cl}(bai\text{-}set, h) = \{h\}$.

Constraint *SL2* above says that P is true in all the closest worlds to w where P is true. The analogous constraint in our system is given by *BA2*

BA2)

For every basic action(ba), interval (i), and world-histories (h and $h2$),
if $h = h2$ and $h2 \in ba(i, h)$ then $\langle i, h2 \rangle \in \text{BAEV}(ba)$

BA2 can be interpreted as saying that basic action instance $ba@i$ occurs in every world-history belonging to $ba(i, h)$ which does not equal h . The last provision is made because of the way that we are treating basic action instances whose standard conditions do not hold at h . If $ba@i$'s standard conditions do not hold at h , we simply set $ba(i, h)$ to $\{h\}$. In this case $ba(i, h)$ contains a world-history (i.e. h) in which $ba@i$ does not occur. Similarly, $F_{cl}(bai\text{-}set, h)$ is set to $\{h\}$ when $bai\text{-}set$ contains two or more basic action instances that cannot occur together.

Both *STL3* and *STL4* describe relations between $SL(P, w)$ and $SL(Q, w)$ when proposition P entails proposition Q . Constraint *SL3* does not apply to our system because it pertains to systems where a closeness function may be applied to a proposition that is not possible at the world it is being applied at. This constraint insures that if P entails Q , then SL cannot be defined so that it encodes that P is possible at world w while Q is not.

SL4 can be seen as a constraint insuring that a compatible closeness metric is used to evaluate P and Q . If P entails Q , then we do not want a world $w2$ in $SL(P, w)$ that is not in $SL(Q, w)$ unless all worlds in $SL(Q, w)$ are closer than any world in $SL(P, w)$. One might try to implement an analogous constraint in our system. The relation between P and Q in our system correspond to a relation between a basic action instance set $bai\text{-}set_p$ which contains another basic action instance set $bai\text{-}set_q$. It can be shown that the analogous relation to *SL4* holds in for the limited case where all elements of $bai\text{-}set_q$ occur in all world-histories in $F_{cl}(bai\text{-}set_p, h)$. In this case $F_{cl}(bai\text{-}set_p, h)$ equals $F_{cl}(bai\text{-}set_q, h)$.

Chapter 4

A Proof Theory

In this chapter, we present a proof theory that is sound with respect to the semantic theory presented in the last chapter. This proof theory is given by specifying a set of sentences, i.e. axioms, and a set inference rules that produce the sentences that are theorems in our system. We demonstrate that our proof theory is sound by showing that all axioms and all theorems produced by the inference rules are valid with respect to our semantic theory.

We break up the rest of the chapter into four sections. We first present a standard axiomatization of the first order connectives and the equality predicate. We then present the axioms concerning the predicates and terms in our non-modal fragment. The last two sections describe the axiomatization of our two modal operators, *INEV* and the *IFTRIED*. In these sections, we intersperse the presentation of the axioms and inference rules with informal discussion. An encapsulated description of the proof theory is given in appendix D.

We will use the notation " $\vdash P$ " to mean that sentence P is a theorem with respect to our proof theory. We will also use " \vdash " in binary form. We describe the relation " $S \vdash P$ " by saying that sentence P is derivable from the set of sentences in S (which we will assume to be finite).

4.1. Axiomatization of the First Order Connectives

We adapt a standard axiomatization of the first order connectives with equality as described in [Hughes and Cresswell 68]. The axioms and inference rules are given in figure 4.1-1. *AX-FO8* is the only axiom that is unique to our system. This axiom captures the fact that the sets of objects corresponding to the different types are pairwise disjoint. Consequently, two objects having different types cannot be equal.

Strictly speaking, *AX-FO1* - *AX-FO8* are *axioms schemas*, not axioms, although we will use both terms interchangeably. Each schema specifies a set of axioms for any substitutions of the non-logical symbols appearing in them. Sentences in the language, including both non-modal and modal sentences, are substituted for capital letters and terms in the object language are substituted for small letters. Substitutions for any sentence or term can be made as long as they meet the provision associated with the schema, if such a provision exists. Schemas only have provisions if noted. Thus, *AX-FO1* - *AX-FO4*, *AX-FO6* have no provisions, while *AX-FO5*, *AX-FO7*, and *AX-FO8* do.

The only difference made in going from an axiomatization of first order logic *simpliciter* to a logic where first order logic is extended with modal operators is that the sentence (meta) variables appearing in the schemas given in figure 4.1-1 range over both modal and non-modal statements, instead of just over first order

Axioms

AX-FO1)

 $\vdash (\text{IF } (\text{OR } P \ P) \ P)$

AX-FO2)

 $\vdash (\text{IF } Q \ (\text{OR } P \ Q))$

AX-FO3)

 $\vdash (\text{IF } (\text{OR } P \ Q) \ (\text{OR } Q \ P))$

AX-FO4)

 $\vdash (\text{IF } (\text{IF } Q \ R) \ (\text{IF } (\text{OR } P \ Q) \ (\text{OR } P \ R)))$

AX-FO5)

 $\vdash (\text{IF } (\forall ?v \ P1) \ P2)$

where $P1$ differs from $P2$ in having all free occurrences of $?v$ in $P1$ by some term t that has the same type as variable $?v$, and if term t has any variables in it, then they must not become bound by the substitution

AX-FO6)

 $\vdash (= \ t \ t)$

AX-FO7)

 $\vdash (\text{IF } (= \ t1 \ t2) \ (\text{IF } P1 \ P2))$

where $P1$ differs from $P2$ in having one or more free occurrences of $t1$ in $P1$ replaced by $t2$ and if term $t2$ has any variables in it, then they must not become bound

AX-FO8)

 $\vdash (\text{NOT } (= \ t1 \ t2))$

where $t1$ and $t2$ have different types

Inference rules:

MP)

From: $S1 \vdash P$ and $S2 \vdash (\text{IF } P \ Q)$ To: $S1 \cup S2 \vdash Q$

where $S1$ and $S2$ are any finite set of sentences, and P and Q are any sentences

UNV-INTRO)

From: $S \vdash (\text{IF } P \ Q)$ To: $S \vdash (\text{IF } P \ (\forall ?v \ Q))$

where S is a finite set of sentences, P and Q are any sentences, $?v$ is a variable, and there does not exist any free occurrences of $?v$ in P or any member of S

Figure 4.1-1

statements. Further provisions are not needed to handle modal statements. This is not true for all modal systems, however. If we allowed terms whose denotation varied from modal context to modal context, we would have to further qualify *AX-FO7* so that the terms being substituted for did not appear in any modal context. If we considered an interpretation where an individual that existed in one context might not exist in another, we would have to qualify *AX-FO5* so that the term being substituted denotes an object that exists in the context in which it is being substituted.

The two inference rules presented in figure 4.1-1 differ from the ones in Hughes and Cresswell in that we specify our rules in *derivability form*, instead of in *theoremhood form*. For example, they formulate *MP*, which stands for Modus Ponens, as a rule from " $\vdash P$ " and " $\vdash (IF P Q)$ " to " $\vdash Q$ ", rather than a rule from " $S1 \vdash P$ " and " $S2 \vdash (IF P Q)$ " to " $S1 \cup S2 \vdash Q$ ", as given in our system. The reason that we write rules in derivability form, rather than theoremhood form, is to facilitate natural deduction style proofs. All our rules will be written in this general form with the exception of two rules (see section 5.3 and 5.4), sometimes referred to as rules of necessitation, where the *derivability form* would lead to an unsound system.

Formally, we define derivability by relating it to theoremhood:

If $|S| > 1$

" $S \vdash P$ " is true iff " $\vdash (IF (AND s_1 s_2 \dots s_n) P)$ " is true for some sequence s_1, s_2, \dots, s_n making up the set S

If $|S| = 1$

" $\{s_1\} \vdash P$ " is true iff " $\vdash (IF s_1 P)$ " is true

If $S = \emptyset$

" $\emptyset \vdash P$ " is true iff " $\vdash P$ " is true

Typically, theoremhood and derivability (" \vdash " as a binary relation) are defined separately, although in many systems, the equivalences given above hold. For our purposes, it is simpler to just define derivability in terms of theoremhood.

In appendix E, we present the first order theorems and derived inference rules that we will be using in proofs in the following sections. Proofs of these first order theorems and proofs that the first order axioms and inference rules are sound are straightforward and are not given here.

4.2. Axioms Describing the Interval Logic Predicates and Terms

In this and the following sections, we use the following conventions for identifying the types associated with constant and variable terms. Token i or any token prefixed by "i" (i.e. $i1$, $i2$, ...) refer to interval constants; pr or any token prefixed by "pr" refer to property constants; ev or any token prefixed by "ev" refer to event constants, and pi or any token prefixed by "pi" refer to plan instance constants. A similar convention is used to identify the types associated with variables. Token $?i$ or any token prefixed by "?i" refer to interval variables; token $?pr$ or any token prefixed by "?pr" refer to property variables, etc.

The interval logic axioms consist of i) an axiom describing a property of the *HOLDS* predicate, ii) axioms describing *COMP*, the function that is used to denote the composition of two plan instances, iii) axioms describing the *TIME-OF* function which denotes the time of occurrence associated with a plan instance term, and iv) axioms describing the *MEETS* relation, capturing the temporal relations between intervals. These axioms are given in figure 4.2-1.

Axiom *AX-IL1* says that if a property pr holds during some interval $i2$ then pr holds over any interval equal to or contained in $i2$.

Axioms *AX-IL1*, *AX-IL2*, and *AX-IL3* capture that the *COMP* function behaves just like set union. Axiom *AX-IL2* says that a plan instance composed to itself equals itself. *AX-IL3* says that *COMP* is commutative, and *AX-IL4* say that *COMP* is associative. Because of the associative property, we can write the composition of three or more plan instance unambiguously by $\lceil (COMP\ pi1\ pi2\ pi3 \dots pin) \rceil$.

The occurrence of plan instance compositions and the time associated with them are characterized by axioms *AX-IL5* and *AX-IL6*. Axiom *AX-IL5* says that the composition of $pi1$ and $pi2$ occurs iff both $pi1$ and $pi2$ occur together. Axiom *AX-IL6* can be interpreted as saying that the time associated with the composition of $pi1$ and $pi2$ is the "smallest interval that contains" both $pi1$'s and $pi2$'s time of occurrence.

The remainder of the axioms describe the *MEETS* predicate. We do not need to axiomatize any other interval relations since all interval relations are defined in terms of *MEETS* and the first order connectives (see appendix B). The axiomatization of *MEETS*, which is taken from [Allen&Hayes 85], is given by axioms *AX-IR1* - *AX-IR5*. Axiom *AX-IR1* can be interpreted as saying that if $i1$ and $i2$ end together then $i1$ meets any interval $i4$ iff $i2$ meets $i4$. Axiom *AX-IR2* is the symmetric relation for two intervals that begin together. Axiom *AX-IR3* states that there are three possible relations between two different meeting places. Either the meeting places are equal, the meeting place of $i1$ and $i2$ is before the meeting place of $i3$ and $i4$, or the meeting place of $i1$ and $i2$ is after the

Axioms

AX-IL1)

$$\vdash (\text{IF } (\text{IN } i1 \ i2) \\ (\text{IF } (\text{HOLDS } pr \ i2) (\text{HOLDS } pr \ i1))))$$

AX-IL2)

$$\vdash (= (\text{COMP } pi1 \ pi1) \ pi1)$$

AX-IL3)

$$\vdash (= (\text{COMP } pi1 \ pi2) (\text{COMP } pi2 \ pi1))$$

AX-IL4)

$$\vdash (= (\text{COMP } pi1 (\text{COMP } pi2 \ pi3)) (\text{COMP } (\text{COMP } pi1 \ pi2) \ pi3))$$

AX-IL5)

$$\vdash (\text{IFF } (\text{OCC } (\text{COMP } pi1 \ pi2)) (\text{AND } (\text{OCC } pi1) (\text{OCC } pi2))))$$

AX-IL6)

$$\vdash (\text{IFF } (\text{AND } (\text{IN } (\text{TIME-OF } pi1) \ i) (\text{IN } (\text{TIME-OF } pi2) \ i)) \\ (\text{IN } (\text{TIME-OF } (\text{COMP } pi1 \ pi2)) \ i)))$$

AX-IR1)

$$\vdash (\text{IF } (\text{AND } (\text{MEETS } i1 \ i3) (\text{MEETS } i2 \ i3)) \\ (\text{IFF } (\text{MEETS } i1 \ i4) (\text{MEETS } i2 \ i4))))$$

AX-IR2)

$$\vdash (\text{IF } (\text{AND } (\text{MEETS } i1 \ i2) (\text{MEETS } i1 \ i3)) \\ (\text{IFF } (\text{MEETS } i0 \ i2) (\text{MEETS } i0 \ i3))))$$

AX-IR3)

$$\vdash (\text{IF } (\text{AND } (\text{MEETS } i1 \ i2) (\text{MEETS } i3 \ i4)) \\ (\text{XOR } (\text{MEETS } i1 \ i4) \\ (\exists ?ix (\text{AND } (\text{MEETS } i1 \ ?ix) (\text{MEETS } ?ix \ i4)) \\ (\exists ?iy (\text{AND } (\text{MEETS } i3 \ ?iy) (\text{MEETS } ?iy \ i2))))))$$

where $(\text{XOR } P \ Q \ R) =_{\text{def}} (\text{OR } (\text{AND } (\text{NOT } P) \ Q \ R) \\ (\text{AND } P \ (\text{NOT } Q) \ R) \\ (\text{AND } P \ Q \ (\text{NOT } R)))$

AX-IR4)

$$\vdash (\exists ?i0 \ ?i2 (\text{AND } (\text{MEETS } ?i0 \ i1) (\text{MEETS } i1 \ ?i2))))$$

AX-IR5)

$$\vdash (\text{IF } (\text{MEETS } i1 \ i2) \\ (\exists ?ix \ ?iy \ ?iz \\ (\text{AND } (\text{MEETS } ?ix \ i1) (\text{MEETS } i2 \ ?iy) (\text{MEETS } ?ix \ ?iz) (\text{MEETS } ?iz \ ?iy))))$$

Figure 4.2-1

meeting place of $i3$ and $i4$. $AX-IR4$ states that for every interval $i1$, there exists an interval that meets it to the left and one that meets it to the right. $AX-IR5$ says that if interval $i1$ meets $i2$, then there exists an interval ($i2$) corresponding to the concatenation of $i1$ and $i2$.

Soundness Proofs

The proofs that axioms $AX-IL1$ - $AX-IL6$ are valid are given in figures 4.2-2 and 4.2-3. Axioms $AX-IR1$ - $AX-IR5$ directly correspond to analogous constraints placed on the MTS relation in the model. For example, $AX-IR1$ corresponds to:

$MTS1)$

For all intervals ($i1, i2, i3$ and $i4$),
if $MTS(i1, i2)$ and $MTS(i2, i3)$ then $MTS(i1, i4)$ iff $MTS(i2, i4)$

Since the *MEETS* atomic formula is directly interpreted in terms of the MTS relation as given below, it is simple to show that axioms $AX-IR1$ - $AX-IR5$ are valid:

For all wffs of the form $\lceil (MEETS\ t_{int1}\ t_{int2}) \rceil$ and world-histories (h),
 $V_s(\lceil (MEETS\ t_{int1}\ t_{int2}) \rceil, h) = \text{TRUE}$ iff $MTS(V_t(t_{int1}), V_t(t_{int2}))$ is true

Soundness Proofs

AX-IL1)

$$\vdash (IF \quad (IN \ i1 \ i2) \\ (IF \ (HOLDS \ pr \ i2) \ (HOLDS \ pr \ i1)))$$

This axiom arises from constraint PROP1 which is given as follows:

For all properties (pr), intervals (ix and iy), and world-histories (h)
If $IN(ix, iy)$ and $\langle ix, h \rangle \in pr$ then $\langle ix, h \rangle \in pr$

For an arbitrary model, let V_s be its interpretation function and let h be any world-history in H . We compute $V_s(\ulcorner (IN \ i1 \ i2) \urcorner, h)$ by replacing $\ulcorner (IN \ i1 \ i2) \urcorner$ by its definition in terms of the MEETS predicate and then using the interpretation of MEETS which is given by: $V_s(\ulcorner (MEETS \ i1 \ i2) \urcorner, h) = \text{TRUE}$ iff $MTS(V_t(i1), V_t(i2))$ is true. Now, the relation between the predicates IN and $MEETS$ is analogous to the relation between the relations MTS and IN (in the model). Thus, $V_s(\ulcorner (IN \ i1 \ i2) \urcorner, h) = \text{TRUE}$ iff $IN(V_t(i1), V_t(i2))$ is true. Using the interpretation of the HOLDS predicate, we get $V_s(\ulcorner (HOLDS \ pr \ i2) \urcorner, h) = \text{TRUE}$ iff $\langle V_t(i2), h \rangle \in V_t(pr)$. Similarly, $V_s(\ulcorner (HOLDS \ pr \ i1) \urcorner, h) = \text{TRUE}$ iff $\langle V_t(i1), h \rangle \in V_t(pr)$. Using PROP1 we get: if $V_s(\ulcorner (IN \ i1 \ i2) \urcorner, h)$ and $V_s(\ulcorner (HOLDS \ pr \ i2) \urcorner, h)$ both equal TRUE, then $V_s(\ulcorner (HOLDS \ pr \ i2) \urcorner, h)$ equals TRUE thereby validating AX-IL1

AX-IL2)

$$\vdash (= \ (COMP \ pi1 \ pi1) \ pi1)$$

AX-IL3)

$$\vdash (= \ (COMP \ pi1 \ pi2) \ (COMP \ pi2 \ pi1))$$

AX-IL4)

$$\vdash (= \ (COMP \ pi1 \ (COMP \ pi2 \ pi3)) \ (COMP \ (COMP \ pi1 \ pi2) \ pi3))$$

The interpretation of COMP is given by:

$$V_t(\ulcorner (COMP \ pi1 \ pi2) \urcorner) = \langle E1, E2 \rangle$$

$$\text{where } E1 =_{\text{def}} V_t(pi1)|_1 \cup V_t(pi2)|_1 \quad \text{and } E2 =_{\text{def}} V_t(pi1)|_2 \cup V_t(pi2)|_2$$

AX-IL2, AX-IL3, and AX-IL4 are validated using the fact that set union is reflexive, associative and commutative

Figure 4.2-2

Soundness Proofs

AX-IL5)

$\vdash (\text{IFF } (\text{OCC } (\text{COMP } \text{pi1 } \text{pi2})) (\text{AND } (\text{OCC } \text{pi1}) (\text{OCC } \text{pi2})))$

The interpretation of OCC can be given by:

$V_s(\ulcorner (\text{OCC } \text{pi}) \urcorner, h) = \text{TRUE}$ iff for all events (ev), intervals (i) and basic actions (ba)
 if $\langle \text{ev}, i \rangle \in V_t(\text{pi})|_1$, then $\langle i, h \rangle \in \text{ev}$, and
 if $\langle \text{ba}, i \rangle \in V_t(\text{pi})|_2$, then $\langle i, h \rangle \in \text{BAEV}(\text{ba})$

For an arbitrary model, let V_s be its interpretation function and let h be any world-history in H . Assume that $V_s(\ulcorner (\text{OCC } (\text{COMP } \text{pi1 } \text{pi2})) \urcorner, h)$ equals TRUE. $V_s(\ulcorner (\text{OCC } (\text{COMP } \text{pi1 } \text{pi2})) \urcorner, h)$ equals TRUE iff the following holds:

for all events (ev), intervals (i) and basic actions (ba)
 if $\langle \text{ev}, i \rangle \in V_t(\text{pi1})|_1 \cup V_t(\text{pi2})|_1$, then $\langle i, h \rangle \in \text{ev}$, and
 if $\langle \text{ba}, i \rangle \in V_t(\text{pi1})|_2 \cup V_t(\text{pi2})|_2$, then $\langle i, h \rangle \in \text{BAEV}(\text{ba})$

From this we derive

for all events (ev), intervals (i) and basic actions (ba)
 if $\langle \text{ev}, i \rangle \in V_t(\text{pi1})|_1$ then $\langle i, h \rangle \in \text{ev}$ and
 if $\langle \text{ba}, i \rangle \in V_t(\text{pi1})|_2$ then $\langle i, h \rangle \in \text{BAEV}(\text{ba})$
 and
 if $\langle \text{ev}, i \rangle \in V_t(\text{pi2})|_1$ then $\langle i, h \rangle \in \text{ev}$, and
 if $\langle \text{ba}, i \rangle \in V_t(\text{pi2})|_2$ then $\langle i, h \rangle \in \text{BAEV}(\text{ba})$

Thus, if $V_s(\ulcorner (\text{OCC } (\text{COMP } \text{pi1 } \text{pi2})) \urcorner, h)$ equals TRUE both $V_s(\ulcorner (\text{OCC } \text{pi1}) \urcorner, h)$ and $V_s(\ulcorner (\text{OCC } \text{pi2}) \urcorner, h)$ equal TRUE. It is easily seen that this implication goes the other way. Consequently, AX-IL5 is valid.

AX-IL6)

$\vdash (\text{IFF } (\text{AND } (\text{IN } (\text{TIME-OF } \text{pi1}) i) (\text{IN } (\text{TIME-OF } \text{pi2}) i))$
 $(\text{IN } (\text{TIME-OF } (\text{COMP } \text{pi1 } \text{pi2})) i))$

The interpretation of TIME-OF can be given by:

$V_t(\ulcorner (\text{TIME-OF } \text{pi}) \urcorner) = \text{COVER}(\{ i \mid \langle \text{ev}, i \rangle \in V_t(\text{pi})|_1 \})$
 where $\text{COVER}(i\text{-set})$ is the smallest interval that contains all intervals in $i\text{-set}$.

Letting $i1$ equal the set $\{ i \mid \langle \text{ev}, i \rangle \in V_t(\text{pi1})|_1 \}$ and $i2$ equal the set $\{ i \mid \langle \text{ev}, i \rangle \in V_t(\text{pi2})|_1 \}$, we have $V_t(\ulcorner (\text{TIME-OF } \text{pi1}) \urcorner)$ equals $\text{COVER}(i1)$ and $V_t(\ulcorner (\text{TIME-OF } \text{pi2}) \urcorner)$ equals $\text{COVER}(i2)$, and $V_t(\ulcorner (\text{TIME-OF } (\text{COMP } \text{pi1 } \text{pi2})) \urcorner)$ equals $\text{COVER}(i1 \cup i2)$. Using the definition of COVER given in appendix C, we can derive that an interval is in both $\text{COVER}(i1)$ and $\text{COVER}(i2)$ iff it is in $\text{COVER}(i1 \cup i2)$, thereby validating AX-IL6.

Figure 4.2-3

4.3. The Axiomatization of the *INEV* Modal Operator

We break up the axiomatization of *INEV* into two groups. For a fixed interval argument, the axiomatization of *INEV* parallels the axiomatization of a *S5* necessity operator [Hughes and Cresswell 68]. These properties, which we present first, are captured by four axioms and one inference rule. In the second part, we present the axioms that are unique to our system. These pertain to the relation between two *INEV* operators with different time arguments and the relation between *INEV* and interval logic statements. We wait until section 4.4 to give the relation between *INEV* and *IFTRIED*.

4.3.1. *S5* Properties

The *INEV* modal operator behaves as a *S5* necessity modality for a fixed interval argument. One can think of a necessity operator as a universal quantifier over "accessible" possible worlds, or branches or world-histories in our case. Thus, we can read $\lceil (INEV\ i\ P) \rceil$ as "*P* is true in all branches that are possible at time *i*". The possibility modal operator, which we have defined as the dual of *INEV* for a fixed time, can be thought of an existential quantifier over branches. This parallels the relation between the universal and existential quantifiers. By characterizing *INEV* as *S5* we are saying that inevitability and possibility are invariant over different branches. That is, whatever is inevitable is inevitably inevitable, and whatever is possible is inevitably possible. The axiomatization of *INEV* as a *S5* necessity operator is taken from [Hughes and Cresswell 68] and is given in figure 4.3-1.

Axiom *AX-INV1* says that whatever is inevitable at any time is also actual. *AX-INV2* says that material implication distributes out of *INEV*. Axiom *AX-INV3* says that if *P* is inevitable at time *i*, then it is inevitable at *i* that *P* is inevitable at *i*, and *AX-INV4* says that if *P* is possible at time *i*, then it is inevitable at *i* that *P* is possible at *i*.

The inference rule involving *INEV* can be referred to as a rule of necessitation. This rule can be read as saying that if *P* is a theorem then $\lceil (INEV\ i\ P) \rceil$ is a theorem for any interval term *i*. It is important to note that *RL-INV* is given in *theoremhood form* rather than in *derivability form*, i.e. from "*S* ⊢ *P*" to "*S* ⊢ (*INEV* *i* *P*)". A rule of necessitation having derivability form is unsound since it could be used to derive that $\lceil (IF\ P\ (INEV\ i\ P)) \rceil$ is a theorem for any sentence *P*. This statement, which says whatever is actual is inevitable, is not valid with respect to our semantics.

Soundness Proofs for *INEV* is *S5* modality

The proofs that axioms *AX-INV1* - *AX-INV4* are valid are given in figures 4.3-2 and 4.3-3. These proofs are established using the interpretation of *INEV* and the constraints on the *R* relation (which is used to interpret *INEV*) that state that *R* is an equivalence relation for a set interval argument (see section 3.2.2 or appendix C).

Axioms

AX-INV1)

$$\vdash (\text{IF } (\text{INEV } i P) P)$$

AX-INV2)

$$\vdash (\text{IF } (\text{INEV } i (\text{IF } P Q)) (\text{IF } (\text{INEV } i P) (\text{INEV } i Q)))$$

AX-INV3)

$$\vdash (\text{IF } (\text{INEV } i P) (\text{INEV } i (\text{INEV } i P)))$$

AX-INV4)

$$\vdash (\text{IF } (\text{POS } i P) (\text{INEV } i (\text{POS } i P)))$$
Inference Rule

RL-INV)

From: $\vdash P$ To: $\vdash (\text{INEV } i P)$

Figure 4 3-1

The proof that *RL-INV* preserves validity is easily established by noting that if statement *P* is valid then it is true in all models at each world-history. Now, $\vdash (\text{INEV } i P)$ is interpreted as true at world-history *h* iff *P* is true in all world-histories that are *R*-accessible to *h* at the time denoted by *i*. Thus, if *P* is valid, it is true at all world-histories, these including the ones that are *R*-accessible at the time denoted by *i*.

Derived Theorems and Rules for INEV is S5 modality

The following theorems and derived inference rules are derivable from *AX-INV1* - *AX-INV4*, inference rule *RL-INV*, along with the axioms and inference rules for the first order connectives (*AX-FO1* - *AX-FO8* and inference rules *MP* and *UNV-INTRO*). We provide proofs only for the theorems and derived rules that are not analogous to proofs in Hughes and Cresswell. The theorems and derived rules that we present can be divided into four categories corresponding to: i) the relation between *POS* and *INEV*, ii) deduction within a modal context, iii) nested modal statements with the same temporal argument, and iv) distributing first order connectives in and out of *INEV* and *POS*.

Soundness Proofs

The interpretation of INEV, which is used in all the proofs in this figure and 4 3-3, can be given by:

$V_s(\ulcorner \text{INEV} \mid P \urcorner, h) = \text{TRUE}$ iff
for all world-histories $(h2)$ if $R(V_t(i), h, h2)$ then $V_s(P, h2) = \text{TRUE}$

The interpretation of $\ulcorner \text{POS} \mid P \urcorner$ which is defined as $\ulcorner (\text{NOT} (\text{INEV} \mid (\text{NOT } P))) \urcorner$ is used in the proof of AX-INV4 and can be given by:

$V_s(\ulcorner \text{POS} \mid P \urcorner, h) = \text{TRUE}$ iff
there exists a world-history $(h2)$ such that $R(V_t(i), h, h2)$ and $V_s(P, h2) = \text{TRUE}$

AX-INV1)

$\vdash (\text{IF} (\text{INEV} \mid P) \mid P)$

For an arbitrary V_s and world-history h , assume that $V_s(\ulcorner \text{INEV} \mid P \urcorner, h)$ equals TRUE. We validate AX-INV1 by showing that $V_s(P, h)$ equals TRUE under our assumption. This is established using constraint R2 (R is symmetric) and R3 (R is transitive) to derive that R is reflexive. In particular $R(V_t(i), h, h)$ is true. Now, since $V_s(\ulcorner \text{INEV} \mid P \urcorner, h)$ equals TRUE, we have: for all world-histories $(h2)$ if $R(V_t(i), h, h2)$ then $V_s(P, h2) = \text{TRUE}$. This relation in conjunction with " $R(V_t(i), h, h)$ is true" yields that $V_s(P, h) = \text{TRUE}$

AX-INV2)

$\vdash (\text{IF} (\text{INEV} \mid (\text{IF } P \mid Q)) \mid (\text{IF} (\text{INEV} \mid P) \mid (\text{INEV} \mid Q)))$

For an arbitrary V_s and world-history h , assume that both $V_s(\ulcorner \text{INEV} \mid (\text{IF } P \mid Q) \urcorner, h)$ and $V_s(\ulcorner \text{INEV} \mid P \urcorner, h)$ equal TRUE. We validate AX-INV2 by showing that $V_s(\ulcorner \text{INEV} \mid Q \urcorner, h)$ equals TRUE under our assumptions. Using $V_s(\ulcorner \text{INEV} \mid (\text{IF } P \mid Q) \urcorner, h) = \text{TRUE}$, we get:

for all world-histories $(h2)$ if $R(V_t(i), h, h2)$ then $V_s(\ulcorner \text{IF } P \mid Q \urcorner, h2) = \text{TRUE}$

Using $V_s(\ulcorner \text{INEV} \mid P \urcorner, h) = \text{TRUE}$, we get:

for all world-histories $(h2)$ if $R(V_t(i), h, h2)$ then $V_s(P, h2) = \text{TRUE}$

Taking this two relations together (and using the interpretation of IF), we get the following which is true iff $V_s(\ulcorner \text{INEV} \mid Q \urcorner, h)$ is TRUE:

for all world-histories $(h2)$ if $R(V_t(i), h, h2)$ then $V_s(Q, h2) = \text{TRUE}$

Figure 4.3-2

Soundness Proofs

AX-INV3)

$$\vdash (IF (INEV \vdash P) (INEV \vdash (INEV \vdash P)))$$

For an arbitrary V_s and world-history h , assume that $V_s(\ulcorner(INEV \vdash P)\urcorner, h)$ equals TRUE. We validate AX-INV3 by showing that $V_s(\ulcorner(INEV \vdash (INEV \vdash P))\urcorner, h)$ equals TRUE under our assumption. This is established using constraint R3 which says that R is transitive and entails the following relation:

For all world-histories (hx, and h2)
if $R(V_t(i), h, hx)$ and $R(V_t(i), hx, h2)$ then $R(V_t(i), h, h2)$:

Now, since $V_s(\ulcorner(INEV \vdash P)\urcorner, h)$ equals TRUE, we have:

For all world-histories (h2) if $R(V_t(i), h, h2)$ then $V_s(P, h2) = \text{TRUE}$.

Taking the two relations above together we derive the following which is true iff $V_s(\ulcorner(INEV \vdash (INEV \vdash P))\urcorner, h)$ equals TRUE:

For all world-histories (hx) if $R(V_t(i), h, hx)$ then for all world-histories (h2)
if $R(V_t(i), hx, h2)$ then $V_s(P, h2) = \text{TRUE}$

AX-INV4)

$$\vdash (IF (POS \vdash P) (INEV \vdash (POS \vdash P)))$$

For an arbitrary V_s and world-history h , assume that $V_s(\ulcorner(POS \vdash P)\urcorner, h)$ equals TRUE. We validate AX-INV3 by showing that $V_s(\ulcorner(INEV \vdash (POS \vdash P))\urcorner, h)$ equals TRUE under our assumption. This is established using the following relation which is derived from R2 (R is symmetric) and R3 (R is transitive)

- 1) For all world-histories (hx and hy)
if $R(V_t(i), h, hx)$ and $R(V_t(i), h, hy)$ then $R(V_t(i), hx, hy)$

Now, since $V_s(\ulcorner(POS \vdash P)\urcorner, h)$ equals TRUE, we have:

- 2) There exists a world-history (h2) such that $R(V_t(i), h, h2)$ and $V_s(P, h2) = \text{TRUE}$.

Substituting the world-history (h2) meeting the property in 2 for hy in 1 gives us the following which is true iff $V_s(\ulcorner(INEV \vdash (POS \vdash P))\urcorner, h)$ equals TRUE

For all world-histories (hx) if $R(V_t(i), h, hx)$ then
there exists a world-history (h2) such that $R(V_t(i), hx, h2)$ and $V_s(P, h2) = \text{TRUE}$

Figure 4.3-3

The relation Between POS and INEV

The definition of $\lceil (POS\ i\ P) \rceil$ was given as $\lceil (NOT\ (INEV\ i\ (NOT\ P))) \rceil$. In other words, *POS* is the dual of *INEV* for a fixed time point. The following theorem captures that duality is a symmetrical relation:

TH-INV-SF1)

$\vdash (IFF\ (INEV\ i\ P)\ (NOT\ (POS\ i\ (NOT\ P))))$

The three following theorems will be used in many later proofs:

TH-INV-SF2)

$\vdash (IF\ P\ (POS\ i\ P))$

TH-INV-SF3)

$\vdash (IF\ (INEV\ i\ P)\ (POS\ i\ P))$

TH-INV-SF4)

$\vdash (IF\ (INEV\ i\ (IF\ P\ Q))\ (IF\ (POS\ i\ P)\ (POS\ i\ Q)))$

TH-INV-SF2 says that whatever is actual is possible at any time. This theorem is derived substituting $\lceil (NOT\ P) \rceil$ for *P* in axiom *AX-INV1* and transposing the implication. Theorem *TH-INV-SF3* says that whatever is inevitable at time *i* is possible at time *i*. This theorem may be derived by chaining together *TH-INV-SF2* and *AX-INV1*. Finally, *TH-INV-SF4* says that if the implication from *P* to *Q* is inevitable at time *i*, then if *P* is possible at *i*, then *Q* is possible at *i*.

Deduction within a Modal Context

An important property of the *INEV* operator is that if statement *Q* is entailed by statement *P*, then $\lceil (INEV\ i\ Q) \rceil$ is entailed by $\lceil (INEV\ i\ P) \rceil$. This is very useful when doing proofs for our planning examples. Many of the proofs require a derivation from a sentence of the form: $\lceil (INEV\ i\ IL1) \rceil$ to a sentence of the form $\lceil (INEV\ i\ IL2) \rceil$ where *IL1* and *IL2* are interval logic sentences. We can prove that $\lceil (INEV\ i\ IL2) \rceil$ follows from $\lceil (INEV\ i\ IL1) \rceil$ by proving that *IL1* follows from *IL2*. This latter step only involves derivation in first order logic. Haas [Haas 85], who has an operator similar to *INEV*, elaborates on this argument and claims that most of the modal reasoning needed to do planning (in his system) requires only first order theorems applied within some modal context. The same argument can be used in our case.

We present a derived rule that captures a generalization of the above relation for any modal context formed by nested *POS* and *INEV* statements. To describe this rule, we refer to a class of functions from sentence to sentence, which we call **modal chains**. A modal chain is any function that when applied to a sentence *P* yields a nesting of *INEV* and *POS* statement with *P* embedded on the inside. The operators in a chain need not have the same interval argument. An example of a modal chain

is $MC1$ where $MC1(P)$ is defined as $\lceil (INEV\ i1\ (POS\ i2\ P)) \rceil$ for any sentence P . Using this terminology, we can succinctly describe our derived inference rule by:

DRL-INV1)

From: $\vdash (IF\ P\ Q)$

To: $\vdash (IF\ MC(P)\ MC(Q))$

where MC is any modal chain formed by $INEV$ and POS operators

The derivation of $DRL-INV1$ is given in appendix F. A more general rule is presented in section 4.4 that covers the cases where $IFTRIED$ statements are part of the modal chain. Now, rule $DRL-INV1$, just like $RL-INV$, is given in *theoremhood form*, not *derivability form*. Adopting this more general form would render our system unsound. That is, we do not specify $DRL-INV1$ as a rule from " $S \vdash (IF\ P\ Q)$ " to " $S \vdash (IF\ MC(P)\ MC(Q))$ ". If this were done, we could derive that invalid statements, such as $\lceil (IF\ P\ (INEV\ i\ P)) \rceil$, are theorems.

A second rule that is useful for working within modal contexts allows us to substitute equivalent statements nested within the scope of $INEV$ and POS operators. This is a generalization of the first order rule $SUBST$ which applies to the substitution of any equivalent sentences that are not nested under any modal context. Our derived rule is given by:

DRL-INV2)

From: $\vdash (IFF\ P\ Q)$

To: $\vdash (IFF\ R1\ R2)$

where $R1$ and $R2$ are sentences in the interval logic fragment extended with $INEV$ (which includes POS since it is defined in terms of $INEV$), and $R2$ differs from $R1$ by replacing one or more occurrences of P with Q

We give a derivation of $DRL-INV2$ in appendix F and present a more general rule in section 4.4 that covers the cases where substitution may be under the scope of an $IFTRIED$ operator. We also make use of the following derived rule which is easily derived from $DRL-INV2$ (see appendix F):

DRL-INV3)

From: $\vdash (IFF\ P\ Q)$ and $S \vdash R1$

To: $S \vdash R2$

where S is a finite set of sentences, $R1$ and $R2$ are sentences in the interval logic fragment extended with $INEV$, and $R2$ differs from $R1$ by replacing one or more occurrences of P with Q

Nested operators with the same temporal index

Any iterated chain of S5 necessity and possibility operators can be collapsed into a single operator. This is reflected in our system by the following set of axioms:

TH-INV-SF5)

$\vdash (\text{IFF } (\text{INEV } i (\text{INEV } i P)) (\text{INEV } i P))$

TH-INV-SF6)

$\vdash (\text{IFF } (\text{INEV } i (\text{POS } i P)) (\text{POS } i P))$

TH-INV-SF7)

$\vdash (\text{IFF } (\text{POS } i (\text{INEV } i P)) (\text{INEV } i P))$

TH-INV-SF8)

$\vdash (\text{IFF } (\text{POS } i (\text{POS } i P)) (\text{POS } i P))$

Theorems *TH-INV-SF5* and *TH-INV-SF8* are both derived using axiom *AX-INV1*, which says whatever is inevitable at *i* is actual and *AX-INV3*, which says whatever is inevitable at *i* is inevitably inevitable at *i*. Theorems *TH-INV-SF6* and *TH-INV-SF7* are both derived using *AX-INV1* and *AX-INV4*, the latter saying, whatever is possible at *i* is inevitably possible at *i*. A chain of three or more *INEV* and *POS* operators with the same temporal index can be reduced to a single operator by successively applying *DRL-INV2* using the equivalences given above. For example, $\lceil (\text{POS } i (\text{INEV } i (\text{POS } i P))) \rceil$ can be reduced to $\lceil (\text{POS } i P) \rceil$ by first using equivalence *TH-INV-SF6* to reduce it to $\lceil (\text{POS } i (\text{POS } i P)) \rceil$ and then using equivalence *TH-INV-SF7* to reduce this intermediate form to $\lceil (\text{POS } i P) \rceil$. In general, a chain consisting of *POS* and *INEV* operators can be reduced to the operator that is nested at the innermost layer.

Interaction with the First Order Connectives

We describe how the first order connectives interact with *INEV* and *POS* by presenting axioms showing whether they distribute in and/or out of these modal contexts. Proofs of analogous theorems can be found in Hughes and Cresswell. We only present the proof associated with the universal quantifier to illustrate the additional complications stemming from the fact that *INEV* and *POS* have a term argument along with a sentence argument, not just a sentence argument as in Hughes and Cresswell.

The relations for conjunction are given by:

TH-INV-SF9)

$\vdash (\text{IFF } (\text{AND } (\text{INEV } i P) (\text{INEV } i Q))$
 $(\text{INEV } i (\text{AND } P Q)))$

TH-INV-SF10)

$$\vdash (\text{IF}(\text{POS } i (\text{AND } P Q)) \\ (\text{AND}(\text{POS } i P)(\text{POS } i Q)))$$

TH-INV-SF9 says that conjunction distributes in and out of *INEV*. By using this theorem along with *DRL-INV1*, we can derive that $\lceil (\text{INEV } i P1) \rceil$ is entailed by $\lceil (\text{INEV } i P2) \rceil$ and $\lceil (\text{INEV } i P3) \rceil$ taken together, if *P1* is entailed by *P2* and *P1* taken together. This is a generalization of the entailment between two sentences that we noted earlier.

Theorem *TH-INV-SF10* says that conjunction distributes out of *POS*. The opposite relation does not hold, however. Conjunction does not necessarily distribute into *POS*. An example of this is when both *P* and $\lceil (\text{NOT } P) \rceil$ are possible at *i*.

The relations between the modal operators and disjunction are given by:

TH-INV-SF11)

$$\vdash (\text{IF}(\text{OR}(\text{INEV } i P)(\text{INEV } i Q)) \\ (\text{INEV } i (\text{OR } P Q)))$$

TH-INV-SF12)

$$\vdash (\text{IFF } (\text{OR}(\text{POS } i P)(\text{POS } i Q)) \\ (\text{POS } i (\text{OR } P Q)))$$

These relations are *symmetric* to the relations given in *TH-INV-SF9* and *TH-INV-SF10*. Theorem *TH-INV-SF11*, which says that disjunction distributes into *INEV*, is symmetric to *TH-INV-SF10*. That is, theorem *TH-INV-SF11* can be derived by substituting $\lceil (\text{NOT } P) \rceil$ for *P* in *TH-INV-SF10* and then transposing the material implication; the symmetric derivation also holds. Similarly, theorem *TH-INV-SF12*, which says that disjunction distributes in and out of *POS*, is symmetric to *TH-INV-SF9*.

The relations between negation and the modal operators are given by:

TH-INV-SF13)

$$\vdash (\text{IF}(\text{INEV } i (\text{NOT } P)) \\ (\text{NOT}(\text{INEV } i P)))$$

TH-INV-SF14)

$$\vdash (\text{IF}(\text{NOT}(\text{POS } i P)) \\ (\text{POS } i (\text{NOT } P)))$$

TH-INV-SF13 says that negation can be brought out of *INEV*, and *TH-INV-SF14* says that negation can be brought into *POS*. These two relations are symmetric to each other. The opposite relations to *TH-INV-SF13* and *TH-INV-SF14* (i.e. negation being brought into *INEV* and negation being brought out of *POS*) do not hold if both *P* and $\lceil (\text{NOT } P) \rceil$ are possible at *i*.

In our system, the relation between the modalities and the two quantifiers parallels the relation between the modalities and conjunction and disjunction, \forall behaving like conjunction and \exists behaving like disjunction. These relations are given as follows:

TH-INV-SF15)

$$\vdash (\text{IFF } (\forall?v (\text{INEV } i P)) \\ (\text{INEV } i (\forall?v P)))$$

where $?v$ does not appear in i

TH-INV-SF16)

$$\vdash (\text{IF } (\text{POS } i (\forall?v P)) \\ (\forall?v (\text{POS } i P)))$$

where $?v$ does not appear in i

TH-INV-SF17)

$$\vdash (\text{IF } (\exists?v (\text{INEV } i P)) \\ (\text{INEV } i (\exists?v P)))$$

where $?v$ does not appear in i

TH-INV-SF18)

$$\vdash (\text{IFF } (\exists?v (\text{POS } i P)) \\ (\text{POS } i (\exists?v P)))$$

where $?v$ does not appear in i

In appendix F, we present a proof of *TH-INV-SF15* to show where the provision " $?v$ does not appear in i " arises from. Without this provision, we could use *TH-INV-SF15* to derive that $\lceil (\text{IFF } (\exists?i (\text{INEV } ?i P) (\forall?i (\text{INEV } ?i P))) \rceil$ is a theorem. This sentence, which is not a theorem, says whatever is inevitable at some time is inevitable at all times (note: The implication from *TH-INV-SF15*'s antecedent to consequent is not a theorem in some systems where the set of objects that exists vary from world to world (see [Hughes and Cresswell 68])).

The opposite relation to *TH-INV-SF17* (the existential quantifier moves out of *INEV*) does not necessarily hold. This has unfortunate consequences for a simple skolemization scheme, such as the one used for first order logic. We cannot simply replace every existential (that is not preceded by a universal) with a skolem constant. If this were done, the distinction between $\lceil (\exists?v (\text{INEV } i P)) \rceil$ and $\lceil (\text{INEV } i (\exists?v P)) \rceil$ would be lost. For example, $\lceil (\exists?i2 (\text{INEV } i (\text{HOLDS pr } ?i2))) \rceil$ and $\lceil (\text{INEV } i (\exists?i2 (\text{HOLDS pr } ?i2))) \rceil$ would both be mapped to $\lceil (\text{INEV } i (\text{HOLDS pr } sk)) \rceil$ for some skolem constant sk . This problem can be avoided in Moore's system [Moore 80], which is a first order theory that describes the interpretation of a modal logic and consequently has terms denoting possible worlds. Very roughly, the distinction between $\lceil (\exists?i2 (\text{INEV } i (\text{HOLDS pr } ?i2))) \rceil$ and $\lceil (\text{INEV } i (\exists?i2 (\text{HOLDS pr } ?i2))) \rceil$ is

captured by the distinction between a skolem function with an argument that ranges over possible worlds and a skolem constant (see [Moore 80] for details).

We conclude this section with the relation between equality and the modal operators. This relation can be given by:

TH-INV-SF19)

$$\vdash (\text{IF}(\text{POS } i (= t1 \ t2)) \\ (\text{INEV } i (= t1 \ t2)))$$

Theorem *TH-INV-SF19* can be interpreted as saying that the equality relation is invariant over possible branches at time i . This relation would not be valid in a system in which the denotation of terms varies from branch to branch. By using *TH-INV-SF19*, axiom *AX-INV1*, and theorem *TH-INV-SF2*, we can derive the following relation, which will be used in later proofs:

TH-INV-SF20)

$$\vdash (\text{IFF } (= t1 \ t2) \\ (\text{INEV } i (= t1 \ t2)))$$

Theorem *TH-INV-SF20* says that two terms are equal iff it is inevitable at i that they are equal. By substituting a variable that does not appear in $t1$ or $t2$ for i in *TH-INV-SF20* and then applying *UNV-INTRO*, we can derive that two terms are equal iff it is inevitable at all times that they are equal. We can also derive that if it is inevitable at some time that $t1$ and $t2$ are equal then it is inevitable at all times that they are equal.

4.3.2. Temporal Properties

In this section, we present the axioms describing to the temporal properties of *INEV*. This concerns the relation between two modal operators with different temporal arguments and the relation between the modalities and the interval logic predicates. These properties are captured by the axioms in figure 4.3-4.

Axiom *AX-INV5* says that whatever is inevitable at some interval $i1$ is inevitable at any interval that ends at the same time or after interval $i1$. Thus, the set of sentences that are inevitable grows as time moves on. We compare the end of intervals because we are interpreting $\lceil (\text{INEV } i \ P) \rceil$ as meaning "regardless of which possible events happen after i , P is true". Thus, the end of an interval argument is all that is used in determining the truth value of an *INEV* statement. If two intervals $i1$ and $i2$ end at the same time, then $\lceil (\text{INEV } i1 \ P) \rceil$ is true iff $\lceil (\text{INEV } i2 \ P) \rceil$ is true. This can be easily derived using axiom *AX-INV5*.

Axioms *AX-INV6*, *AX-INV7*, and *AX-INV8* capture the fact that any condition holding or occurring over an interval that ends earlier than or at the same time as i is inevitable at i if it is possible at i . This is equivalent to saying that any condition holding or occurring over an interval that ends earlier than or at the same time as i

<u>Axioms</u>	
AX-INV5)	
┌ (IF	(ENDS \leq i1 i2)
	(IF (INEV i1 P)
	(INEV i2 P)))
AX-INV6)	
┌ (IF	(ENDS \leq i1 i2)
	(IF (POS i2 (HOLDS pr i1))
	(INEV i2 (HOLDS pr i1))))
AX-INV7)	
┌ (IF	(ENDS \leq i1 i2)
	(IF (POS i2 (OCCURS ev i1))
	(INEV i2 (OCCURS ev i1))))
AX-INV8)	
┌ (IF	(ENDS \leq (TIME-OF pi) i)
	(IF (POS i (OCC pi))
	(INEV i (OCC pi))))
AX-INV9)	
┌ (IF	(POS i (MEETS i1 i2))
	(INEV i (MEETS i1 i2)))

Figure 4.3-4

is inevitably true or inevitably false at i . Axiom *AX-INV9* captures the fact that the interval relation *MEETS* is either inevitably true or inevitably false at any interval i .

Soundness Proofs

The proofs that axioms *AX-INV5* - *AX-INV9* are valid are given in figures 4.3-5, 4.3-6, and 4.3-7

Soundness Proofs

The interpretations of INEV and POS, which are used in the proofs in this figure and 3.3-6, and can be given by:

$V_s(\ulcorner \text{INEV } i \text{ P} \urcorner, h) = \text{TRUE}$ iff
for all world-histories (h_2) if $R(V_t(i), h, h_2)$ then $V_s(P, h_2) = \text{TRUE}$

$V_s(\ulcorner \text{POS } i \text{ P} \urcorner, h) = \text{TRUE}$ iff
there exists a world-history (h_2) such that $R(V_t(i), h, h_2)$ and $V_s(P, h_2) = \text{TRUE}$

AX-INV5)

$\vdash (\text{IF } (\text{ENDS} \leq i_1 i_2) \\ (\text{IF } (\text{INEV } i_1 P) (\text{INEV } i_2 P)))$

For an arbitrary V_s and world-history h , assume that both $V_s(\ulcorner \text{ENDS} \leq i_1 i_2 \urcorner, h)$ and $V_s(\ulcorner \text{INEV } i_1 P \urcorner, h)$ equal TRUE. We validate AX-INV5 by showing that $V_s(\ulcorner \text{INEV } i_2 P \urcorner, h)$ equals TRUE under our assumptions. This is established using constraints R1 and R4 to derive:

- 1) For all world-histories (h_2)
if $\text{ENDS} \leq (V_t(i_1), V_t(i_2))$ and $R(V_t(i_2), h, h_2)$ then $R(V_t(i_1), h, h_2)$

Now, the relation between the relations $\text{ENDS} \leq$ and MTS (in the model) is analogous to the relation between the $\text{ENDS} \leq$ and MEETS predicates. Thus, from $V_s(\ulcorner \text{ENDS} \leq i_1 i_2 \urcorner, h)$ and $V_s(\ulcorner \text{INEV } i_1 P \urcorner, h)$ equal TRUE we get:

- 2) $\text{ENDS} \leq (V_t(i_1), V_t(i_2))$ is true.
- 3) for all world-histories (h_2) if $R(V_t(i_1), h, h_2)$ then $V_s(P, h_2) = \text{TRUE}$.

Taking 1-3 together leads to the following, which is true iff $V_s(\ulcorner \text{INEV } i_2 P \urcorner, h)$ equals TRUE:

for all world-histories (h_2) if $R(V_t(i_2), h, h_2)$ then $V_s(P, h_2) = \text{TRUE}$

AX-INV6)

$\vdash (\text{IF } (\text{ENDS} \leq i_1 i_2) \\ (\text{IF } (\text{POS } i_2 (\text{HOLDS } pr \ i_1)) (\text{INEV } i_2 (\text{HOLDS } pr \ i_1))))$

For an arbitrary V_s and world-history h , assume that both $V_s(\ulcorner \text{ENDS} \leq i_1 i_2 \urcorner, h)$ and $V_s(\ulcorner \text{POS } i_2 (\text{HOLDS } pr \ i_1) \urcorner, h)$ equal TRUE. We validate AX-INV6 by showing that $V_s(\ulcorner \text{INEV } i_2 (\text{HOLDS } pr \ i_1) \urcorner, h)$ equals TRUE under our assumptions. This is established using constraint R5 which entails:

- 1) For all world-histories $(h_1 \text{ and } h_2)$ if $\text{ENDS} \leq (V_t(i_1), V_t(i_2))$ and $R(V_t(i_2), h_1, h_2)$ then $\langle V_t(i_1), h_1 \rangle \in V_t(pr)$ iff $\langle V_t(i_1), h_2 \rangle \in V_t(pr)$

From $V_s(\ulcorner \text{ENDS} \leq i_1 i_2 \urcorner, h)$ equals TRUE we get:

- 2) $\text{ENDS} \leq (V_t(i_1), V_t(i_2))$ is true.

Soundness proof for AX-INV6 continued in 4.3-6

Figure 4.3-5

Soundness Proofs

Continuation of AX-INV6 soundness proof

From 1 and 2, we can derive:

- 3) For all world-histories (h_1 and h_2) if $R(V_t(i_2), h_1, h_2)$
then $\langle V_t(i_1), h_1 \rangle \in V_t(pr)$ iff $\langle V_t(i_1), h_2 \rangle \in V_t(pr)$

From $V_s(\ulcorner (POS\ i_2\ (HOLDS\ pr\ i_1)) \urcorner, h)$ equals TRUE we get:

- 4) There exists a world-history (h_1) such that $R(V_t(i_2), h, h_1)$ and
 $\langle V_t(i_1), h_1 \rangle \in V_t(pr)$

From 3, 4, and R2 and R3 (R is an equivalence relation for a fixed time argument), we can derive the following which is true iff $V_s(\ulcorner (INEV\ i_2\ (HOLDS\ pr\ i_1)) \urcorner, h)$ equals TRUE:

For all world-histories (h_2), if $R(V_t(i_2), h, h_2)$ then $\langle V_t(i_1), h_2 \rangle \in V_t(pr)$

AX-INV7)

- $\vdash (IF\ (ENDS \leq i_1\ i_2)\ (IF\ (POS\ i_2\ (OCCURS\ ev\ i_1))\ (INEV\ i_2\ (OCCURS\ ev\ i_1))))$

The proof is just like the proof for AX-INV6, with the exception that R6 is used for R5. R6 entails the following relation:

For all world-histories (h_1 and h_2) if $ENDS \leq (V_t(i_1), V_t(i_2))$ and $R(V_t(i_2), h_1, h_2)$ then
 $\langle V_t(i_1), h_1 \rangle \in V_t(ev)$ iff $\langle V_t(i_1), h_2 \rangle \in V_t(ev)$

AX-INV8)

- $\vdash (IF\ (ENDS \leq (TIME-OF\ pi)\ i)\ (IF\ (POS\ i\ (OCC\ pi))\ (INEV\ i\ (OCC\ pi))))$

For an arbitrary V_s and world-history h , assume that both $V_s(\ulcorner (ENDS \leq (TIME-OF\ pi)\ i) \urcorner, h)$ and $V_s(\ulcorner (POS\ i\ (OCC\ pi)) \urcorner, h)$ equal TRUE. We validate AX-INV8 by showing that $V_s(\ulcorner (INEV\ i\ (OCC\ pi)) \urcorner, h)$ equals TRUE under our assumptions. From $V_s(\ulcorner (ENDS \leq (TIME-OF\ pi)\ i) \urcorner, h) = TRUE$, we get:

- 1) $ENDS \leq (i-cov, V_t(i))$ where $i-cov =_{def} COVER(\{ix \mid \langle ev, ix \rangle \in V_t(pi)|_1\})$

The interval $i-cov$ is the smallest interval that contains all the intervals in $\{ix \mid \langle ev, ix \rangle \in V_t(pi)|_1\}$. Thus, any interval in $\{ix \mid \langle ev, ix \rangle \in V_t(pi)|_1\}$ is contained in or equals $i-cov$. Using constraint P11, we derive that any interval in $\{ix \mid \langle ba, ix \rangle \in V_t(pi)|_2\}$ is contained in or equals $i-cov$. Using the interval relation "if $ENDS \leq (ic, i_2)$ and ix is contained in or equals ic , then $ENDS \leq (ix, i_2)$ ", we get:

- 2) for all events (ev), intervals (ix) and basic actions (ba)
if $\langle ev, ix \rangle \in V_t(pi)|_1$ then $ENDS \leq (ix, V_t(i))$ and
if $\langle ba, ix \rangle \in V_t(pi)|_2$ then $ENDS \leq (ix, V_t(i))$

Soundness proof for AX-INV8 continued in 4.3-7

Figure 4.3-6

Soundness Proofs

Continuation of AX-INV8 soundness proof

From constraint R6, we get:

- 3) For all world-histories (h1 and h2), events (evx) , and intervals (ix),
if $ENDS \leq (ix, V_t(i))$ and $R(V_t(i), h1, h2)$ then $\langle ix, h1 \rangle \in evx$ iff $\langle ix, h2 \rangle \in evx$

From 2 and 3 used twice with the substitutions {ev/evx, BAEV(ba)/evx}:

- 4) For all world-histories (h1 and h2), if $R(V_t(i), h1, h2)$
then for all events (ev), intervals (ix) and basic actions (ba)
if $\langle ev, ix \rangle \in V_t(pi)|_1$ then $\langle ix, h1 \rangle \in ev$ iff $\langle ix, h2 \rangle \in ev$
if $\langle ba, ix \rangle \in V_t(pi)|_2$ then $\langle ix, h1 \rangle \in BAEV(ba)$ iff $\langle ix, h2 \rangle \in BAEV(ba)$

From $V_s(\ulcorner (POS\ i\ (OCC\ pi)) \urcorner, h) = TRUE$, we get:

- 5) There exists a world-history (h1) such that $R(V_t(i), h, h1)$ and
for all events (ev), intervals (ix) and basic actions (ba)
if $\langle ev, ix \rangle \in V_t(pi)|_1$, then $\langle ix, h1 \rangle \in ev$, and
if $\langle ba, ix \rangle \in V_t(pi)|_2$, then $\langle ix, h1 \rangle \in BAEV(ba)$

From 4,5 and R2 and R3 (R is an equivalence relation for a fixed time argument)
we can derive the following which is true iff $V_s(\ulcorner (INEV\ i\ (OCC\ pi)) \urcorner, h) = TRUE$

For all world-histories (h2) if $R(V_t(i), h, h2)$ then
for all events (ev), intervals (ix) and basic actions (ba)
if $\langle ev, ix \rangle \in V_t(pi)|_1$, then $\langle ix, h2 \rangle \in ev$, and
if $\langle ba, ix \rangle \in V_t(pi)|_2$, then $\langle ix, h2 \rangle \in BAEV(ba)$

AX-INV9)

$\vdash (IF\ (POS\ i\ (MEETS\ i1\ i2))\ (INEV\ i\ (MEETS\ i1\ i2)))$

This is easily established because the interpretation of $\ulcorner (MEETS\ i1\ i2) \urcorner$ does not mention a world-history, i.e. $V_s(\ulcorner (MEETS\ i1\ i2) \urcorner, h) = MTS(V_t(i1), V_t(i2))$. Thus, if $V_s(\ulcorner (POS\ i\ (MEETS\ i1\ i2)) \urcorner, h) = TRUE$ which is equivalent to:

there exists a world-history (h2) such that $R(V_t(i), h, h2)$ and $MTS(V_t(i1), V_t(i2))$

from this we get $V_s(\ulcorner (INEV\ i\ (MEETS\ i1\ i2)) \urcorner, h) = TRUE$, since it is equivalent to

for all world-histories (h2) if $R(V_t(i), h, h2)$ then $MTS(V_t(i1), V_t(i2))$

Figure 4.3-7

Theorems

We divide the theorems that we are presenting into two groups. The first set describes the relation between interval logic statements (i.e. non-modal statements) and the *INEV* modality. They are derived using axioms *AX-INV6* - *AX-INV9*. The second group of theorems describes the relation between two *POS* statements with different temporal arguments and the properties of iterated *INEV* and *POS* statements with different temporal arguments. These theorems are derived using axiom *AX-INV5* along with a theorem we present in the first group.

Theorems Relating *INEV* with Interval Logic Statements

We present some general theorems that permit us to take certain interval logic statements out of the scope of the *INEV* modal operator. In section 4.4, we present related theorems that allow us take certain interval logic sentences out of the scope of the *IFTRIED* modal operator.

To describe the following theorems, we make use of the function *CI* which stands for "condition intervals". Function *CI* takes an interval logic sentence *ILS* as an argument and yields the set of interval terms, possibly empty, that appear in a *OCCURS* or *HOLDS* sentence in *ILS* and terms of the form $\lceil(\text{TIME-OF } \text{pi1})\rceil$ for each $\lceil(\text{OCC } \text{pi1})\rceil$ that appear in *ILS*. For example, $\text{CI}(\lceil(\text{OR } (\text{HOLDS } \text{pr } \text{i1}) (\text{OCC } \text{pi}))\rceil)$ is equal to $\{\text{i1}, \lceil(\text{TIME-OF } \text{pi})\rceil\}$.

Using *CI*, we can succinctly describe the following theorem:¹

TH-INV-IL1)

$$\{\lceil(\text{ENDS} \leq \text{ix } \text{i})\rceil \mid \text{ix} \in \text{CI}(\text{ILS})\} \vdash (\text{IF } (\text{POS } \text{i } \text{ILS}) (\text{INEV } \text{i } \text{ILS}))$$

where *ILS* is an interval logic sentence that does not contain any quantifiers (although *ILS* may contain free variables)

Examples of instances of *TH-INV-IL1* are:

$$\begin{aligned} &\{\lceil(\text{ENDS} \leq \text{i3 } \text{i})\rceil\} \\ &\vdash (\text{IF } (\text{POS } \text{i } (\text{IF } (\text{MEETS } \text{i1 } \text{i3}) (\text{HOLDS } \text{pr } \text{i3}))) \\ &\quad (\text{INEV } \text{i } (\text{IF } (\text{MEETS } \text{i1 } \text{i3}) (\text{HOLDS } \text{pr } \text{i3})))) \\ &\{\lceil(\text{ENDS} \leq \text{i1 } \text{i})\rceil, \lceil(\text{ENDS} \leq \text{i } (\text{TIME-OF } \text{pi2}))\rceil\} \\ &\vdash (\text{IF } (\text{POS } \text{i } (\text{AND } (\text{OCCURS } \text{ev1 } \text{i1}) (\text{OCC } \text{pi2}))) \\ &\quad (\text{INEV } \text{i } (\text{AND } (\text{OCCURS } \text{ev1 } \text{i1}) (\text{OCC } \text{pi2})))) \end{aligned}$$

¹ A more general theorem may be developed that allows quantifiers in *ILS*. The form of this theorem, however, would be more complex because the statements in $\{\lceil(\text{ENDS} \leq \text{ix } \text{i})\rceil \mid \text{ix} \in \text{CI}(\text{ILS})\}$ must be correctly scoped with respect to the quantifiers in *ILS*

Theorem *TH-INV-IL1* says that if all conditions mentioned in interval logic statement *ILS* occur or hold over intervals that end before or at the same time as *i*, then if *ILS* is possible at *i* then *ILS* is inevitable at *i*. The proof of theorem *TH-INV-IL1* is given in appendix F. Using *TH-INV-IL1*, axiom *AX-INV1*, which says whatever is inevitable is actual, and theorem *TH-INV-SF2*, which says whatever is actual is possible, we can derive the following theorem:

TH-INV-IL2)

$$\{ \ulcorner (ENDS \leq ix \ i) \urcorner \mid ix \in CI(ILS) \} \vdash (IFF \ ILS \ (INEV \ i \ ILS))$$

where *ILS* is an interval logic sentence that does not contain any quantifiers

Theorem *TH-INV-IL2* may be used to lift an interval statement, all of whose condition intervals end before or at the same time as *i*, out of the scope of *INEV* with argument *i*.

If we have an interval logic statement that only mentions relations between intervals, then we can use the following theorem, the proof of which is given in appendix F:

TH-INV-IL3)

$$\vdash (IF \ (POS \ i \ IRS) \ (INEV \ i \ IRS))$$

where *IRS* is an interval relation sentence, that is, one formed only by the first order connectives, including equality, and the *MEETS* predicate.

Unlike the two theorems above, *IRS* is not restricted so that it does not contain any quantifiers. Thus, theorem *TH-INV-IL3* is applicable for all interval relations (i.e. *IN*, *ENDS* \leq , *PRIOR*, ...) substituted for *IRS* since these relations are defined in terms of *MEETS* and the first order connectives. Theorem *TH-INV-IL3* is a special case of *TH-INV-IL1* when *IRS* does not contain any quantifiers. This is because for any statement *IRS* only containing *MEETS* and equality predicates, $CI(IRS)$ is empty, and consequently $\{ \ulcorner (ENDS \leq ix \ i) \urcorner \mid ix \in CI(IRS) \}$ is empty. Thus, " $\{ \ulcorner (ENDS \leq ix \ i) \urcorner \mid ix \in CI(IRS) \} \vdash (IF \ (POS \ i \ IRS) \ (INEV \ i \ IRS))$ " is equivalent to " $\vdash (IF \ (POS \ i \ IRS) \ (INEV \ i \ IRS))$ " since, by definition, " $\emptyset \vdash P$ " is equivalent to " $\vdash P$ " for any sentence *P*.

We can also derive a form analogous to *TH-INV2-IL3* for interval relation statements. This is given by:

TH-INV-IL4)

$$\vdash (IFF \ IRS(INEV \ i \ IRS))$$

where *IRS* is an interval relation sentence

POS and INEV with Different Temporal Arguments

The relation between time movement and possibility is the opposite of that with inevitability. As time goes on, the set of possible sentences shrinks. This is captured by the following theorem:

TH-INV-T1)

$$\vdash (\text{IF } (\text{ENDS} \leq i1\ i2) \\ (\text{IF } (\text{POS } i2\ P) (\text{POS } i1\ P)))$$

Theorem *TH-INV-T1* says that whatever is possible at interval *i2* is possible at any interval that ends at the same time or earlier than interval *i2*. *TH-INV-T1* may be derived by substituting $\lceil (\text{NOT } P) \rceil$ in *AX-INV5*, transposing the material implication, and then substituting the definition for *POS*. In the last section, we have characterized a relation such as the one between *TH-INV-T1* and *AX-INV5* as being symmetric statements. In the following development, if two relations are symmetrical, we only give proofs for one of the pairs, the other one being easily derived from its symmetric counterpart.

The nesting of two *INEV* statement with different temporal arguments can be described by:

TH-INV-T2)

$$\vdash (\text{IF } (\text{ENDS} \leq i1\ i2) \\ (\text{IFF } (\text{INEV } i1\ (\text{INEV } i2\ P)) \\ (\text{INEV } i1\ P)))$$

TH-INV-T3)

$$\vdash (\text{IF } (\text{ENDS} \leq i1\ i2) \\ (\text{IFF } (\text{INEV } i2\ (\text{INEV } i1\ P)) \\ (\text{INEV } i1\ P)))$$

The proofs of *TH-INV-T2* and *TH-INV-T3* are given in appendix F. Taking *TH-INV-T2* and *TH-INV-T3* together, we see that two nested *INEV* statements may be collapsed into a single operator with the earlier time (or either one, if they end at the same time) serving as the index. This relation may be generalized to a chain of three or more *INEV* operators. In this case, by successively using *TH-INV-T2* and *TH-INV-T3* we can derive that a chain of three or more *INEV* operators can be reduced to a single operator that has any interval argument *i* meeting the property: there are no interval arguments in the chain that end before *i*.

The relation between two nested *POS* statements with different intervals can be given by:

TH-INV-T4)

$$\vdash (\text{IF } (\text{ENDS} \leq i1 \ i2) \\ (\text{IFF } (\text{POS } i1 (\text{POS } i2 \ P)) \\ (\text{POS } i1 \ P)))$$

TH-INV-T5)

$$\vdash (\text{IF } (\text{ENDS} \leq i1 \ i2) \\ (\text{IFF } (\text{POS } i2 (\text{POS } i1 \ P)) \\ (\text{POS } i1 \ P)))$$

These two relations are symmetric to the relations for nested *INEV* statements. Theorem *TH-INV-T4* is symmetric to *TH-INV-T2*, and theorem *TH-INV-T5* is symmetric to *TH-INV-T3*. This leads to the same nesting relation for *POS* chains as for *INEV* chains. That is, two or more nested *POS* statements may be collapsed into a single operator that has any interval argument *i* meeting the property: there are no interval arguments in the chain that end before *i*.

There are two relations describing the collapsing of a *INEV* and a *POS* statement nested together:

TH-INV-T6)

$$\vdash (\text{IF } (\text{ENDS} \leq i1 \ i2) \\ (\text{IFF } (\text{POS } i2 (\text{INEV } i1 \ P)) \\ (\text{INEV } i1 \ P)))$$

TH-INV-T7)

$$\vdash (\text{IF } (\text{ENDS} \leq i1 \ i2) \\ (\text{IFF } (\text{INEV } i2 (\text{POS } i1 \ P)) \\ (\text{POS } i1 \ P)))$$

Theorems *TH-INV-T6* and *TH-INV-T7* are symmetric, thus we only prove one of them (see Appendix F).

There are two nested forms, $\lceil (\text{INEV } i1 (\text{POS } i2 \ P)) \rceil$ and $\lceil (\text{POS } i1 (\text{INEV } i2)) \rceil$, that cannot be reduced to a single operator. If one of these statements could be reduced then the other could be reduced since they are duals of each other with respect to *P*. That is, $\lceil (\text{INEV } i1 (\text{POS } i2 \ P)) \rceil$ is equivalent to $\lceil (\text{NOT } (\text{POS } i1 (\text{NOT } i2 (\text{NOT } P)))) \rceil$ and $\lceil (\text{POS } i1 (\text{INEV } i2)) \rceil$ is equivalent to $\lceil (\text{NOT } (\text{INEV } i1 (\text{POS } i2 (\text{NOT } P)))) \rceil$. Statements of the form $\lceil (\text{INEV } i1 (\text{POS } i2 \ P)) \rceil$ are useful for describing situations where some occurrence is possible no matter what transpires between the end of *i1* up until the end of *i2*.

4.4. The Axiomatization of IFTRIED

We break the axiomatization of *IFTRIED* into three different groups. The first group captures the properties associated with a subjunctive conditional. The second group captures the relation between *IFTRIED* and *INEV*, and the third group describes the relation between the attempt of a composite plan instance and the attempt of the individual plan instances constituting the composition.

4.4.1. IFTRIED as a Subjunctive Conditional

The axioms capturing the properties of *IFTRIED* that are associated with a subjunctive conditional are given in figure 4.4-1. Axiom *AX-IFTR1* captures that

<u>Axioms</u>	
AX-IFTR1)	
⊢ (IF (IFTRIED pi P)	
(NOT (IFTRIED pi (NOT P))))	
AX-IFTR2)	
⊢ (IF (IFTRIED pi (IF P Q))	
(IF (IFTRIED pi P) (IFTRIED i Q)))	
AX-IFTR3)	
⊢ (IF (OCC pi)	
(IFF (IFTRIED pi P) P))	
AX-IFTR4)	
⊢ (IFF (IFTRIED pi (IFTRIED pi P))	
(IFTRIED pi P))	
<u>Inference Rule</u>	
RL-IFTR)	
From: ⊢ P	
To: ⊢ (IFTRIED pi P)	

Figure 4.4-1

$\lceil \text{IFTRIED pi P} \rceil$ and $\lceil \text{IFTRIED pi (NOT P)} \rceil$ cannot both hold together. This contrasts with the material conditional where $\lceil \text{IF P Q} \rceil$ and $\lceil \text{IF P (NOT Q)} \rceil$ are both (vacuously) true when *P* is false. Axiom *AX-IFTR2* says that material implication distributes out of the *IFTRIED* modality.

Axiom *AX-IFTR3* corresponds to a property found in both Stalnaker's [Stalnaker 68] and Lewis' [Lewis 73] theories of conditionals which says that if antecedent *A* holds, then the conditional "if *A* then *C*" is true iff *C* is true. Remember that $\lceil \text{IFTRIED pi P} \rceil$ can be interpreted as saying "if *pi* were to be attempted then *P*

would be true". Now, in our language, we do not have an atomic formula that corresponds to "*pi* is attempted". We do, however, have a formula corresponding to *pi* occurs (i.e. $\lceil(\text{OCC } pi)\rceil$) which is stronger than "*pi* is attempted" (see section 3.2.3). Thus in our system, if $\lceil(\text{OCC } pi)\rceil$ is true, $\lceil(\text{IFTRIED } pi \ P)\rceil$ is true iff *P* is true.

Axiom *AX-IFTR4* reflects that two nested *IFTRIED* operators having the same argument can be collapsed to a single operator. Generalizing, we can use *AX-IFTR4* to prove that two or more nested *IFTRIED* operators with the same plan instance argument can be collapsed to a single operator. In section 4.4.3, we present axioms describing nested *IFTRIED* operators with different plan instance arguments, these being the more interesting cases.

Inference rule *RL-IFTR* can be characterized as a rule of necessitation. This rule says: if *P* is a theorem then $\lceil(\text{IFTRIED } pi \ P)\rceil$ is a theorem for any plan instance term *pi*. Just like the rule of necessitation for *INEV*, *RL-IFTR* is given in *theoremhood form* rather than in *derivability form*, i.e. from " $S \vdash P$ " to " $S \vdash (\text{IFTRIED } i \ P)$ ". This more general form would lead to an unsound system.

Soundness Proofs

The proofs that axioms *AX-IFTR1* - *AX-IIFTR4* are valid are given in figures 4.4-2 and 4.4-3. These proofs are established using the interpretation of *IFTRIED*, the constraints *BA0*, *BA1*, and *BA2* (which are placed on the basic action functions), and the F_{cl} function, which is defined in terms of the basic action functions. The proof that *RL-IFTR* preserves validity is established in a similar fashion as was done to show that *RL-INV* preserves validity (see section 4.3). The statement $\lceil(\text{IFTRIED } pi \ P)\rceil$ is interpreted as true at world-history *h* iff *P* is true in all the closest world-histories to *h* where the plan instance denoted by *pi* is attempted. If *P* is valid, it is true at all world-histories in all models, this includes the closest world-histories to *h* where the plan instance denoted by *pi* is attempted.

Soundness Proofs

The interpretation of IFTRIED, which is used in all the proofs in this figure and 4.4-3, and can be given by:

$V_s(\ulcorner \text{IFTRIED } \pi \text{ } P \urcorner, h) = \text{TRUE}$ iff
for all world-histories (h_2) if $h_2 \in F_{cl}(V_t(\pi)|_2, h)$ then $V_s(P, h_2) = \text{TRUE}$

AX-IFTR1)

$\vdash (\text{IF } (\text{IFTRIED } \pi \text{ } P) (\text{NOT } (\text{IFTRIED } \pi \text{ } (\text{NOT } P))))$

For an arbitrary V_s and world-history h , assume that $V_s(\ulcorner \text{IFTRIED } \pi \text{ } P \urcorner, h)$ equals TRUE. We validate AX-IFTR1 by showing that $V_s(\ulcorner (\text{NOT } (\text{IFTRIED } \pi \text{ } (\text{NOT } P))) \urcorner, h)$ equals TRUE under our assumption. This is established using constraint BA0 and the definition of F_{cl} which can be used to derive:

1) $F_{cl}(V_t(\pi)|_2, h) \neq \emptyset$

From $V_s(\ulcorner \text{IFTRIED } \pi \text{ } P \urcorner, h)$ equals TRUE, we get:

2) For all world-histories (h_2) if $h_2 \in F_{cl}(V_t(\pi)|_2, h)$ then $V_s(P, h_2) = \text{TRUE}$

Taking 1 and 2 together yields the following which is true iff $V_s(\ulcorner (\text{NOT } (\text{IFTRIED } \pi \text{ } (\text{NOT } P))) \urcorner, h)$ equals TRUE:

It is not true that for all world-history (h_3) if $h_3 \in F_{cl}(V_t(\pi)|_2, h)$ then $V_s(P, h_3) = \text{FALSE}$

AX-IFTR2)

$\vdash (\text{IF } (\text{IFTRIED } \pi \text{ } (\text{IF } P \text{ } Q))$
 $\quad (\text{IF } (\text{IFTRIED } \pi \text{ } P) (\text{IFTRIED } \pi \text{ } Q)))$

For an arbitrary V_s and world-history h , assume that both $V_s(\ulcorner \text{IFTRIED } \pi \text{ } (\text{IF } P \text{ } Q) \urcorner, h)$ and $V_s(\ulcorner \text{IFTRIED } \pi \text{ } P \urcorner, h)$ equal TRUE. We validate AX-IFTR2 by showing that $V_s(\ulcorner \text{IFTRIED } \pi \text{ } Q \urcorner, h)$ equals TRUE under our assumptions. Using $V_s(\ulcorner \text{IFTRIED } \pi \text{ } (\text{IF } P \text{ } Q) \urcorner, h)$ equals TRUE, we get:

1) For all world-histories (h_2) if $h_2 \in F_{cl}(V_t(\pi)|_2, h)$ then $V_s(\ulcorner \text{IF } P \text{ } Q \urcorner, h_2) = \text{TRUE}$

Using $V_s(\ulcorner \text{IFTRIED } \pi \text{ } P \urcorner, h)$ equals TRUE, we get:

2) For all world-histories (h_2) if $h_2 \in F_{cl}(V_t(\pi)|_2, h)$ then $V_s(P, h_2) = \text{TRUE}$

Taking 1 and 2 together (and using the interpretation of IF), we can derive the following which is true iff $V_s(\ulcorner \text{IFTRIED } \pi \text{ } Q \urcorner, h)$ equals TRUE:

For all world-histories (h_2) if $h_2 \in F_{cl}(V_t(\pi)|_2, h)$ then $V_s(Q, h_2) = \text{TRUE}$

Figure 4.4-2

Soundness Proofs

AX-IFTR3)

$$\vdash (\text{IF}(\text{OCC } \pi) \rightarrow (\text{IFF}(\text{IFTRIED } \pi \mid P) \mid P))$$

For an arbitrary V_s and world-history h , assume that $V_s(\neg(\text{OCC } \pi) \mid h)$ equals TRUE which gives us:

- 1) For all events (ev), intervals (i) and basic actions (ba)
 - if $\langle \text{ev}, i \rangle \in V_t(\pi) \mid_1$, then $\langle i, h \rangle \in \text{ev}$, and
 - if $\langle \text{ba}, i \rangle \in V_t(\pi) \mid_2$, then $\langle i, h \rangle \in \text{BAEV}(\text{ba})$

Using constraint BA1 and the definition of F_{cl} we can derive:

- 2) If for all intervals (i) and basic actions (ba)
 - if $\langle \text{ba}, i \rangle \in V_t(\pi) \mid_2$ then $\langle i, h \rangle \in \text{BAEV}(\text{ba})$
 - then $F_{cl}(V_t(\pi) \mid_2, h) = \{h\}$

Taking the first conjunct in 1 and 2 together yields " $F_{cl}(V_t(\pi) \mid_2, h) = \{h\}$ " which implies that $V_s(\neg(\text{IFTRIED } \pi \mid P) \mid h)$ equals TRUE iff $V_s(P, h)$ equals TRUE

AX-IFTR4)

$$\vdash (\text{IFF}(\text{IFTRIED } \pi \mid (\text{IFTRIED } \pi \mid P)) \rightarrow (\text{IFTRIED } \pi \mid P))$$

Let V_s be an arbitrary interpretation function and let h be an arbitrary world-history. Using constraint BA2 and the definition of F_{cl} we can derive:

- 1) For all world-histories (h_2),
 - if $h \neq h_2$ and $h_2 \in F_{cl}(V_t(\pi) \mid_2, h)$ then for all intervals (i) and basic actions (ba)
 - if $\langle \text{ba}, i \rangle \in V_t(\pi) \mid_2$ then $\langle i, h_2 \rangle \in \text{BAEV}(\text{ba})$

Taking 1 and step 2 from the proof of AX-IFTR3 above, we get:

- 2) For all world-histories (h_2),
 - if $h \neq h_2$ and $h_2 \in F_{cl}(V_t(\pi) \mid_2, h)$ then $F_{cl}(V_t(\pi) \mid_2, h_2) = \{h_2\}$

From 2 we can derive the following which implies that $V_s(\neg(\text{IFTRIED } \pi \mid (\text{IFTRIED } \pi \mid P)) \mid h)$ equals TRUE iff $V_s(\neg(\text{IFTRIED } \pi \mid P) \mid h)$ equals TRUE:

$$hx \in F_{cl}(V_t(\pi) \mid_2, h) \text{ iff } hx \in F_{cl}(V_t(\pi) \mid_2, h_2) \text{ for some } h_2 \text{ s.t. } h_2 \in F_{cl}(V_t(\pi) \mid_2, h)$$

Figure 4.4-3

Theorems

In section 4.3.1., we discussed deduction in a modal context and presented a general rule allowing us to infer from "P entails Q" to "MC(P) entails MC(Q)" where *MC* is a modal chain consisting of *INEV* and *POS* statements. In this section, we give a more general rule covering modal chains that also contain *IFTRIED* and *P-IFTRIED* statements (*P-IFTRIED* is the dual of *IFTRIED* for a fixed plan instance argument). This generalized rule is given by *DRL-IFTR1*:

DRL-IFTR1)

From: $\vdash (IF\ P\ Q)$

To: $\vdash (IF\ IMC(P)\ IMC(Q))$

where *IMC(P)* is a modal chain formed by *INEV*, *POS*, *IFTRIED*, and *P-IFTRIED* operators with *P* embedded on the inside. Any modal chain *IMC(P)* must have one of the following forms:

- | | |
|---|--|
| i) $\lceil (INEV\ i\ P) \rceil$ | v) $\lceil (IFTRIED\ pi\ P) \rceil$ |
| ii) $\lceil (POS\ i\ P) \rceil$ | vi) $\lceil (P-IFTRIED\ pi\ P) \rceil$ |
| iii) $\lceil (INEV\ i\ IMC2(P)) \rceil$ | vii) $\lceil (IFTRIED\ pi\ IMC2(P)) \rceil$ |
| iv) $\lceil (POS\ i\ IMC2(P)) \rceil$ | viii) $\lceil (P-IFTRIED\ pi\ IMC2(P)) \rceil$ |

where *IMC2(P)* is any modal chain formed by *INEV*, *POS*, *IFTRIED*, and *P-IFTRIED* operators with *P* embedded on the inside

We can also generalize the rules *DRL-INV2* and *DRL-INV3* given in section 4.3.1. The following two rules permit substitutions within an *IFTRIED* modal operator:

DRL-IFTR2)

From: $\vdash (IFF\ P\ Q)$

To: $\vdash (IFF\ R1\ R2)$

where *R1* and *R2* are any sentences, and *R1* differs from *R2* by replacing one or more occurrences of *P* with *Q*

DRL-IFTR3)

From: $\vdash (IFF\ P\ Q)$ and $S \vdash R1$

To: $S \vdash R2$

where *S* is a set of sentences, *R1* and *R2* are any sentences, and *R1* differs from *R2* by replacing one or more occurrences of *P* with *Q*

Two other rules that will be used in later proofs are given by:

DRL-IFTR4)

From: $S1 \vdash NP-MC(P)$ and $S2 \vdash NP-MC(\ulcorner (IF\ P\ Q) \urcorner)$

To: $S1 \cup S2 \vdash NP-MC(Q)$

where $NP-MC$ is a modal chain consisting only of *IFTRIED* and *INEV* operators, $S1$ and $S2$ are sets of sentences, and P and Q are sentences

DRL-IFTR5)

From: $S1 \vdash NP-MC(P)$ and $S2 \vdash NP-MC(Q)$

To: $S1 \cup S2 \vdash NP-MC(\ulcorner (AND\ P\ Q) \urcorner)$

where $NP-MC$ is a modal chain consisting only of *IFTRIED* and *INEV* operators, $S1$ and $S2$ are sets of sentences, and P and Q are sentences

The relation between *IFTRIED* and the logical connectives parallels the relation between *INEV* and the logical connectives (see section 4.3.1). These relations are given by the following theorems:

TH-IFTR-SC1)

$\vdash (IFF\ (AND\ (IFTRIED\ \pi\ P)\ (IFTRIED\ \pi\ Q))\ (IFTRIED\ \pi\ (AND\ P\ Q)))$

TH-IFTR-SC2)

$\vdash (IF\ (OR\ (IFTRIED\ \pi\ P)\ (IFTRIED\ \pi\ Q))\ (IFTRIED\ \pi\ (OR\ P\ Q)))$

TH-IFTR-SC3)

$\vdash (IF\ (IFTRIED\ \pi\ (NOT\ P))\ (NOT\ (IFTRIED\ \pi\ P)))$

TH-IFTR-SC4)

$\vdash (IFF\ (\forall?v\ (IFTRIED\ \pi\ P))\ (IFTRIED\ \pi\ (\forall?v\ P)))$

where π has no free occurrences of $?v$

TH-IFTR-SC5)

$\vdash (IF\ (\exists?v\ (IFTRIED\ \pi\ P))\ (IFTRIED\ \pi\ (\exists?v\ P)))$

where π has no free occurrences of $?v$

4.4.2. The Relation Between IFTRIED and INEV

The relation between *IFTRIED* and *INEV* are given by the two axioms in figure 4.4-4. Axiom *AX-IFTR5* can be interpreted as saying that if a proposition *P* is inevitably true at time *i*, then the attempt of a plan instance that starts later than *i* cannot negate *P*. This axiom is used to derive that the attempt of a plan instance cannot affect earlier conditions. Axiom *AX-IFTR6* says that if the attempt of plan instance *pi* would result in *P* being inevitably true at time *i*, then it is inevitable at time *i* that the attempt of *pi* would result in *P* being true. An instantiation of this axiom, where *pi* ends before interval *i*, reflects the fact that the attempt of plan instance *pi* cannot be influenced by possible conditions that happen later than *pi*'s time of occurrence.

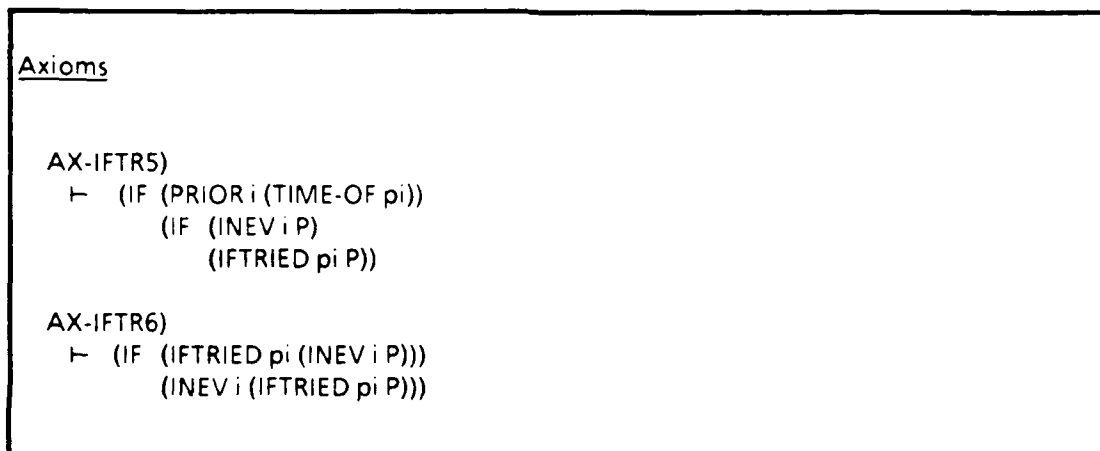


Figure 4.4-4

Soundness Proofs

The proofs that axioms *AX-IFTR5* and *AX-IFTR6* are valid are given in figures 4.4-5 and 4.4-6.

Soundness Proofs

AX-IFTR5)

$\vdash (IF \text{ (PRIOR } i \text{ (TIME-OF } pi))$
 $(IF \text{ (INEV } i \text{ P) (IFTRIED } pi \text{ P))})$

For an arbitrary V_s and world-history h , assume that both $V_s(\ulcorner(PRIOR \ i \text{ (TIME-OF } pi))\urcorner, h)$ and $V_s(\ulcorner(INEV \ i \text{ P})\urcorner, h)$ equal TRUE. We validate AX-IFTR5 by showing that $V_s(\ulcorner(IFTRIED \ pi \text{ P})\urcorner, h)$ equals TRUE under these assumptions. This is established by making use of constraint BA-R1 which entails the following:

- 1) For all world-histories ($h2$), basic actions (ba), and intervals (ix),
 if $h2 \in ba(ix, h)$ and $MTS(V_t(i), ix)$ then $R(V_t(i), h, h2)$

From 1, R1, and interval relation "if $V_t(i)$ is prior to ix , then there exists an interval iy that ends at the same time as ix and is met by $V_t(i)$ ", we can derive:

- 2) For all world-histories ($h2$), basic actions (ba), and intervals (ix),
 if $h2 \in ba(ix, h)$ and $PRIOR(V_t(i), ix)$ then $R(V_t(i), h, h2)$

From 2, the definition of F_{cl} and R3 (R is transitive) we can derive:

- 3) For all world-histories ($h2$)
 if $h2 \in F_{cl}(V_t(pi)|_2, h)$ and for all basic actions (ba) and intervals (ix)
 if $\langle ba, ix \rangle \in V_t(pi)|_2$ then $PRIOR(V_t(i), ix)$
 then $R(V_t(i), h, h2)$

From $V_s(\ulcorner(PRIOR \ i \text{ (TIME-OF } pi))\urcorner, h)$ equals TRUE, we can derive:

- 4) for all basic actions (ba) and intervals (ix),
 if $\langle ba, ix \rangle \in V_t(pi)|_2$ then $PRIOR(V_t(i), ix)$

Taking 3 and 4 together yields:

- 5) For all world-histories ($h2$) if $h2 \in F_{cl}(V_t(pi)|_2, h)$ then $R(V_t(i), h, h2)$

From $V_s(\ulcorner(INEV \ i \text{ P})\urcorner, h)$ equals TRUE we get:

- 6) For all world-histories ($h2$) if $R(V_t(i), h, h2)$ then $V_s(P, h2) = \text{TRUE}$

Combining 5 and 6 gives us the following which is true iff $V_s(\ulcorner(IFTRIED \ pi \text{ P})\urcorner, h)$ equals TRUE:

For all world-histories ($h2$) if $h2 \in F_{cl}(V_t(pi)|_2, h)$ then $V_s(P, h2) = \text{TRUE}$

Figure 4.4-5

Soundness Proofs

AX-IFTR6)

$$\vdash (IF (IFTRIED \pi (INEV \mid P)) \\ (INEV \mid (IFTRIED \pi P)))$$

For an arbitrary V_s and world-history h , assume that $V_s(\ulcorner (IFTRIED \pi (INEV \mid P)) \urcorner, h)$ equals TRUE. We show that $V_s(\ulcorner (INEV \mid (IFTRIED \pi P)) \urcorner, h)$ equals TRUE follows from our assumption. From $V_s(\ulcorner (IFTRIED \pi (INEV \mid P)) \urcorner, h)$ equals TRUE we have:

- 1) For all world-histories (h_2) if $h_2 \in F_{cl}(V_t(\pi) \mid_2, h)$ then for all world-histories (h_3) if $R(V_t(i), h_2, h_3)$ then $V_s(P, h_3)$

Rewriting 1 in an equivalent form:

- 2) If there exists a world-history (h_2) such that $h_2 \in F_{cl}(V_t(\pi) \mid_2, h)$ then for all world-histories (h_3) if $R(V_t(i), h_2, h_3)$ then $V_s(P, h_3)$

From BA-R2, the definition of F_{cl} and R2 and R3 (R is an equivalence relation for a fixed time argument), we can derive:

- 3) For all world-histories (h_1), if $R(V_t(i), h, h_1)$ then for all world-histories (h_3) if $h_3 \in F_{cl}(V_t(\pi) \mid_2, h_1)$ then there exists a world-history (h_2) such that $h_2 \in F_{cl}(V_t(\pi) \mid_2, h)$ and $R(V_t(i), h_2, h_3)$

From 2 and 3, we can derive: the following which is true iff $V_s(\ulcorner (INEV \mid (IFTRIED \pi P)) \urcorner, h)$ equals TRUE:

For all world-histories (h_1), if $R(V_t(i), h, h_1)$ then for all world-histories (h_3) if $h_3 \in F_{cl}(V_t(\pi) \mid_2, h_1)$ then $V_s(P, h_3)$

Figure 4.4-6

Theorems

The following three theorems, which are derived from axiom *AX-IFTR5*, are useful for bringing interval logic statements in and out of the scope of *IFTRIED* operators. These theorems are similar to theorems *TH-INV-IL1 - TH-INV-IL4* (see the end of section 4.3) which are used to bring interval logic statements in and out of the scope of *INEV* operators. The three theorems below capture that an interval logic statement *ILS* can be brought in and out of the scope of an *IFTRIED* operator with index *pi* if all the conditions in *ILS* refer to times that are prior to *pi*'s time of occurrence.

To describe the following axioms we make use of the *CI* function which we described in section 4.3. Reiterating, function *CI* takes an interval logic sentence *ILS* as an argument and yields the set of interval terms, possibly empty, that appear in a *OCCURS* or *HOLDS* sentence in *ILS* and terms of the form $\lceil(\text{TIME-OF } pi1)\rceil$ for each $\lceil(\text{OCC } pi1)\rceil$ that appear in *ILS*. For example, $CI(\lceil(\text{IF } (\text{MEETS } i1 \ i2) \text{ (OR } (\text{HOLDS } pr \ i1) \text{ (OCC } pi))\rceil)\rceil$ is equal to $\{i1, \lceil(\text{TIME-OF } pi)\rceil\}$.

Theorem *TH-IFTR-IN1* below says that if all conditions mentioned in interval logic statement *ILS* occur or hold over intervals that are prior to *pi*'s time of occurrence *i*, then if *ILS* is true then if *pi* would be attempted then *ILS* would (still) be true.

TH-IFTR-IN1)

$$\{\lceil(\text{PRIOR } ix \text{ (TIME-OF } pi)\rceil \mid ix \in CI(ILS)\} \vdash (\text{IF } ILS \text{ (IFTRIED } pi \ ILS))$$

where *ILS* is a sentence in the interval logic fragment containing no quantifiers (although it might have free variables)

$$(\text{PRIOR } i1 \ i2) =_{\text{def}} (\text{OR } (\text{MEETS } i1 \ i2) \text{ (}\exists ?i \text{ (AND } (\text{MEETS } i1 \ ?i) \text{ (MEETS } ?i \ i2))))$$

Some instances of *TH-IFTR-IN1* are:

$$\{\lceil(\text{PRIOR } i \text{ (TIME-OF } pi)\rceil\} \vdash (\text{IF } (\text{HOLDS } pr \ i) \text{ (IFTRIED } pi \text{ (HOLDS } pr \ i)))$$

$$\{\lceil(\text{PRIOR } i2 \text{ (TIME-OF } pi)\rceil\} \vdash (\text{IF } (\text{IF } (\text{MEETS } i \ ?i2) \text{ (OCCURS } ev \ i2)) \text{ (IFTRIED } pi \text{ (IF } (\text{MEETS } i \ i2) \text{ (OCCURS } ev \ i2))))$$

$$\{\lceil(\text{PRIOR } i \text{ (TIME-OF } pi2)\rceil, \lceil(\text{PRIOR } (\text{TIME-OF } pi1) \text{ (TIME-OF } pi2)\rceil\} \vdash (\text{IF } (\text{OR } (\text{HOLDS } pr \ i) \text{ (OCC } pi1)) \text{ (IFTRIED } pi2 \text{ (OR } (\text{HOLDS } pr \ i) \text{ (OCC } pi1))))$$

The proof of theorem *TH-IFTR-IN1* is given in appendix F. Using *TH-IFTR-IN1*, axiom *AX-INV1*, which says whatever is inevitable is actual, we can derive the following theorem:

TH-IFTR-IN2)

$$\{ \ulcorner (\text{PRIOR } ix (\text{TIME-OF } pi)) \urcorner \mid ix \in \text{CI}(\text{ILS}) \} \\ \vdash (\text{IFF ILS} (\text{IFTRIED } pi \text{ ILS}))$$

where *ILS* is a sentence in the interval logic fragment containing no quantifiers

TH-IFTR-IN2 may be used to lift an interval statement *ILS* out of the scope of an *IFTRIED* operator with argument *pi* if the conditions in *ILS* refer to intervals that are prior to *pi*'s time of occurrence.

The following theorem is applicable to interval relation statements, that is, statements formed by the *MEETS* atomic formula and the first order connectives (including the equality relation):

TH-IFTR-IN3)

$$\vdash (\text{IFF IRS} (\text{IFTRIED } pi \text{ IRS}))$$

where *IRS* is an interval relation sentence

Unlike the two theorems above, *IRS* may contain quantifiers. For the case where *IRS* does not contain quantifiers, *TH-IFTR-IN3* is a special case of *TH-IFTR-IN2*.

The following two theorems are derived using axiom *AX-IFTR6*, the proof of which is given in appendix F:

TH-IFTR-IN4)

$$\vdash (\text{IF} (\text{ENDS } i (\text{TIME-OF } pi) i)) \\ (\text{IF} (\text{POS } i (\text{EXECUTABLE } pi)) \\ (\text{INEV } i (\text{EXECUTABLE } pi))))$$

TH-IFTR-IN5)

$$\vdash (\text{IF} (\text{PRIOR} (\text{TIME-OF } pi1) (\text{TIME-OF } pi2)) \\ (\text{IFF} (\text{EXECUTABLE } pi1) \\ (\text{IFTRIED } pi2 (\text{EXECUTABLE } pi1))))$$

Theorem *TH-IFTR-IN4* says that if a plan instance *pi* ends before or at the same as interval *i* then if it is possible at *i* that *pi* is executable, then it is inevitable at *i* that *pi* is inevitable. *TH-IFTR-IN4* can also be interpreted as saying that if a plan instance *pi* ends before or at the same as interval *i* then it is inevitably true or inevitably false at *i* that *pi* is executable.

Theorem *TH-IFTR-IN5* reflects the fact that the attempt of a plan instance cannot affect whether or not an earlier plan instance is executable. It is derived using *TH-IFTR-IN4*, substituting *pi1* for *pi* and $\ulcorner (\text{TIME-OF } pi2) \urcorner$ for *i*, where *pi1* is prior to *pi2*, and then using axiom *AX-IFTR5*.

4.4.3. The Relation Between IFTRIED and Composition

The three axioms in figure 4.4-7 describe relations between the attempt of a composite plan instance and the individual attempts of the plan instances making up the composition. In this section, we will only briefly describe these axioms. For a better understanding of these axioms, the reader may wish to re-examine section 3.2.5 where the constraints in the model from which these axioms are validated are described in detail. In a later section (5.3), we will discuss the relation between composite plan instances and their component parts in some detail. Theorems that are derived from the axioms in figure 4.4-7 will be presented in this later section, instead of here.

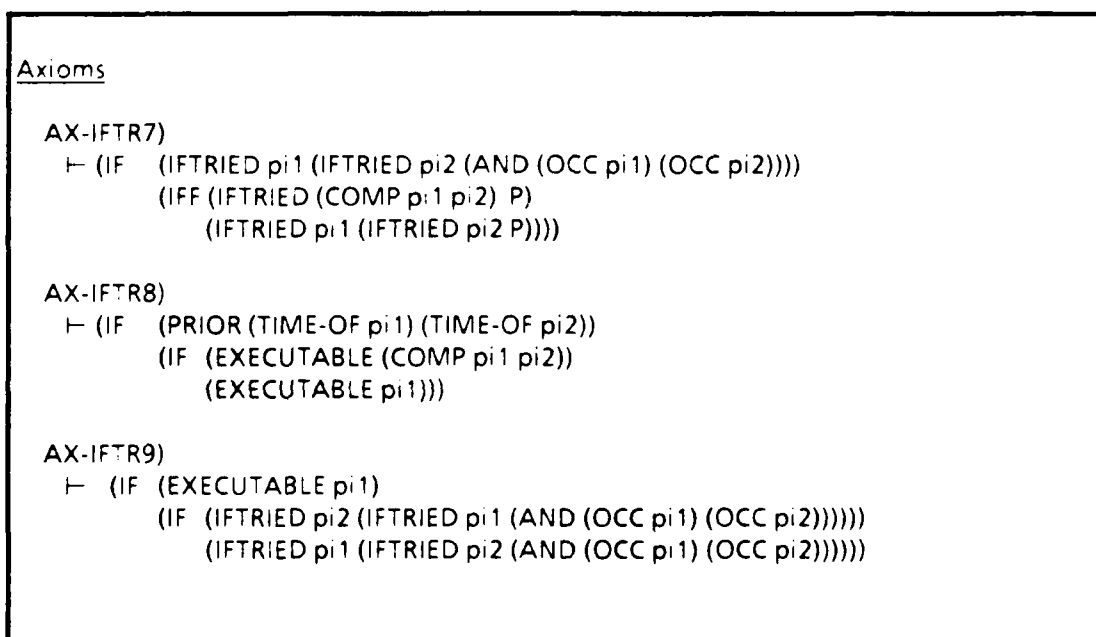


Figure 4.4-7

To justify axiom *AX-IFTR7* we use an explanation that parallels the one we gave in section 3.2.5 where we described the composition of basic action instances. *AX-IFTR7*'s antecedent is true iff both $pi1$ and $pi2$ would occur if $pi2$ were to be attempted in the scenario that would result from $pi1$ being attempted (note: This does not imply that $pi2$'s time of occurrence must be after $pi1$'s time of occurrence; the two plan instances may have any temporal relation). We take the truth of this antecedent to imply that $pi1$ and $pi2$ do not interfere, and consequently they could be executed together. Moreover, if *AX-IFTR7*'s antecedent is true then the statements that would be true if the composition of $pi1$ and $pi2$ (i.e., $\uparrow(\text{COMP } pi1 \ pi2)^{\uparrow}$) were to be attempted are exactly the statements that would be true if $pi2$ were to be attempted in the scenario that would result from $pi1$ being attempted.

Axiom *AX-IFTR8* captures that if the composition of $pi1$ and $pi2$ is executable, where $pi1$ is properly before $pi2$, then $pi1$ must be executable by itself. Axiom *AX-*

IFTR9 reflects the fact that the only way that $\lceil (\text{IFTRIED } \text{pix} (\text{IFTRIED } \text{piy} (\text{AND} (\text{OCC } \text{pix}) (\text{OCC } \text{piy})))) \rceil$ and $\lceil (\text{IFTRIED } \text{piy} (\text{IFTRIED } \text{pix} (\text{AND} (\text{OCC } \text{pix}) (\text{OCC } \text{piy})))) \rceil$ can have different truth values is if either *pix* or *piy* is not executable. Using this axiom along with *AX-IFTR7*, we can derive that if *pix*, *piy* and their composition are all executable, then $\lceil (\text{IFTRIED } \text{pix} (\text{IFTRIED } \text{piy } P)) \rceil$ is true iff $\lceil (\text{IFTRIED } \text{piy} (\text{IFTRIED } \text{pix } P)) \rceil$ is true for all sentences *P*.

Soundness proofs

The proofs that axioms *AX-IFTR7* - *AX-IFTR9* are valid are given in figures 4.4-8 - 4.4-10.

Soundness Proofs

AX-IFTR7)

$\vdash (IF \text{ (IFTRIED } \pi_1 \text{ (IFTRIED } \pi_2 \text{ (AND (OCC } \pi_1) \text{ (OCC } \pi_2))))$
 $\text{ (IFF (IFTRIED (COMP } \pi_1 \pi_2) P) \text{ (IFTRIED } \pi_1 \text{ (IFTRIED } \pi_2 P))))$

For an arbitrary V_s and world-history h , assume that $V_s(\ulcorner \text{IFTRIED } \pi_1 \text{ (IFTRIED } \pi_2 \text{ (AND (OCC } \pi_1) \text{ (OCC } \pi_2))} \urcorner, h)$ equals TRUE. We show that $V_s(\ulcorner \text{IFTRIED (COMP } \pi_1 \pi_2) P \urcorner, h)$ equals $V_s(\ulcorner \text{IFTRIED } \pi_1 \text{ (IFTRIED } \pi_2 P) \urcorner, h)$ follows from this assumption.

From our assumption we can derive:

- 1) For all world-histories (h_2 and h_3) if $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_2)|_2, h_2)$ then $V_s(\ulcorner \text{OCC } \pi_1 \urcorner, h_3)$ and $V_s(\ulcorner \text{OCC } \pi_2 \urcorner, h_3)$ equal TRUE

From 1 and the interpretation of OCC we can derive:

- 2) For all world-histories (h_2 and h_3) if $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_2)|_2, h_2)$ then for all basic actions (ba) and intervals (i) if $\langle ba, i \rangle \in V_t(\pi_1)|_2 \cup V_t(\pi_2)|_2$ then $\langle i, h_3 \rangle \in BAEV(ba)$

From 2 and definition of F_{cl} , we can derive:

- 3) For all world-histories (h_3) $h_3 \in F_{cl}(V_t(\pi_1)|_2 \cup V_t(\pi_2)|_2, h)$ iff there exists a world-history (h_2) such that $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_2)|_2, h_2)$

From 3 and the equality " $V_t(\ulcorner \text{COMP } \pi_1 \pi_2 \urcorner)_2 = V_t(\pi_1)|_2 \cup V_t(\pi_2)|_2$ ", we can derive:

- 4) For all world-histories (h_3) $h_3 \in F_{cl}(\ulcorner \text{COMP } \pi_1 \pi_2 \urcorner)_2, h)$ iff there exists a world-history (h_2) such that $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_2)|_2, h_2)$

From 4, we can derive the following which is true iff $V_s(\ulcorner \text{IFTRIED (COMP } \pi_1 \pi_2) P \urcorner, h)$ equals $V_s(\ulcorner \text{IFTRIED } \pi_1 \text{ (IFTRIED } \pi_2 P) \urcorner, h)$:

- 5) [For all world-histories (h_3) if $h_3 \in F_{cl}(\ulcorner \text{COMP } \pi_1 \pi_2 \urcorner)_2, h)$ then $V_s(P, h_3)$ equals TRUE] iff [For all world-histories (h_2 and h_3) if $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_2)|_2, h_2)$ then $V_s(P, h_3)$ equals TRUE]

Figure 4.4-8

Soundness Proofs

AX-IFTR8)

$\vdash (\text{IF } (\text{PRIOR } (\text{TIME-OF } \pi_1) (\text{TIME-OF } \pi_2))$
 $(\text{IF } (\text{EXECUTABLE } (\text{COMP } \pi_1 \pi_2)) (\text{EXECUTABLE } \pi_1)))$

For an arbitrary V_s and world-history h , assume that $V_s(\ulcorner (\text{PRIOR } (\text{TIME-OF } \pi_1) (\text{TIME-OF } \pi_2)) \urcorner, h)$ and $V_s(\ulcorner (\text{EXECUTABLE } (\text{COMP } \pi_1 \pi_2)) \urcorner, h)$ equal TRUE. We show that $V_s(\ulcorner (\text{EXECUTABLE } \pi_1) \urcorner, h)$ equals TRUE follows from these assumptions. For convenience, we will use PRIOR-PI2(i) to mean that interval i is prior to all the basic action instances contained in plan instance π_2 :

$\text{PRIOR-PI2}(i) =_{\text{def}} \text{ for all basic actions } (ba_2) \text{ and intervals } (i_2) \text{ if } \langle ba_2, i_2 \rangle \in V_t(\pi_2)|_2 \text{ then } \text{PRIOR}(i, i_2)$

From $V_s(\ulcorner (\text{PRIOR } (\text{TIME-OF } \pi_1) (\text{TIME-OF } \pi_2)) \urcorner, h)$ equals TRUE and constraint P11, we can derive:

- 1) For all events (ev_1) , basic actions (ba_1) and intervals (i_1) if $\langle ev_1, i_1 \rangle \in V_t(\pi_1)|_1$ or $\langle ba_1, i_1 \rangle \in V_t(\pi_1)|_2$ then $\text{PRIOR-PI2}(i_1)$

From $V_s(\ulcorner (\text{EXECUTABLE } (\text{COMP } \pi_1 \pi_2)) \urcorner, h)$ equals TRUE and the definition of EXECUTABLE, " $(\text{EXECUTABLE } \pi_i) =_{\text{def}} (\text{IFTRIED } \pi_i (\text{OCC } \pi_i))$ ", we can derive:

- 2) For all world-histories (h_2) if $h_2 \in F_{cl}(V_t(\ulcorner (\text{COMP } \pi_1 \pi_2) \urcorner)|_2, h)$ then $V_s(\ulcorner (\text{OCC } (\text{COMP } \pi_1 \pi_2)) \urcorner, h_2)$ equals TRUE

From 3, the interpretation of OCC and the equality " $V_t(\ulcorner (\text{COMP } \pi_1 \pi_2) \urcorner)|_2 = V_t(\pi_1)|_2 \cup V_t(\pi_2)|_2$ ", we can derive:

- 3) For all world-histories (h_2) if $h_2 \in F_{cl}((V_t(\pi_1)|_2 \cup V_t(\pi_2)|_2), h)$ then for all basic actions (ba) and intervals (i) if $\langle ba, i \rangle \in V_t(\pi_1)|_2 \cup V_t(\pi_2)|_2$, then $\langle i, h_2 \rangle \in \text{BAEV}(ba)$

From 1,3 and constraint BA-DRV which is derived from BA-CMP2, BA-R1, BA-R2, and R6 (See appendix C for derivation), we can derive:

- 4) For all world-histories $(h_2 \text{ and } h_3)$ if $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ and $h_3 \in F_{cl}((V_t(\pi_1)|_2 \cup V_t(\pi_2)|_2), h)$ then for all events (ev) and intervals (i) if $\text{PRIOR-PI2}(i)$ then $\langle i, h_2 \rangle \in ev \text{ iff } \langle i, h_3 \rangle \in ev$

Finally, from 1,2, and 4 we can derive the following which is true iff $V_s(\ulcorner (\text{EXECUTABLE } \pi_1) \urcorner, h)$ equals TRUE:

For all world-histories (h_2) if $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ then $V_s(\ulcorner (\text{OCC } \pi_1) \urcorner, h_2)$ equals TRUE

Figure 4.4-9

Soundness Proofs

AX-IFTR9)

- $$\vdash (\text{IF } (\text{EXECUTABLE } \pi_1) \\ (\text{IF } (\text{IFTRIED } \pi_2 (\text{IFTRIED } \pi_1 (\text{AND } (\text{OCC } \pi_1) (\text{OCC } \pi_2)))) \\ (\text{IFTRIED } \pi_1 (\text{IFTRIED } \pi_2 (\text{AND } (\text{OCC } \pi_1) (\text{OCC } \pi_2))))))$$

For an arbitrary V_s and world-history h , assume that both $V_s(\ulcorner \text{EXECUTABLE } \pi_1 \urcorner, h)$ and $V_s(\ulcorner (\text{IFTRIED } \pi_2 (\text{IFTRIED } \pi_1 (\text{AND } (\text{OCC } \pi_1) (\text{OCC } \pi_2)))) \urcorner, h)$ equal TRUE. From our assumptions and definition of EXECUTABLE " $(\text{EXECUTABLE } \pi) =_{\text{def}} (\text{IFTRIED } \pi (\text{OCC } \pi))$ ", we can derive:

- 1) For all world-histories $\{h_2\}$ if $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ then $V_s(\ulcorner (\text{OCC } \pi_1) \urcorner, h_2)$ equals TRUE
- 2) For all world-histories $\{h_2 \text{ and } h_3\}$ if $h_2 \in F_{cl}(V_t(\pi_2)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_1)|_2, h_2)$ then $V_s(\ulcorner (\text{OCC } \pi_1) \urcorner, h_3)$ and $V_s(\ulcorner (\text{OCC } \pi_2) \urcorner, h_3)$ equal TRUE

From 1,2 and the interpretation of OCC we can derive:

- 3) For all world-histories $\{h_2\}$ if $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ then for all basic actions $\{ba\}$ and intervals $\{i\}$ if $\langle ba, i \rangle \in V_t(\pi_1)|_2$ then $\langle i, h_2 \rangle \in \text{BAEV}(ba)$
- 4) For all world-histories $\{h_2 \text{ and } h_3\}$ if $h_2 \in F_{cl}(V_t(\pi_2)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_1)|_2, h_2)$ then for all basic actions $\{ba\}$ and intervals $\{i\}$ if $\langle ba, i \rangle \in V_t(\pi_1)|_2 \cup V_t(\pi_2)|_2$ then $\langle i, h_3 \rangle \in \text{BAEV}(ba)$

From 3,4, constraint BA-CMP2 and the definition of F_{cl} , we can derive:

- 5) For all world-histories $\{h_2 \text{ and } h_3\}$ if $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_2)|_2, h_2)$ then for all basic actions $\{ba\}$ and intervals $\{i\}$ if $\langle ba, i \rangle \in V_t(\pi_1)|_2 \cup V_t(\pi_2)|_2$ then $\langle i, h_3 \rangle \in \text{BAEV}(ba)$

From 4,5 and constraint BA-CMP1, we can derive:

- 6) For all world-histories $\{h_3\}$ there exists a world-history $\{h_2\}$ such that $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_2)|_2, h_2)$ iff there exists a world-history $\{h_4\}$ such that $h_4 \in F_{cl}(V_t(\pi_2)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_1)|_2, h_4)$

Finally, from 2 and 6 we can derive the following which is true iff $V_s(\ulcorner (\text{IFTRIED } \pi_1 (\text{IFTRIED } \pi_2 (\text{AND } (\text{OCC } \pi_1) (\text{OCC } \pi_2)))) \urcorner, h)$ equals TRUE:

For all world-histories $\{h_2 \text{ and } h_3\}$ if $h_2 \in F_{cl}(V_t(\pi_1)|_2, h)$ and $h_3 \in F_{cl}(V_t(\pi_2)|_2, h_2)$ then $V_s(\ulcorner (\text{OCC } \pi_1) \urcorner, h_3)$ and $V_s(\ulcorner (\text{OCC } \pi_2) \urcorner, h_3)$ equal TRUE

Figure 4.4-10

Chapter 5

Analyzing the Planning Problem

In this chapter, we analyze the planning problem using the logic that we have developed and show that the noted problems and limitations associated with state-based systems are circumvented by switching to this new framework. We pay particular attention to the interaction between plan instances, both sequential and concurrent, and to the *persistence problem* [McDermott 82], which is the problem of determining how long a property remains true in a formalism that allows simultaneous events

Our general conception of a planning problem can be given as follows:

Input

- IN1) a goal condition to be solved
- IN2) a description of the world in which the plan is to be executed (the planning environment)
- IN3) for each member *a1* of a set of simple actions, the conditions under which *a1* can be executed and the effects produced by *a1* (the action specifications)

Output

- OT1) a plan, which is a composition of simple actions, that can be executed and if executed achieves the goal in any world that meets the description given by the planning environment and the action specifications

Such a system must be able to perform these principle operations:

- OP1) determining the conditions under which a composition of simple actions can be executed together
- OP2) determining the combined effects of a composition of simple actions

For typical state-based planners such as STRIPS [Fikes&Nilsson 71] and NOAH [Sacerdoti 77], the goal condition is a set of properties that must hold immediately following plan execution. The planning environment is a description of the initial state in which the plan is to be executed, and the action specification consist of properties called preconditions, describing the states in which each simple action can be executed, and effects, which are properties that are produced after execution. A plan that solves the goal is either taken to be a linear sequence

of simple actions or a partially ordered set of simple actions to be linearized upon execution.

In our framework, goal conditions are described using interval logic statements. Thus, we can describe goals that refer to a set of conditions that hold at various times, not just an instantaneous state. This enables us to represent goals such as avoiding some condition while performing some task, achieving a collection of goals to be done in some specified order, and preventing an undesirable condition that possibly will happen.

Simple plan instances take the place of actions and composite plan instances take the place of action sequences. The composition of a set of plan instances refers to a plan instance that occurs iff all its components occur. Since each component has a time associated with it, we can form composite plan instances that have concurrent actions, ones that correspond to action sequences, and ones that have gaps in between the time when two components are executed.

Given goal G , and a set of sentences S describing the planning environment and action specifications, a composition of simple plan instances pi is sought that meets the following conditions:

- SG1) $S \models (\text{INEV } Ip (\text{IFTRIED } pi (\text{OCC } pi)))$
- SG2) $S \models (\text{INEV } Ip (\text{IF } (\text{OCC } pi) \ G))$
- SG3) $S \models (\text{POS } Ip (\text{NOT } G))$

$SG1$, $SG2$, and $SG3$ are the necessary and sufficient conditions under which " pi solves goal G with respect to S " is true. These are three of the four conditions that we put forth in chapter 2. The fourth condition " $S \models (\text{POS } Ip (\text{OCC } pi))$ " is not needed because if $SG1$ is true then " $S \models (\text{POS } Ip (\text{OCC } pi))$ " is true for any plan instance pi in the future of Ip (see appendix G).

In the rest of this chapter, we discuss the elements making up S (i.e. the action specifications and planning environment) and the relation between composite plan instances and their components. In section 5.1, we discuss the planning environment, which consists of a description of conditions that will possibly hold and conditions that will inevitable hold in the future of planning time. In section 5.2, we discuss the **executability conditions** of simple plan instances. Executability conditions in our system take the place of preconditions. In section 5.3, we discuss the interaction between plan instances, both concurrent and sequential, and discuss how these interactions relate to the composition of two plan instances. In section 5.4, we discuss the effects of both simple plan instances and composite plan instances. In section 5.5, we discuss the *persistence problem*. This brings to light some of the problems encountered when the STRIPS assumption is used in an inappropriate setting. We also demonstrate that the *persistence assumption*, as put forth by some authors [McDermott 82] [Hanks&McDermott 85] as a replacement to the STRIPS assumption, can lead to problems when reasoning about planning. In the final section, we describe plan

instances that maintain properties over intervals which we use in place of the persistence assumption when reasoning about plans.

5.1. The Planning Environment

The planning environment is given by statements having the form $\lceil (\text{INEV } I_p C) \rceil$ and $\lceil (\text{POS } I_p C) \rceil$, which describe conditions that are inevitable and possible at planning time I_p . Thus, we can specify conditions that possibly or inevitably hold while a plan is being executed, not just conditions that hold prior to execution, which are the only type of conditions that are treated in most state-based planners.¹

The agent can only plan around or work with conditions that are inevitable at planning time. The attempt of a plan instance in the future of planning time cannot prevent one of these conditions from holding. Conditions that are possible but not inevitable, can be classified into three different categories: i) conditions that the agent can bring about under all possible circumstances, ii) conditions that the agent cannot prevent under any possible circumstances, and iii) conditions that are influenced by both the external world and by the agent.

If under all possible circumstances at planning time I_p , the agent can bring about condition C (by performing a future plan instance), then the following holds:

PE1)

(INEV I_p

($\exists ?pi$ (AND (PRIOR I_p (TIME-OF $?pi$)) (IFTRIED $?pi$ C))))

PE1 says that it is inevitable at planning time I_p that there exists a plan instance pi in the future of I_p such that if pi is attempted then C would be true. A stronger condition than PE1 is given by:

PE1')

($\exists ?pi$ (INEV I_p (AND (PRIOR I_p (TIME-OF $?pi$)) (IFTRIED $?pi$ C))))

¹ As we mentioned, Vere's system [Vere 81] is a notable exception. He allows scheduled external events which correspond to inevitable occurrences in our system. He cannot, however, represent possible conditions that may be prevented

$PE1'$ is true if there is a particular plan instance that achieves C under all possible circumstances. $PE1$ may be true when there is a different plan instance that achieves C for each possible circumstance.²

If a condition cannot be prevented by the agent under any possible circumstances, then the following is true:

PE2)

(IF(PRIOR I_p (TIME-OF ? π_i))
(INEV I_p (IF C (IFTRIED ? π_i C))))

$PE2$ says that it is inevitable at I_p that if C happens to hold, then C would still hold no matter which future plan instance the agent were to attempt. If both $PE1$ and $PE2$ are false, then C is a condition that is influenced by both the external world and the agent.

Now, there are cases where the agent cannot prevent some condition under any circumstances, but could always bring about its negation. One example is where the agent cannot prevent the property "the mainframe is not operational" from happening, but could always bring about this state by deliberately crashing the machine. More generally, the classification of a property and that of its negation are independent for the three categories. Thus, there are nine different classifications for a property taking into account its negation.

If the agent cannot prevent a property or prevent its negation under any possible conditions, we say that the agent cannot affect this condition, or equivalently, the condition is out of the agent's control. If condition C is out of the agent's control at time I_p , then the following holds:

PE3)

(IF(PRIOR I_p (TIME-OF ? π_i))
(AND (INEV I_p (IF C (IFTRIED ? π_i C)))
(INEV I_p (IF (NOT C) (IFTRIED ? π_i (NOT C))))))

Just like properties that are inevitably true, the agent can only plan around properties out of its control that are both possibly true and possibly false. Any solution to a goal must provide for both possibilities. That is, if condition C is out of the agent's control and is both possibly true and possibly false, then any plan

² This distinction relates to the use of *conditional plan instances*. Consider a simple example. Suppose if condition A is true, π_1 brings about condition C , but π_2 does not, and if A is false then π_2 brings about C , while π_1 does not. In this case, condition $PE1$ holds. Whether $PE1'$ holds in this situation depends on whether there exists a conditional plan instances corresponding to "if CND is true then do π_1 else do π_2 ". Treating such plan instances, however, involves some complications and relates to some issues considered by Moore [Moore 80]. In section 7.2. we discuss these complications and issues.

instance that solves a goal must be executable and must bring about the goal when C is true and when C is false. Similar problems have been investigated by Stuart [Stuart 86] who presents a logic of action that distinguishes between angelic and demonic non-determinism. He describes these two forms by saying that angelic non-determinism corresponds to choices by the agent while demonic non-determinism corresponds to choices under the control of the external environment. Angelic non-determinism relates in our system to possibilities that the agent can bring about by choosing different plan instances. Demonic non-determinism relates to events out of the agent's control that possibly occur and possibly do not occur. In both Stuart's system and our system, we are interested in plans that achieve their goals under all possibilities out of the agent's control. Stuart also treats a form of demonic non-determinism that affects the amount of time that the planning agent's actions take to complete. In order to handle this in our framework, we would have to relax the restriction that a plan instance's time of occurrence is fixed over all possible world-histories.

5.4. Executability Conditions

In a traditional planning system, preconditions are given for each simple action as part of the specification of a planning problem. In our framework, **executability conditions** take the place of preconditions. If we say that interval logic statement EC describes the executability conditions for future plan instance pi , then we assume that pi is executable in all branches possible at planning time in which EC holds. Thus, if the specification of a planning problem is given by the set of sentences S , EC describes the executability conditions for pi with respect to S only if the following holds:¹

EC-CND)

$S \models (\text{INEV } I_p (\text{IF } EC (\text{EXECUTABLE } pi)))$

where I_p denotes planning time

Typically, in a planning system, preconditions are not given for each action individually. Instead, preconditions are given for a whole class by making use of function terms. The same may be done in our logic. For example, we might use the function term $\lceil (\text{move } loc1 \text{ } loc2) @ I \rceil$ to refer to the plan instance where the agent moves from $loc1$ to $loc2$ during interval I (For simplicity, we are assuming that there is a unique way to perform the event $\lceil (\text{move } loc1 \text{ } loc2) \rceil$ which is presupposed by our

¹ $EC-CND$ is not sufficient conditions for " EC describes the executability conditions for pi with respect to S " because, for one, $EC-CND$ holds for any logically false statement substituted for EC . Intuitively, if EC describes the executability conditions for pi with respect to S , then EC must not be a fallacy and there must be not exist any weaker conditions that satisfy $EC-CND$. It is problematic to precisely formalize the "weakest" relation. One property that we want, though, is that $C1$ is not weaker than $C2$ if $C1$ entails $C2$.

use of the @ function) The executability condition for $\lceil(\text{move } loc1 \ loc2)@I\rceil$ is that the agent is at $loc1$ just prior to execution time I . The relation between a "move" plan instance and its executability conditions can be given by:²

MV1)

(INEV I_p (IF ($\exists i0$ (AND (MEETS $?i0 \ ?i$) (HOLDS (at agt $?loc1$) $?i0$)
(EXECUTABLE (move $?loc1 \ ?loc2$)@ $?i$))))

where agt refers to the planning agent

Relation Between Executability and Preconditions

We have used the term "executability conditions", instead of "preconditions", because preconditions have been treated in an ad hoc manner and consequently have a variety of interpretations. In the next section, we show that executability conditions are more general than the preconditions that are found in state-based planners. In this section, we describe the different possible interpretations that can be given to the preconditions specifications that are found in a planner such as NOAH [Sacerdoti 77]. These interpretations were noted by Pollack [Pollack 86].

Pollack describes different interpretations for preconditions by showing how they can be encoded in a language that represents basic actions, standard conditions, and generation, which are all concepts from Goldman's theory of action [Goldman 70]. She uses Allen's logic of time and action [Allen 84] to represent her adaptation of Goldman's concepts (her objective for doing so is to describe a theory of plan recognition with invalid queries). She defines "executability" in terms of Goldman's concepts. Her use of this term coincides with our usage with the exception that she encodes executability in a first order theory, while we make use of modal connectives.³

The precondition specifications for some action in NOAH is given by i) a header action, ii) a precondition list, and iii) a body, which consists of a sequence of actions specifying a way of executing the header action. Typically, a specification is given for a whole class of actions by using function terms. We will just show how one ground instance is translated. We let A stand for the event associated with the header action, B stand for the event associated with the sequence of actions in the body, and P stand for a term that denotes the conjunction of the conditions in the precondition list. We take some liberties in showing how Pollack's set of possible

² We are glossing over the fact that we want to restrict the variables $?loc1$ and $?loc2$ to only range over locations. Secondly, we might want to restrict the arguments to $\lceil(\text{move } ?loc1 \ ?loc2)@?i\rceil$ so that the duration of $?i$ is greater than or equal to the minimal time it takes the agent to get from $?loc1$ to $?loc2$.

³ In section 3.3.3, we related plan instance attempts to Goldman's concepts. This served to relate executability to these concepts since " p_i is executable" is defined as "if p_i were to be attempted then p_i would occur"

translations are described using our formal language. These translations are given in figure 5-3.1.

The header-body-precondition specifications allow one to describe a header action with two different bodies to indicate that there are (at least) two different ways to perform the action. In our formalism, it does not make sense to talk about two different ways of performing some plan instance since these objects refer to a set of events at specified times to be brought about by a particular execution. We can, however, relate plan instances and header-body-precondition specifications. The particular execution associated with a plan instance is given by a set of basic actions at specified times (see section 3.2.3). A plan instance is executable if i) the standard conditions associated with the set of basic actions hold and ii) if these basic actions were to be executed then the events associated with the plan instance would occur. This suggests the following translation of header-body-precondition specification into executability condition specifications for the case where the body consists of a set of basic actions. In this case, each header body pair $\langle A, B \rangle$ corresponds to the set of plan instances associated with event A occurring by executing the basic action instances in B. A particular plan instance can be picked out by specifying a time of occurrence. Using this translation, our use of executability conditions most closely resembles the interpretation of preconditions given by i) in figure 5.3-1. In this case, preconditions relate an event to be brought about with the particular execution that is performed in an attempt to bring about the event.

Different Types of Executability Conditions

Executability conditions are more general than preconditions that are used in state-based systems. In these systems, an action's preconditions are properties that describe the state in which the action is to be executed. In our system, a plan instance's executability conditions may have any temporal relation with the plan instance and may include statements about event occurrences along with statements about properties.

Although there is no restriction imposed on the temporal relation between a plan instance and its executability conditions, typically, executability conditions hold prior to or during plan instance execution. The reason for this is that the truth of $\lceil \text{(EXECUTABLE } pi) \rceil$ cannot be determined by conditions that possibly occur in the future of pi . This is reflected by the following theorem:

EC1)

(OR (INEV (TIME-OF pi) (EXECUTABLE pi))
(INEV (TIME-OF pi) (NOT (EXECUTABLE pi))))

Statement EC1 says that it is inevitable at pi 's time of occurrence that pi is executable or inevitable that pi is not executable. Consequently, it does not make sense to use executability conditions that are both possibly true and possibly false at a time in the future of pi 's time of occurrence.

Possible translations of:

Header: *A*

Body: *B*

Preconditions: *P*

- i) The performance of *B* generates the action corresponding to event *A* under conditions *P*. This entails the following relation:

```
(INEV Ip
  (IF (AND (HOLDS P ?i) (OCCURS B ?i))
    (OCCURS A ?i)))
```

- ii) "Property *P* holding during *i*" is a necessary condition under which *A* can occur during *i*. Thus, the following must hold:

```
(INEV Ip
  (IF (OCCURS A ?i) (HOLDS P ?i)))
```

- iii) "Property *P* holding during *i*" is a necessary condition under which *B* can occur during *i*. Thus, the following must hold:

```
(INEV Ip
  (IF (OCCURS B ?i) (HOLDS P ?i)))
```

- iv) *A* is executable during *i* if *P* holds during *i*:

```
(INEV Ip
  (IF (HOLDS P ?i) (EXECUTABLE A@?i)))
```

note: one might qualify iv) by putting a restriction on the duration of ?i

- v) *B* is executable during *i* if *P* holds during *i*:

```
(INEV Ip
  (IF (HOLDS P ?i) (EXECUTABLE B@?i)))
```

note: one might qualify v) by putting a restriction on the duration of ?i

-OR-

a variation on ii) - v) where the precondition holds immediately prior to the time when *A* and *B* occur, instead of during their time of occurrence.

Figure 5.3-1

A plan instance's executability conditions may include a condition describing another plan instance's occurrence, i.e. a statement of the form $\lceil (\text{OCC } pi) \rceil$. The use of these conditions may lead to an implementation that is more efficient than state-based systems, where preconditions only refer to properties. This is explained as follows. In a state-based system, if an action's preconditions do not hold in the initial situation, then an earlier action or set of actions must be introduced to bring about these preconditions. Thus, preconditions are used as intermediate values to determine the actions that enable another's preconditions. This intermediate step can be eliminated by using the OCC predicate as part of the executability specification. A plan instance can be directly related to the plan instance or set of plan instance that enables it. For example, if under all possible circumstances at planning time the occurrence of $pi2$ enables pi , then we could specify $\lceil (\text{OCC } pi2) \rceil$ as executability conditions for pi .⁴

Lanksy [Lanksy 85] presents a representation where an enablement relation (which she calls *the causal relation*) between two events is treated as a primitive relation, this being analogous to an enablement relation between two plan instances in our system. In this formalism, events are treated as primitive objects, while properties, if included, are defined in terms of events. This is in direct contrast to state-based systems where states, and properties which describe them, are treated as the primitive objects, while events are defined as mappings from state to state. In her work, she argues that a more succinct description and more efficient planning algorithm can be achieved by working in a language where events are directly related. In section 5.3, we show that the description of two actions that cannot occur simultaneously is awkward when only using property preconditions while straight forward when using a language such as ours or Lanky's that represents direct relations between events.

⁴ Savings such as these have been exploited in the state-based planning paradigm under the guise of *triangle tables* [Fikes&Hart&Nilsson 72]. Triangle tables store the relations between the preconditions and effects of a sequence of actions. One can use these table to enter a sequence of operators into a plan without the need to re-compute the enablement relations within the sequence. For example, if action $a1$ is followed by $a2$ in a triangle table and $a1$'s effects achieves $a2$'s preconditions, then the planner does not have to actively achieve $a2$'s preconditions when introducing the sequence $a1;a2$ into a plan (assuming that no other action is inserted between $a1$ and $a2$))

Interaction Between Plan instances and Simultaneous External Events

An important feature of our logic is the ability to model executability conditions that hold while the plan instance occurs. This allows us to represent different forms of interactions between plan instances and events that occur simultaneous with them. We describe three types of interaction by considering a simple resource conflict example where the planning agent shares a terminal with some other agent.

Let $\lceil(\text{use-terminal } \text{agt})\rceil$ refer to the event "the agent *agt* uses the terminal". We let the term *agtp* refer to the planning agent and *agt2* refer to another agent that may also use the terminal. In all cases examined below, we assume that it is inevitable at time *I_p* that the planning agent cannot use the terminal at the same time as the other agent:

```

UT1)
  (IF (NOT (DISJOINT ?i ?i2))
    (INEV Ip
      (NOT (AND (OCCURS (use-terminal agtp) ?i)
        (OCCURS (use-terminal agt2) ?i2))))))

```

The first case to consider is where, under all circumstances possible at planning time *I_p*, agent *agt2* has priority over the planning agent. We assume that it is inevitable at *I_p* that if *agt2* is using the terminal then *agtp* must wait until *agt2* is done. Furthermore, we assume that it is inevitable at *I_p* that *agt2* can interrupt *agtp* at any time and gain use of the terminal. As a consequence, a necessary condition under which it is inevitable at *I_p* that $\lceil(\text{use-terminal } \text{agtp})@I\rceil$ is executable is that it is inevitable at *I_p* that $\lceil(\text{use-terminal } \text{agt2})\rceil$ does not occur during any time that overlaps with *I*. For simplicity, we also assume that this condition is sufficient for the executability of $\lceil(\text{use-terminal } \text{agtp})@I\rceil$. This can be expressed in our language as follows:

```

UT2)
  (INEV Ip
    (IFF (EXECUTABLE (use-terminal agtp)@?i)
      (NOT (∃?i2 (AND (NOT (DISJOINT ?i ?i2))
        (OCCURS (use-terminal agt2) ?i2))))))

```

Under this "priority scheme", in order to guarantee that it is inevitable at *I_p* that the planning agent can use the terminal during interval *I₁*, it must not be possible at *I_p* that *agt2* will be using the terminal during any time during *I₁*.

The second case to consider is the converse of the above priority relation, where *agtp* has priority over *agt2*. Thus, we assume that it is inevitable at *I_p* that if *agtp* is using the terminal then *agt2* must wait until *agtp* is finished. Furthermore, we assume it is inevitable at *I_p* that *agtp* can interrupt *agt2* at any time and gain use of the terminal. If terminal availability is the only condition needed for executability, then $\lceil(\text{use-terminal } \text{agtp})@I\rceil$ is executable under all conditions possible at time *I_p*:

UT3)

(INEV I_p (EXECUTABLE (use-terminal agtp)@?i))

Under this priority relation, it is inevitable at I_p that the planning agent can prevent the other agent from using the terminal just by using the terminal itself:

UT4)

(INEV I_p
 (IFTRIED (use-terminal agtp)@?i
 (NOT (OCCURS (use-terminal agt2) ?i1))))

The third relation that we examine is the case where the first agent who tries to use the terminal gets the terminal until its done. If both agents try to use it at the same time then we assume that the planning agent gets the terminal. In this case, it is inevitable at I_p that \lceil (use-terminal agtp)@ I \rceil is executable iff $agt2$ has not started to use the terminal before I begins and is still using it at least through the beginning of I :

UT5)

(INEV I_p
 (IFF (EXECUTABLE (use-terminal agtp)@?i)
 (NOT ($\exists ?i2$ (AND (OR (OVERLAP ?i0 ?i)
 (FINISHES ?i ?i0) (DURING ?i ?i0))
 (OCCURS (use-terminal agt2) ?i0)))))))

In the above examples, we have illustrated that we can make a finer distinction than just saying that a plan instance and an event occurrence that overlap in time cannot occur together. Their "priority relation" can also be specified by indicating whether the event occurrence prevents the plan instance from (successfully) occurring if attempted, or whether the attempt of the plan instance prevents the event from occurring.

These type of priority relations do not arise in connection with overlapping plan instances. Just like the relation between a plan instance and an overlapping event occurrence, it is possible that two overlapping plan instances can occur separately, but not together (which we will discuss in the next section). There is no need to worry about their priority relation, however. This is because it is under the agent's control to choose which plan instance is to be executed. If there is a priority relation between two conflicting plan instances, the agent could always execute the one with lower priority by simply choosing not to perform the one with the higher priority. On the other hand, if there is an external event with higher priority that possibly occurs, it is not in the agent's control to choose not do it or to prevent it from occurring. Consequently, when planning, the agent must find a plan that works whether this external event occurs or not.

5.3. Plan Instance Interactions

One of the essential features of our logic is that it provides a formal basis for determining when two or more plan instances, concurrent or sequential, can be executed together. In a state-based formalism, the interactions of interest involve an action enabling a later one's preconditions, and an action interfering with a later one's preconditions. These interactions only concern linearly ordered actions. In our formalism, concurrent interactions must also be treated. In this section, we first examine the interaction between plan instances that do not overlap in time and then consider concurrent interactions.

Consider two plan instances $pi1$ and $pi2$ that do not have overlapping execution times. Without loss of generality assume that $pi1$'s execution time is prior to $pi2$'s execution time. In this case, the composition of $pi1$ and $pi2$ is executable iff i) $pi1$ is executable and ii) if $pi1$ were to be attempted then $pi2$ would be executable. That is, the following is a theorem in our logic, the proof of which is given in appendix G:

SEQ-TH1)

(IF (PRIOR (TIME-OF $pi1$) (TIME-OF $pi2$))
 (IFF (EXECUTABLE (COMP $pi1$ $pi2$))
 (AND (EXECUTABLE $pi1$)
 (IFTRIED $pi1$ (EXECUTABLE $pi2$))))))

It is important to reiterate that statements in our logic are interpreted with respect to a branch in a tree of possible futures or what we have called world-histories in our semantic model. Thus, SEQ-TH1 describes the interaction between two non-overlapping plan instances with respect to the circumstances given by a branch. Now, when solving a planning problem, the planner must determine whether or not the composition of two plan instances is executable in all branches that are possible at planning time I_p . Consequently, we are interested in the conditions under which it is inevitable at I_p that $\uparrow(\text{COMP } pi1 \text{ } pi2)$ is executable. This relation is given by:

SEQ-TH2)

(IF (PRIOR (TIME-OF $pi1$) (TIME-OF $pi2$))
 (IFF (INEV I_p (EXECUTABLE (COMP $pi1$ $pi2$))
 (INEV I_p (AND (EXECUTABLE $pi1$)
 (IFTRIED $pi1$ (EXECUTABLE $pi2$))))))

Theorem SEQ-TH2 may be derived using SEQ-TH1, the proof being given in appendix G.

Using SEQ-TH2, we can derive that for $pi1$ and $pi2$ in the future of I_p and $pi1$ prior to $pi2$, if EC1 refers to the executability conditions for $pi1$ and the occurrence of

$pi1$ brings about $EC1$, then it is inevitable at Ip that $\lceil (COMP\ pi1\ pi2) \rceil$ is executable if $EC1$ holds:

SEQ-TH3)

```
(IF(AND (PRIOR Ip (TIME-OF pi1))
        (PRIOR (TIME-OF pi1) (TIME-OF pi2)))
  (IF(AND (INEV Ip (IF EC1 (EXECUTABLE pi1)))
        (INEV Ip (IF EC2 (EXECUTABLE pi2)))
        (INEV Ip (IF (OCC pi1) EC2)))
    (INEV Ip (IF EC1 (EXECUTABLE (COMP pi1 pi2))))))
```

The relation given by *SEQ-TH3* is analogous to the relation in situation calculus: if action $a1$ brings about action $a2$'s preconditions, then the preconditions for the sequence $a1:a2$ hold in situation s if $a1$'s preconditions hold in s .

Composing Concurrent Plan instances

We now examine the composition of plan instances that overlap in time. There are a number of different ways that two overlapping plan instances may interact. Just like the relation between two plan instances that do not overlap in time, one concurrent plan instance may be executable and if it were to occur, then the other plan instance would be executable. As we will shortly see, this type of interaction between overlapping plan instances differs from the non-overlapping case because it does not necessarily follow that their composition is executable.

A second interaction is where two concurrent plan instances are executable individually, but their composition is not. This situation arises if two plan instances interfere with each other. Examples of this is where two plan instances share the same resource and where two plan instances are alternative choices, one of which can be executed at one time. Consequently, it is incorrect to define the executability conditions for a composite plan instance as the conjunction of its component's executability conditions.

The converse of the above statement does not hold either. As we just mentioned, it might be the case that $\lceil (COMP\ pi1\ pi2) \rceil$ is executable while $pi2$ is not because the occurrence of $pi1$ brings about the conditions under which $pi2$ is executable. There are also examples where $\lceil (COMP\ pi1\ pi2) \rceil$ is executable, but neither $pi1$ or $pi2$ is executable. Such is the case if $pi1$ and $pi2$ are *truly parallel actions*, ones that must be executed together. An example of this is where an object is lifted by applying pressure to two ends of the object, one hand at each end. If pressure were applied to only one end, the result would be a pushing action, not part of a lifting action.

If $pi1$ is earlier than $pi2$, but the two plan instances overlap in time, the following relation, which holds between two non-overlapping plan instances, may not hold: if $pi1$ is executable and an attempt of pi would make $pi2$ executable, then $\lceil (COMP\ pi1\ pi2) \rceil$ is executable. We show that this relation, which we refer to by *OVRLP**, is not

valid by constructing a world-history within some legal model where a statement having *OVRLP1**'s form does not hold.

Consider the function term $\lceil(\text{walk home store})@I2\rceil$ which denotes the plan instance where the agent walks from home to the store during interval *I2*. We assume this plan instance is executable in any world-history in the model where the agent is at home just prior to execution. The effects of this plan instance are that the agent is outside during execution and is at the store at a time immediately following execution. Let us also consider the term $\lceil(\text{stay-at home})@I1\rceil$ which refers to the plan instance where the agent stays at home during interval *I1*. This plan instance is executable as long as the agent is at home just prior to execution and its effects are that the agent is at home during execution. Assume that interval *I1* starts before but also overlaps interval *I2*. Consequently, $\lceil(\text{walk home store})@I2\rceil$ is executable in any world-history where $\lceil(\text{stay-at home})@I1\rceil$ occurs. This is because if $\lceil(\text{stay-at home})@I1\rceil$ occurs then the agent will be at home during *i1*, and in particular, the agent will be at home just prior to *i2* since we are assuming interval *I1* starts before and overlaps interval *I2*. We also assume that there are no world-histories in which both plan instances occur together, this being a consequence of the principle that the agent cannot be at two places at once. In particular, the agent cannot be at home and on its way to the store during the time common to the overlapping intervals *I1* and *I2*.

Let *h* be a world-history in which $\lceil(\text{stay-at home})@I1\rceil$ is executable. This is equivalent to saying that the following statement holds at *h*:

$(\text{IFTRIED}(\text{stay-at home})@I1 (\text{OCC}(\text{stay-at home})@I1))$

Since we are assuming that $\lceil(\text{walk home store})@I2\rceil$ is executable in any world-history where $\lceil(\text{stay-at home})@I1\rceil$ occurs, the following holds at *h*:

$(\text{IFTRIED}(\text{stay-at home})@I1 (\text{EXECUTABLE}(\text{walk home store})@I2))$

Thus, we find that the antecedent of *OVRLP1**, substituting $\lceil(\text{stay-at home})@I1\rceil$ for *pi1* and $\lceil(\text{walk home store})@I2\rceil$ for *pi2*, holds at *h*. *OVRLP1**'s consequent, with the same substitution, does not hold at *h* since the two plan instances cannot occur together in any world-history. Thus, at world-history *h*, the negation of *OVRLP1** with the above substitutions is satisfiable and consequently the relation described by *OVRLP1** cannot be a theorem.

The next case to be examined is where it is inevitable at planning time that two plan instances are executable. We look at the cases where i) It is inevitable at planning time that two plan instances interfere, ii) inevitable they do not interfere, and iii) the case where two plan instances conditionally interfere.

We start with a simple resource conflict example. Consider a scenario where there is a number of burners on a stove which may be used to heat some pan. Let the function term $\lceil(\text{heating pn brnr})@I\rceil$ denote the plan instance where pan *pn* is being heated on burner *brnr* during the interval *I*. It is inevitable at planning time *I_p* that

one burner cannot be used to heat two different pans at the same time and one pan cannot be heated on two different burners at the same time:

HT1)

```
( $\forall$  ?pnx ?pny ?brnrx ?brnry ?ix ?iy
  (INEV Ip
    (IF (AND (OCC (heating ?pnx ?brnrx)@?ix)
              (OCC (heating ?pny ?brnry)@?iy))
      (OR (DISJOINT ?ix ?iy)
           (AND (NOT (= ?pnx ?pny))
                  (NOT (= ?brnrx ?brnry)))
           (AND (= ?pnx ?pny) (= ?brnrx ?brnry)
                  (= ?ix ?iy))))))
```

HT1 says that under all circumstances possible at time *Ip*, if two "heating" plan instances occur then either i) their times of occurrence are not overlapping, ii) they refer to different pans and different burners, or iii) they refer to the same plan instance (we are implicitly assuming that (heating *pnx brnrx*)@*ix* and (heating *pny brnry*)@*iy* denote different plan instances if they disagree on any of their three arguments). This type of relationship between plan instances is what Lansky [Lansky 85] calls a *behavioral constraint*; it is a constraint that directly relates two plan instances (actions) instead of indirectly relating two plan instances by use of action precondition-effect lists. Lansky also refers to the relation between an action and an earlier one that enables it, which we mentioned in section 5.2, as a behavioral constraint.

Using precondition-effect lists, it is awkward to represent the behavioral constraint captured by *HT1*. First of all, properties such as "burner *brnr* is in use" and "pan *pn* is in use" would have to be introduced. Secondly, the action "heat pan *pn* on *brnr*" could not be treated as a simple action, one modeled by a precondition-effect list. Instead, this action would have to be treated as two simple actions to be performed consecutively (Vere's system [Vere 81] has such a facility). The reason for this is that "burner *brnr* is in use" holds during the execution of "heat pan *pn* on *brnr*" and ceases to hold immediately following execution, similarly for the property "pan *pn* is in use". In a state-based system these type of effects can be modeled by consecutive actions where the earlier action's effect is that the object (i.e. burner or pan) is in use and the later one's effect is that the object is free. As noted in section 1.2, SIPE [Wilkins 83] has a special mechanism for treating a limited class of resource conflicts. This mechanism could be used to solve the above example without recourse to the "in use" properties.

In order for the behavioral constraint *HT1* to be useful, it must lead to the conclusion that a composite plan instance consisting of two overlapping *heating* plan instances using the same burner but different pans, or different burners but the same pan, is not executable. This is easily shown. Let us consider plan instances $\lceil (\text{heating } pn1 \text{ brnr})@I1 \rceil$ and $\lceil (\text{heating } pn2 \text{ brnr})@I2 \rceil$ in the case where *pn1* and *pn2* denote different objects and the intervals *I1* and *I2* overlap in time. We want to

prove that it is inevitable at I_p that the composition of $\lceil(\text{heating } pn1 \text{ brnr})@I1\rceil$ and $\lceil(\text{heating } pn2 \text{ brnr})@I2\rceil$ is not executable, i.e.,

HT2)

(INEV I_p

(NOT (EXECUTABLE (COMP (heating $pn1$ brnr)@ $I1$
(heating $pn2$ brnr)@ $I2$))))

From i) statement $HT1$, ii) $I1$ and $I2$ overlap in time and are later than I_p , and iii) $pn1$ and $pn2$ are unequal, it logically follows that $HT2$ is true. A proof of this is given in appendix G. A similar proof can also be given to show that two overlapping *heating* plan instances using the same pan, but different burners, are not executable under any possible conditions.

The relation given by $HT1$ only describes when two *heating* plan instances cannot be done together. $HT1$ does not specify when two *heating* plan instances can be done together. Typically, we would want to model the above scenario so that any two *heating* plan instances that meet the "constraints" given in the consequent of $HT1$ may be executed together if they can be executed individually. That is, we would want the following to hold:

HT3)

(IF (OR (DISJOINT ?ix ?iy)
(AND (NOT (= ?pnx ?pny)) (NOT (= ?brnrx ?brnry)))
(AND (= ?pnx ?pny) (= ?brnrx ?brnry) (= ?ix ?iy)))
(IF (AND (INEV I_p (EXECUTABLE (heating ?pnx ?brnrx)@?ix))
(INEV I_p (EXECUTABLE (heating ?pny ?brnry)@?iy)))
(INEV I_p (EXECUTABLE (COMP (heating ?pnx ?brnrx)@?ix
(heating ?pny ?brnry)@?iy))))))

The above statement is true iff for any two *heating* plan instances $pi1$ and $pi2$ that meet the constraints in the antecedent, if it is inevitable at I_p that both $pi1$ and $pi2$ are executable individually then it is inevitable at I_p that their composition is executable.

Relation $HT3$ is not a theorem; if two *heating* plan instances refer to different pans and different burners, it does not necessarily follow that these two plan instance can be executed together. In order to derive that they can be done together, they must not interfere with each other. If plan instance $pi1$ does not interfere with $pi2$ under all circumstances possible at I_p , then the following holds:

NI-CND)

(AND (INEV I_p (IF (OCC $pi1$) (IFTRIED $pi2$ (OCC $pi1$))))
(INEV I_p (IF (OCC $pi2$) (IFTRIED $pi1$ (OCC $pi2$))))))

$NI-CND$ is true iff it is inevitable at I_p that if $pi1$ occurs and $pi2$ were to be attempted then $pi1$ would (still) occur, and if $pi2$ occurs and $pi1$ were to be attempted then $pi2$ would (still) occur. If plan instance $pi1$ and $pi2$ are related by $NI-CND$, and it is

inevitable at I_p that both $pi1$ and $pi2$ are individually executable, then it is inevitable at I_p that $\lceil(\text{COMP } pi1 \ pi2)\rceil$ is executable:

NI-TH)

```
(IF(AND (INEV  $I_p$  (IF (OCC  $pi1$ ) (IFTRIED  $pi2$  (OCC  $pi1$ ))))
      (INEV  $I_p$  (IF (OCC  $pi2$ ) (IFTRIED  $pi1$  (OCC  $pi2$ ))))))
 (IF(AND (INEV  $I_p$  (EXECUTABLE  $pi1$ ))
        (INEV  $I_p$  (EXECUTABLE  $pi2$ ))
        (INEV  $I_p$  (EXECUTABLE (COMP  $pi1 \ pi2$ ))))))
```

Plan Instances that Conditionally Interfere

Although two plan instances having the form $\lceil(\text{heating } pn \ brnr)@I\rceil$ interfere with each other depending on the arguments to the heating function, two particular *heating* plan instances either interfere under all possible conditions or do not interfere under all possible conditions. We now examine two plan instances that conditionally interfere. Suppose that the agent can always carry on one suitcase for a plane flight, but can only carry on two suitcases if the plane is not completely full. We assume that this relation is inevitable at the time of planning I_p . Let the function term $\lceil(\text{carry } sc)@I\rceil$ refer to a plan instance where the agent carries on suitcase sc for the plane flight during interval I . If we assume that the terms $sc1$ and $sc2$ denote distinct suitcases, we have the following relation:

CR1)

```
(INEV  $I_p$ 
  (IF(AND (OCC  $\lceil(\text{carry } sc1)@I\rceil$ ) (OCC  $\lceil(\text{carry } sc2)@I\rceil$ )
        (HOLDS plane-not-full  $I$ )))
```

Now, $CR1$ alone does not indicate that $\lceil(\text{HOLDS plane-not-full } I)\rceil$ is a necessary condition needed in order to execute $\lceil(\text{carry } sc1)@I\rceil$ and $\lceil(\text{carry } sc2)@I\rceil$ together. It might be the case that $\lceil(\text{HOLDS plane-not-full } I)\rceil$ is the joint effect of executing $\lceil(\text{carry } sc1)@I\rceil$ and $\lceil(\text{carry } sc2)@I\rceil$ together.

Allen's and Koomen's system [Allen&Koomen 83b] cannot distinguish between the case where $\lceil(\text{HOLDS plane-not-full } I)\rceil$ is a necessary condition needed to execute $\lceil(\text{carry } sc1)@I\rceil$ and $\lceil(\text{carry } sc2)@I\rceil$ together from the case where $\lceil(\text{HOLDS plane-not-full } I)\rceil$ is their joint effect. This lead to some problems that we noted earlier in section 2.3. Suppose, using their system, the planning environment specifies that the material implication embedded within $INEV$ in $CR1$ holds, but the planning environment does not specify that $\lceil(\text{HOLDS plane-not-full } I)\rceil$ is either true or is false. If both $\lceil(\text{carry } sc1)@I\rceil$ and $\lceil(\text{carry } sc2)@I\rceil$ are executable individually (their preconditions hold), the system would allow a plan with both plan instances in it. The reason for this is that a statement saying that both $\lceil(\text{carry } sc1)@I\rceil$ and $\lceil(\text{carry } sc2)@I\rceil$ occur is consistent (in interval logic) with the material implication embedded within $INEV$ in $CR1$. The behavior of Allen's and Koomen's system is

correct if $\lceil(\text{HOLDS plane-not-full } I)\rceil$ is the joint effect of the two plan instances. On the other hand, their system leads to incorrect results if $\lceil(\text{HOLDS plane-not-full } I)\rceil$ is a necessary condition under which the two plan instances can be executed together. If the agent cannot bring about $\lceil(\text{HOLDS plane-not-full } I)\rceil$, then we would want the planner to conclude that the plans success is contingent on whether $\lceil(\text{HOLDS plane-not-full } I)\rceil$ is true. Conversely, if the agent can bring about $\lceil(\text{HOLDS plane-not-full } I)\rceil$, we would want the plan to contain a step that brings about this condition.

In our logic, we can state that $\lceil(\text{HOLDS plane-not-full } I)\rceil$ is a necessary condition under which $\lceil(\text{carry sc1})@I\rceil$ and $\lceil(\text{carry sc2})@I\rceil$ can be executed together:

```
CR2)
  (INEV Ip
    (IF (EXECUTABLE (COMP (carry sc1)@I (carry sc2)@I))
      (HOLDS plane-not-full I)))
```

A specification of a planning problem may explicitly encode relations such as *CR2*. For example, the planning algorithm we present in chapter six requires the user to specify, for each pair of overlapping plan instances, the conditions under which they do not interfere. One, however, can also derive such relations in our logic from other facts about the world. For example, *CR2* is true if the agent cannot bring about the property *plane-not-full* at any time under any circumstances possible at *Ip*. This relation can be given by:

```
CR3)
  (INEV Ip
    (IF (NOT (HOLDS plane-not-full ?i))
      (IFTRIED ?pi (NOT (HOLDS plane-not-full ?i)))))
```

In appendix G, we show that *CR2*, can be derived from *CR1* and *CR3* (if *I* is in the future of *Ip*). We can even prove something stronger from *CR1* and *CR3*:

```
CR4)
  (INEV Ip
    (IF (NOT (HOLDS plane-not-full I))
      (NOT (IFTRIED ?pi
        (EXECUTABLE (COMP (carry sc1)@I (carry sc2)@I)))))
```

CR4 says that it is inevitable at *Ip* that if $\lceil(\text{NOT (HOLDS plane-not-full } I)\rceil$ holds then there is no plan instance that the agent can perform to make $\lceil(\text{COMP (carry sc1)@I (carry sc2)@I})\rceil$ executable.

In the above example, *CR1* relates two concurrent plan instances to a condition that holds during their execution time. If we modified this specifications so that it relates two concurrent plan instances to a condition that holds prior to their

execution time, then we can prove that this condition is a necessary condition, not a joint effect. That is, from:

CR5)
 (AND (Prior I0 I)
 (INEV Ip
 (IF (AND (OCC (carry sc1)@I) (OCC (carry sc2)@I))
 (HOLDS plane-not-full I0))))))

we can derive:

CR6)
 (INEV Ip
 (IF (EXECUTABLE (COMP (carry sc1)@I (carry sc2)@I))
 (HOLDS plane-not-full I0))))

Truly Parallel Plan Instances

We conclude this section by describing an example where we compose two "truly parallel" plan instances, ones that must be executed together in order to occur. Consider a situation where the agent may lift an object by applying pressure to two ends of the object, one hand at each end. If pressure is applied to only one end, the result is a pushing action, not part of a lifting action. Let *lift-left@I* refer to the plan instance that occurs if applying pressure to the left end of the object and raising this arm results in a lifting event. Similarly, let *lift-right@I* refer to the plan instance that occurs if applying pressure to the right end of the object and raising this arm results in a lifting event. To state that these two plan instances can only occur together we write:

TP1)
 (INEV Ip (IFF (OCC lift-left@I) (OCC lift-right@I)))

We introduce *pressure-left@I* to refer to the plan instance associated with the generator of *lift-left@I* (see section 3.2.3) and use *pressure-right@I* to refer to the plan instance associated with the generator of *lift-right@I*. Thus, saying that *lift-left@I* is attempted means that *pressure-left@I* occurs, similarly, for the connection between *pressure-right@I* and *lift-right@I*. Both these relations, which we assume to be inevitable at *Ip*, are given by:

TP2)
 (AND (INEV Ip (IFTRIED lift-left@I (OCC pressure-left@I))
 (INEV Ip (IFTRIED lift-right@I (OCC pressure-right@I))))

We also assume that it is inevitable at *Ip* that if *lift-left@I* were to be attempted then it would occur (i.e. it is executable) iff pressure is simultaneously being applied to the

right end. Similarly, we assume that the symmetric relation holds for *lift-right@I*. These relations are given by:

TP3)

(AND (IFF (OCC pressure-right@I) (EXECUTABLE lift-left@I))
(IFF (OCC pressure-left@I) (EXECUTABLE lift-right@I)))

Using *TP1 - TP3*, we can prove:

TP4)

(IF (NOT (INEV *I_p* (AND (OCC lift-left@I) (OCC lift-right@I))))
(AND (POS *I_p* (NOT (EXECUTABLE lift-left@I)))
(POS *I_p* (NOT (EXECUTABLE lift-right@I)))
(INEV *I_p* (EXECUTABLE (COMP lift-left@I lift-right@I)))))

TP4 says that if it is not inevitable at *I_p* that *lift-left@I* and *lift-right@I* both occur, then i) it is possible at *I_p* that *lift-left@I* is not executable, ii) it is possible at *I_p* that *lift-right@I* is not executable, but iii) inevitable at *I_p* that their composition is executable. In appendix G, we give a proof showing the derivation from *TP1 - TP2* to *TP4*.

5.4. Plan Instance Effects

In a traditional planning system, the specification of a planning problem includes the effects produced by each simple action. In these systems, an action's effects can be given by a set of properties that hold immediately after the action completes. The effects produced by a sequence of simple actions can be computed from the specification of the effects of the simple action's along with *deductive operators*, which we will shortly describe. In our framework, we will speak about the effects produced by a plan instance. A plan instance's effects can be given by an interval logic statement. In this section, we show how the effects produced by simple plan instances may be described and how the effects produced by a composite plan instance can be determined from its component's effects.

If we say that interval logic statement *EFF* is an effect of plan instance *pi*, then we assume that *EFF* holds in all branches possible at planning time *I_p* in which *pi* occurs. Thus, if the specification of a planning problem is given by the set of sentences *S*, *EFF* is an effect of *pi* with respect to *S* only if the following holds:¹

EFF-CND)

$$S \models (\text{INEV } I_p (\text{IF } (\text{OCC } p_i) \text{ EFF}))$$

This treatment of effects is more general than the treatment of effects in state-based systems. An action's effect in state-based systems refers to properties that hold immediately after the action is executed. In our framework, a plan instance's effects may refer to both events that occur and properties that hold while a plan instance is being executed along with referring to conditions that hold after execution.

Just like executability conditions, the effects for a whole class of plan instances may be described by a function term. For example, we might use the function term $\lceil (\text{grasp } obj)@I \rceil$ to refer to the plan instance where the agent grasps the object *obj* during interval *I*. The effect produced by $\lceil (\text{grasp } obj)@I \rceil$ is that the planning agent is holding *obj* at a time immediately following *I*. The relation between a "grasp" plan instance and its effect can be given by:

¹ *EFF-CND* is not sufficient grounds to conclude that "*EFF* is an effect of *pi* with respect to *S*" because, for one, *EFF-CND* is true for any logical truth that is substituted for *EFF*. It is only proper to say that *EFF* is an effect of plan instance *pi* if there is some connection between *pi*'s occurrence and *EFF* being true, that is, *pi* brings about *EFF*. Thus, we might want another necessary condition such as "if *pi* were not to be executed, then *EFF* would not be true". This condition could be represented in our framework, if we had a construct, *opposite* to $\lceil (\text{IFTRIED } p_i P) \rceil$, meaning that *P* would be true if *pi* were not to be attempted. This glosses over some complications such as situations where two agents bring about the same event.

```

GR1)
  (INEV Ip
    (IF (OCC (grasp ?obj)@?i)
      (∃?i2 (AND (MEETS ?i ?i2) (HOLDS (holding obj) ?i2))))))

```

Direct and Indirect Effects

In some planning systems [Wilkins 83] and formalisms [Georgeff 86], a distinction is made between an action's direct effects and its indirect effects. In state-based planning systems, direct effects refer to properties that are explicitly in the precondition-effect list. An action's indirect effects may be inferred from its direct effects using a *deductive operator*, a term which we use from Wilkins. A deductive operator specifies a relation between properties that hold in all states. For example, a deductive operator can be used to represent that a block is clear iff there is no block on top of it. Thus, if we specify that the action "stack block *b1* on block *b2*" has the direct effect "block *b1* is on block *b2*", we can use a deductive operator to infer the indirect effect "block *b2* is not clear". Without using deductive operators, both "block *b1* is on block *b2*" and "block *b2* is not clear" would have to be modeled as direct effects and would have to be explicitly included in the precondition-effect list. Thus, the use of deductive operators affords a more succinct planning problem description. Wilkins also notes that deductive operators can be used to specify conditional effects, something that cannot be done in systems such as NOAH [Sacerdoti 77] and NONLIN [Tate 77].

Georgeff uses the term *causal law* to describe statements that relate properties and events (In section 5.5, we describe Georgeff's theory, which treats simultaneous events, in some detail). Causal laws are more general than deductive operators since they relate events with events, properties with properties, properties with events, etc. They can be used to represent the relation between the "on" relation and the "clear" relation that we just described. They can also be used to describe a relation between events such as "If a cup is resting on a saucer and the saucer is moved then the cup also moves".

In our framework, we would encode causal laws so that they are relations between conditions that inevitably hold at planning time. For example, the causal law "if the vase is dropped, it will break" can be given by:

```

CL1)
  (INEV Ip
    (IF (OCCURS (drop vase) ?i)
      (∃?i2 (PRIOR ?i ?i2) (HOLDS (broken vase) ?i2))))

```

Statements having form *CL1* and statements describing the relation between a plan instance and its effects can be combined together. Given that *EFF* is the effect of some plan instance *pi* and that it is inevitable that if *EFF* holds then *EFF2* will also hold, we can derive that it is inevitable at planning time that if *pi* occurs then *EFF2*.

Thus, if $\lceil (\text{OCCURS (drop vase) } I2) \rceil$ is an effect of the plan instance where the agent pushes the vase off the table at a time $I1$ which is immediately prior to $I2$:

CL2)
 (INEV I_p
 (IF (OCC push-vase-off@ $I1$)
 (OCCURS (drop vase) $I2$)))

we can derive using *CL1* that it is inevitable at I_p that if the agent pushes the vase off during $I1$, the vase will break at a time after $I2$:

CL3)
 (INEV I_p
 (IF (OCC push-vase-off@ $I1$)
 ($\exists i3$ (PRIOR $I2$ $i3$) (HOLDS (broken vase) $i3$))))

When solving a goal, it does not matter whether a plan instance's direct effects achieves the goal, or whether the action's indirect effects achieves the goal. We are only interested in whether or not it is inevitable at planning time that if a plan instance occurs, then the goal holds. Moreover, our logic does not provide a basis for distinguishing between indirect and direct effects.

The Effects Produced by Composite Plan Instances

The relation between the effects of a composite plan instance and the effects of its components stems from the following relation. If *EFF1* is an effect of plan instance *pi1*, then *EFF1* holds if *pi1* occurs, no matter which other plan instances are executed in conjunction with *pi1*. That is, the following is a theorem in our logic:

CE1)
 (IF (INEV I_p (IF (OCC *pi1*) *EFF1*))
 (INEV I_p (IF (OCC (COMP *pi1 pi2*)) *EFF1*)))

CE1 can be interpreted as saying that if *EFF1* holds in all branches possible at I_p in which *pi1* occurs, then *EFF1* also holds in all branches possible at I_p in which both *pi1* and *pi2* occur together. Consequently, if *EFF1* is an effect of *pi1* and *EFF2* is an effect of *pi2*, both effects hold if the composition of *pi1* and *pi2* occurs:

CE2)
 (IF (AND (INEV I_p (IF (OCC *pi1*) *EFF1*))
 (INEV I_p (IF (OCC *pi2*) *EFF2*)))
 (INEV I_p (IF (OCC (COMP *pi1 pi2*)) (AND *EFF1 EFF2*))))

The converse of relation *CE2* does not necessarily hold, however. There may be effects that are only produced if two plan instances are executed together, such as

when an object can be lifted by using both arms together, but not by either of the arms individually.

The relation captured by *CE2* suggests a simple connection between non-linear planning [Sacerdoti 77] and our framework. If the goal condition consists of a conjunction of two conditions *G1* and *G2*, we can look for two plan instances *pi1* and *pi2* that respectively achieve *G1* and *G2*, knowing that if they occur together then both *G1* and *G2* will hold. This does not mean, however, that these two plan instances can be executed together. We must also show that it is inevitable that their composition is executable. As we illustrated in section 5.3, the relation between a composition's executability conditions and the executability conditions for its components is more complex than the relation for effects. It is not simply the case that if both *pi1*'s and *pi2*'s executability conditions hold, then it is inevitable that their composition is executable.

There is no relation analogous to *CE2* in situation calculus. That is, it is not necessarily true that if *EFF1* is an effect produced by action *a1*, and *EFF2* is an effect produced by *a2*, then the conjunction of *EFF1* and *EFF2* is an effect of a composition of *a1* and *a2* (a sequence). The reason for this is that the effects of a sequence are the effects produced when the sequence completes. In situation calculus, one does not say that the effects of sequence *a1;a2* are that *a1*'s effects are true between *a1*'s and *a2*'s execution and *a2*'s effects are true after *a2*'s execution. When *a1;a2* completes, *a1*'s effects may not hold, this being the case if *a2* negates these effects. Thus, to compute the effects of a sequence, one must know which properties that the actions in the sequence affect along with the properties that they do not affect. This involves a solution to the frame problem. For example, the effects of *a1;a2* are the effects of *a2* conjoined with the effects of *a1* that are not adversely affected by *a2*.

In our system, the above situation surfaces under a different guise. Suppose that we have two plan instances *ev1@I1* and *ev2@I2* where interval *I1* meets interval *I2*. Thus, the composition of *ev1@I1* and *ev2@I2* can be thought of a sequence. Also assume that *ev1@I1* makes property *pr1* true immediately following its execution and *ev2@I2* makes property *pr2* true immediately following its execution. These relations can be given by:

CE3)

```
(AND (INEV Ip (IF (OCC ev1@I1)
                  (∃?i (AND (MEETS I1 ?i) (HOLDS pr1 ?i))))))
      (INEV Ip (IF (OCC ev2@I2)
                  (∃?i (AND (MEETS I2 ?i) (HOLDS pr2 ?i))))))
```

Now, if both *ev1@I1* and *ev2@I2* occur together, it is not necessarily true that both *pr1* and *pr2* hold after this composition completes, i.e. immediately following *I2*. Rather, if this composition occurs, then *pr1* is true immediately after *I1*, and *pr2* is true immediately after *I2*:

CE4)

(INEV Ip

(IF (OCC (COMP *ev1@I1* *ev2@I2*))

(AND (∃?i (AND (MEETS *I1* ?i) (HOLDS *pr1* ?i)))

(∃?i (AND (MEETS *I2* ?i) (HOLDS *pr2* ?i))))))

CE4 says nothing about whether *pr1* holds immediately after *I2*. This would not be the case if the occurrence of *ev2@I2* negates *pr2*, or if some other event, that possibly occurs during *I2*, negates *pr1*. One way to insure that *pr1* remains true throughout interval *I2*, is to explicitly introduce a plan instance that maintains this property, if such a plan instance exists. Determining how long a property, that is true at some time, remains true and whether there is a "maintenance plan instance" that can be executed in conjunction with other plan instances is where the frame problem surfaces in our system. This and related issues are discussed in the next two sections.

5.5. Persistence and Simultaneous Events

In this section, we examine the *persistence problem* [McDermott 82] which is the problem of determining how long a property remains true in a formalism that allows simultaneous events. We first show that the STRIPS assumption (see section 1.2) is inappropriate in a formalism that allows simultaneous events. Recall that the STRIPS assumption is used to determine which properties remain constant from one state to the next when an event is applied. We then analyze the persistence problem by breaking it up into two steps: i) finding the set of possible events that can negate some property, and ii) determining whether any of these events possibly occur. Only the first step is related to the STRIPS assumption. We demonstrate that *the persistence assumption*, as put forth by some authors [McDermott 82] [Hanks&McDermott 85] as a replacement to the STRIPS assumption, hides these two steps and can lead to problems when reasoning about planning. In the following section (5.6), we investigate plan instances that maintain properties over intervals, which we use in place of the persistence assumption when reasoning about planning.

In situation calculus, the result of applying an event in a situation typically leaves most properties unchanged. The reason that a property does not change from situation to situation is not really due to the event that occurs between the situations, but instead is due to the non-occurrence of other events that can negate this property. This distinction is not made when using the STRIPS assumption but this is not problematic since in the state-based representation, only one event can occur between two successive situations. On the other hand, it is important to make this distinction in a formalism, such as ours, that represents simultaneous events. If property *pr* persists through interval *I*, we must attribute this persistence to the non-occurrence of any event that can negate property *pr* during interval *I*. The persistence is not attributed to the occurrence of an event or set of events that occurs during *I*. Georgeff makes a similar point in [Georgeff 86] where he describes a theory of action and events that models simultaneous events. Very roughly, Georgeff's theory is a modification and extension of situation calculus that can model simultaneous events. We will shortly describe his theory in some detail.

We clarify the above point with an example. So as not to distract from the main points and to facilitate a comparison between our work, situation calculus, and Georgeff's theory, we will work with a temporal theory that is simplified in the following ways. We assume that there is a set of intervals that are numbered by the non-negative integers. Interval *I*0 corresponds to the "initial situation" which meets interval *I*1 which meets *I*2 etc. An interval term such as *I*1-2 will be used to refer to the interval that is the concatenation of intervals *I*1 and *I*2. We also assume that events only occur during the odd numbered intervals and produce effects that hold throughout the following even numbered interval. Secondly, no change takes place during the even numbered intervals. Thus, either a property is true throughout an entire even interval or false throughout the entire interval (Allen and Hayes [Allen&Hayes 84] refer to these type of intervals as moments). This simplified theory is like situation calculus in that the world over time can be seen as an alternating sequence of events and static situations.

Consider an "initial situation" in which two blocks *A* and *B* are at location *loc1*:

S1)

```
(AND (HOLDS (at A loc1) I0)
      (HOLDS (at B loc1) I0))
```

Let the term $\lceil \text{move A loc2} \rceil$ refer to the event "block *A* is moved to location *loc2*". The result of this event is that block *A* is at location *loc2* following its occurrence. We assume that this connection is inevitable at time *I0*. The result of $\lceil \text{move A loc1} \rceil$ occurring during *I1* may be given by:

S2)

```
(INEV I0
  (IF (OCCURS (move A loc2) I1)
      (HOLDS (at A loc2) I2)))
```

If only one event can occur at a time, as in situation calculus, we would be able to infer that $\lceil \text{at B loc1} \rceil$ remains true throughout interval *I1-2* if $\lceil \text{move A loc2} \rceil$ occurs during *I1* and it does not affect this property. The STRIPS assumption could be used to make such an inference. On the other hand, in a representation that allows simultaneous events, this conclusion would be incorrect if it is possible at *I0* that there is an event that occurs simultaneously with $\lceil \text{move A loc2} \rceil$ that changes *B*'s location. For example, consider the event $\lceil \text{move B loc2} \rceil$ whose effect is that block *B* is at location *loc2* following its occurrence. Thus, the result of event $\lceil \text{move B loc2} \rceil$ occurring during interval *I1* may be given by:

S3)

```
(INEV I0
  (IF (OCCURS (move B loc2) I1)
      (HOLDS (at B loc2) I2)))
```

If we assume that it is possible at *I0* that both $\lceil \text{move A loc2} \rceil$ and $\lceil \text{move B loc2} \rceil$ occur simultaneously during *I1*,

S4)

```
(POS I0
  (AND (OCCURS (move A loc2) I1)
        (OCCURS (move B loc2) I1)))
```

then it is possible at *I0* that $\lceil \text{move A loc2} \rceil$ occurs during *I1* while block *B* ends up at location *loc2* at *I2*:

S5)
 (POS I0
 (AND (OCCURS (move A loc2) I1)
 (HOLDS (at B loc2) I2))))

In appendix G, we present a proof that demonstrates that *S5* follows from *S3* and *S4*. Now, from statements *S1* and *S5* along with a statement saying that an object cannot be at two locations at once, we can derive:

S6)
 (AND (HOLDS (at B loc1) I0)
 (POS I0
 (AND (OCCURS (move A loc2) I1)
 (NOT (HOLDS (at B loc1) I1-2))))))

Statement *S6* can be interpreted as saying that it is possible at *I0* that the property "block *B* is at location *loc1*" is negated during interval *I1-2* as block *A* is moved to location *loc2* during interval *I1*.

The Persistence Problem

We can analyze the persistence problem by breaking it up into two steps: i) finding the set of possible events that can negate some property and ii) determining whether any of these events occur. Using the simplified temporal theory that we described above, it is straightforward to describe the set of possible events that can negate some property. In this theory, a property that holds during an "even interval" will persist through the two intervals that immediately follow it iff there is no event that negates the property that occurs during the "odd interval" that immediately follows the even interval. To state that property *pr* will persist through the two intervals that immediately follow *i* as long as no member of {*ev1*, *ev2*, . . . , *evm*} occurs during the interval following *i*, we write:

PP1)
 (INEV I0
 (IF (HOLDS *pr i*)
 (IF (AND (NOT (OCCURS *ev1* (S *i*)))
 (NOT (OCCURS *ev2* (S *i*)))
 .
 .
 (NOT (OCCURS *evm* (S *i*))))
 (HOLDS *pr* (SS *i*))))))

where $\lceil (S\ i) \rceil$ refers to the interval that immediately follows *i*, and $\lceil (SS\ i) \rceil$ refers to the concatenation of the two intervals that successively follow *i*. For example, $\lceil (S\ I0) \rceil$ equals *I1* and $\lceil (SS\ I0) \rceil$ equals *I1-2*.

As an example, we might state that property $\lceil(\text{at } B \text{ loc1})\rceil$ will persist through interval $I1-2$ as long as no member of $\{\lceil(\text{move } B \text{ loc})\rceil \mid \text{loc is a location}\}$ occurs during $I1$:

```
PP2)
  (INEV I0
    (IF (HOLDS (at B loc1) I0)
      (IF (V?loc (NOT (OCCURS (move B ?loc) I1))))
      (HOLDS (at B loc1) I1-2))))
```

From *PP2* and a statement saying that it is not possible at $I0$ that $\lceil(\text{move } B \text{ loc})\rceil$ occurs during $I1$ for any location loc , we can derive that it is inevitable at $I0$ that if $\lceil(\text{at } B \text{ loc1})\rceil$ holds during $I0$, then it persists through $I1-2$.

Using a more general temporal theory, one must take into account the time when a property is negated with respect to the time when an event occurs. Is the property negated at the completion of the event, negated when it begins, or negated in the middle of execution? For example, suppose that it is inevitable at time I_p that if event $ev1$ occurs, then property pr is negated at the completion of execution. If we also assume that only $ev1$ can negate pr , we have the following relation:

```
PP3)
  (IF (MEETS ?i1 ?i2)
    (INEV I_p
      (IF (HOLDS pr ?i1)
        (IF (V?i (IF (ENDS-DURING ?i ?i2) (NOT (OCCURS ev1 ?i))
          (HOLDS pr ?i2))))
```

PP3 says that it is inevitable at I_p that if property pr holds during an interval $?i1$, then it persists throughout any following interval $?i2$ if event $ev1$ does not end during $?i2$. Vere [Vere 81], whose system handles actions with durations, makes the assumption that simple actions produce changes at their completion. He also allows actions that produce changes at the start of execution by modeling them as "consecutive actions" which are two simple actions that must be executed together.

Georgeff [Georgeff 86] develops a representation that can succinctly describe the set of actions that may negate some property. This formalism is a modification and extension of situation calculus to allow for simultaneous events and actions. An event in this theory is modeled by a *transition function*. This function is a mapping from situation to set of situations, instead of a mapping from situation to situation as in situation calculus. This modification allows the representation of simultaneous events. Let TR_{ev1} refer to the transition function associated with event $ev1$. Any situation belonging to the set $TR_{ev1}(s)$ is the result produced by the occurrence of $ev1$ along with any event that possibly can occur simultaneously with $ev1$ starting from situation s . If property pr is true in situation s , and $ev1$ does not affect property pr , it does not necessarily follow that pr holds in any situation belonging to $TR_{ev1}(s)$. Property pr might be negated by an event that occurs simultaneously with $ev1$

starting at s . This result is analogous to the result that we have just shown in our system.

An action in Georgeff's theory is more finely distinguished than an event. An action is associated with a transition function along with a *direct effects formula* for each n -ary predicate symbol in the language. Action $a1$'s transition function corresponds to the event that is brought about by performing $a1$. Action $a1$'s direct effects formulas are used for two purposes: i) to compute which actions can be performed simultaneously with $a1$, and ii) to describe the properties that are directly effected by $a1$, the ones that would necessarily be affected if $a1$ were to be executed in isolation or in conjunction with other actions. In this section, we are primarily concerned with the second function which gives rise to a schema for determining when a property persists.

For simplicity, we only describe the direct effects formulas for propositional symbols and will describe how Georgeff's system works if there are no other n -ary predicates, that is, there are no n -ary predicates for $n > 0$. Propositional symbols refer to propositions that either hold or do not hold at each situation. With relation to our system, they correspond to constant terms that denote properties. (note: n -ary predicates correspond to n -ary property functions in our system). For each proposition symbol P and action $a1$, we let $EFF_p(a1)$ refer to $a1$'s direct effects formula for P . The truth value for the formula $EFF_p(a1)$ may vary from situation to situation. If $EFF_p(a1)$ is true at situation s , then the performance of $a1$ directly affects property P and its negation.

Using the direct effect formulas, Georgeff encodes "a persistence law" that says¹:

GPL)

Under all possible circumstances, if P (not P) holds in situation s , and there does not exist an action $a1$ that occurs starting in s where $EFF_p(a1)$ is true in s , then P (not P) holds in the successor to s .

We can translate Georgeff's persistence law into a schema in our language. We make use of our simplified temporal theory where "even intervals" refer to static situations and events occur during "odd intervals". The following schema ranges over all property constants substituted for pr :

¹ To encode the law GPL, situation calculus is modified so the actual course of events can be described. Typically, situation calculus has been used to describe only the possible states that would arise if different sequences of events were to occur.

```

PP4)
  (IF (EVEN ?i)
    (AND (INEV I0
      (IF (AND (HOLDS pr ?i)
        (∀?ev (IF (HOLDS EFFpr (?ev) ?i)
          (NOT (OCC ?ev (S ?i))))))
        (HOLDS pr (SS ?i))))
      (INEV I0
        (IF (AND (NOT (HOLDS pr ?i))
          (∀?ev (IF (HOLDS EFFpr (?ev) ?i)
            (NOT (OCC ?ev (S ?i))))))
            (NOT (HOLDS pr (SS ?i)))))))

```

where $\lceil \text{(EVEN } i) \rceil$ is true iff i is an even interval, $\lceil (S \ i) \rceil$ refers to the interval that immediately follows i , and $\lceil (SS \ i) \rceil$ refers to the concatenation of the two intervals that successively follow i .

PP4 says that for any even interval i , it is inevitable at $I0$ that if property pr holds at i , then pr will persist through the two intervals that successively follow i as long as no event ev , that makes $\text{EFF}_{pr}(ev)$ true at i , occurs in the interval following i , similarly for pr not holding during i . Thus, we see that *GPL* is a schema for specifying statements of the form *PP1* where $\{ev \mid \text{EFF}_{pr}(ev) \text{ holds during } i\}$ represents the set of possible events, occurring during $\lceil (S \ i) \rceil$, that can negate the persistence of pr and its negation during interval $\lceil (SS \ i) \rceil$.

Persistence and Non-Deductive Schemas

The difficulties associated with the persistence problem arise when a description of a temporal scenario does not include statements, such as *PP1*, that specify all the possible events that can negate some property. For example, a description might specify that events $ev1$ and $ev2$ negate property pr , but does not indicate whether these are the only events that can negate pr . Using deductive rules alone, one cannot infer from this description that pr persists over interval I even if the description includes that $ev1$ and $ev2$ do not occur during a time that can negate pr during I . This inability to reach conclusions based on deductive inference has led to non-deductive schemas such as:

PER-ND)

if property pr holds immediately prior to interval I , and there is no indication that pr is not true throughout I , assume that pr holds throughout I

We call *PER-ND* a non-deductive schema because a conclusion can be reached from the lack of information. It is non-monotonic since later information can nullify an earlier conclusion. McDermott [McDermott 82] encodes a schema similar to *PER-ND* by extending his deductive logic of events and time with a non-monotonic operator. Very roughly, he equates "there is no indication that P is false" with "the

negation of P is not provable from the set of sentences (describing the scenario under consideration)". Dean [Dean 84] has developed a system that represents temporal scenarios that makes use of this interpretation of the persistence schema suggested by McDermott. To implement this mechanism he has to be able to efficiently compute when a set of properties are inconsistent when taken together. This is done by restricting the way properties can be logically related.

McDermott states that his persistence schema is intended to represent the relation "properties stay constant unless they are forced to change". This is not what his non-monotonic rule is capturing, however. (He does note that the introduction of a non-monotonic operator might cause some problems.) The statement "properties remain constant unless they are forced to change" describes the outside world and makes no mention of an agent's set of beliefs. On the other hand, by using a non-monotonic operator, he is encoding a relation between a set of sentences, presumably representing an agent's belief state, and a conclusion that may be reached from these sentences. He is encoding "Unless it is known (from some set of sentences) that a property is forced to change assume that this property does not change". Now, such an inference may be warranted, but only if we assume that the description of a temporal scenario adheres to the following principle:

DSCRPT-ASMPT)

A description of a temporal scenario explicitly mentions when properties change and, by omission, it is assumed that a property remains constant.

In summary, by proposing his non-deductive persistence assumption, McDermott is implicitly making an assumption about the kind of descriptions that are used to represent a temporal scenario (i.e., they adhere to principle *DSCRPT-ASMPT*). He does not take care in distinguishing the issues concerned with assumptions about the type of descriptions that are explicitly stored from the issues concerned with a logic that describes times and events. Georgeff [Georgeff 86] makes a similar point by saying, "Unfortunately, this problem [reasoning with incomplete descriptions] is often confused with the representation of actions, with the result that there is no clear model-theoretic semantics for the representation". It seems that the persistence problem has been taken to be the problem of computing and representing the persistence of properties when working with a description that meets *DSCRPT-ASMPT*.

Breaking the persistence assumption into two steps

Earlier in this section, we noted that the persistence problem can be analyzed by breaking it up into two steps: i) finding the set of possible events that can negate some property and ii) determining whether any of these events occur. Only the first part relates to the STRIPS assumption. The second part is not concerned with describing what events do not affect, which we take to be the general problem that the STRIPS assumption is addressing. The non-deductive persistence assumption

given by *PER-ND* does not divide the problem into these two parts. This may lead to unexpected "side-effects". We present an example to clarify this and then show that two different types of non-deductive rules may be used to replace the persistence assumption.

Consider the following example which we represent using our simplified temporal theory. Suppose that *pr* is true at *I0* and it is inevitable at *I0* that if event *ev* occurs during *I1* then *pr* will not persist throughout *I1-2*. This can be given by the following statements:

S1) (HOLDS *pr* *I0*)

S2) (INEV *I0* (IF (OCCURS *ev* *I1*) (NOT (HOLDS *pr* *I1-2*))))

If this is the entire description of a temporal scenario, then there is no evidence that contradicts "it is inevitable at *I0* that *pr* holds during *I1-2*". Thus, if we used a persistence assumption similar to McDermott's, we would get the following conclusion:

S3) (INEV *I0* (HOLDS *pr* *I1-2*))

Taking S2 and S3 together leads to $\lceil \text{INEV } I0 \text{ (NOT (OCCURS } ev1 \text{ } I1)) \rceil$, the conclusion that it is inevitable at *I0* that event *ev* does not occur during *I1*. Thus, a side-effect of applying a persistence assumption is that if there is an event that can negate some persistence and there is no information about this event's occurrences, then it follows that this event does not occur. This side-effect can lead to problems when reasoning about planning. If it is not explicitly asserted that the planning agent's actions are possible, then the conclusion may be reached that the agent cannot perform any actions that can negate some persistence.

An alternative to having a persistence assumption, such as *PER-ND*, is to have two assumptions that correspond to the two steps involved in the persistence problem. Thus, we might have the following non-deductive schemas:

NOT-NGT-ND)

If there is no evidence that event *ev* occurring during *I1* negates the persistence of property *pr* during *I2*, assume that this occurrence does not negate this persistence

NOT-OC-ND)

If there is no evidence that event *ev* occurs during *I*, assume that *ev* does not occur during *I*.

Alternatively, we can weaken *NOT-OC-ND* so that it only applies to events not performed or caused by the planning agent so as not to lead to problems when reasoning about planning. One might even choose to only use one of these two assumptions, or weaken them in different ways.

We intend to use *NOT-NGT-ND* so that we can go from a description saying that the persistence of *pr* during *I1-2* is negated if *ev1* or *ev2* occur during *I1*:

```
NDS1)
  (INEV I0
    (IF (HOLDS pr I0)
      (IF (OR (OCCURS ev1 I1)
              (OCCURS ev2 I1))
        (NOT (HOLDS pr I1-2))))))
```

to a description saying that the persistence of *pr* during *I1-2* is negated iff *ev1* or *ev2* occur during *I1*:

```
NDS2)
  (INEV I0
    (IF (HOLDS pr I0)
      (IFF (OR (OCCURS ev1 I1)
                (OCCURS ev2 I1))
        (NOT (HOLDS pr I1-2))))))
```

In this work, we have not tried to formalize *NOT-NGT-ND* in some non-monotonic logic thereby formalizing an inference from *NDS1* to *NDS2*. This problem can be factored out from the problems that we are concentrating on here, namely characterizing a deductive logic of action and time.

If we had a procedure that could go from *NDS1* to *NDS2*, we could start with a description such as *NDS1*, preprocess it to form *NDS2*, and then use deductive reasoning (in our logic) to form a plan. We make use of such "preprocessed" descriptions in the next section where we describe the conditions under which the planning agent can cause some property to persist.² As we will see, in order to conclude that the agent can cause some property to persist under all conditions, we must know that there are no possible external events that can negate the property.

² Treating non-deductive reasoning as a preprocessing operation seems more in the flavor of circumscription [McCarthy 80] than default logic [Reiter 80] or McDermott's and Doyle's logic [McDermott&Doyle 80] where deductive and non-deductive reasoning are intermixed. We must note, however, that we do not necessarily favor one approach over the other. We are describing non-deductive inference as a preprocessing stage to the deductive part for analysis purpose. This enables us to look at the role played by deductive reasoning. When constructing an inference mechanism, however, we do not necessarily want to do all the non-deductive reasoning first.

5.6. Maintaining a Property

If the agent can execute some plan instance to make property pr persist, we say that pr can be **maintained**. In an environment in which all changes are caused by the planning agent, any property can be maintained. In a planning environment in which external events can also produce changes, the treatment of maintenance is more complex. There are some properties that the planning agent can maintain, there are properties that the planning agent cannot maintain, and there are properties that the planning agent can conditionally maintain.

If it is inevitable at $I0$ that pr can be maintained throughout interval $I1-2$, then the following holds ^{1,2}:

MTN)
 (∃?pi (INEV I0
 (IF (HOLDS pr I0)
 (IFTRIED ?pi (HOLDS pr I1-2))))))

A case where MTN does not hold is when the agent cannot prevent a property from being negated. Consider the following example. Assume that i) a mainframe is operational during $I0$, ii) it is possible that this machine will not be operational during $I1-2$, and iii) it is not in the agent's control to prevent the mainframe from going down. This can be represented in our language by:

MF1)
 (AND (HOLDS (mf-status running) I0)
 (POS I0 (NOT (HOLDS (mf-status running) I1-2)))
 (INEV I0
 (IF (NOT (HOLDS (mf-status running) I1-2))
 (IFTRIED ?pi (NOT (HOLDS (mf-status running) I1-2))))))

where the term \lceil (mf-status running) \rceil refers to the property "the mainframe is operational".

In appendix G, we give a proof that shows that MTN , substituting \lceil (mf-status running) \rceil for pr , is inconsistent with $MF1$. We must also point out that the agent may not be able to maintain a property although it can bring about its negation. For

¹ We are glossing over the fact that its is not entirely correct to say that pr is maintained through interval $I1-2$ if it is inevitable at $I0$ that pr holds during $I1-2$. In this case, no matter what the agent does, pr will hold during $I1-2$.

² There is weaker sense of "it is inevitable at $I0$ that pr can be maintained throughout interval $I1-2$ " corresponding the case where the \exists symbol is on the inside of INEV, instead of on the outside. This corresponds to the case where in different possible branches, a different plan instances may be required to keep property pr true. A similar distinction was discussed in section 5.1

example, the agent may be able to deliberately crash the machine but cannot prevent the machine from going down.

Let us now examine a case where it is entirely in the agent's control to maintain some property. Suppose that if the agent's car is parked on Wilson Boulevard during time I_0 , it will remain parked through interval I_1-2 if the car is not moved to another location during interval I_1 . We can express this in our language as follows:

```
CP1)
  (INEV I0
    (IF (HOLDS (car-parked Wilson) I0)
      (IF (V?loc (NOT (OCCUR (car-moving-to ?loc) I1)))
        (HOLDS (car-parked Wilson) I1-2))))
```

If we assume that only the planning agent can cause a *car moving* event (and the agent is not forced to perform a *car moving* event) then the agent can choose not to perform a *car moving* event. Consequently, there will be a plan instance that corresponds to the non-occurrence of any *car moving* event during interval I_1 . We will let *not-move-car@i1* refer to this plan instance.³ The relation between *not-move-car@I1* and $\lceil(\text{car-moving-to loc})\rceil$ is given as follows:

```
CP2)
  (INEV I0
    (IF (OCC not-move-car@i1)
      (V?loc (NOT (OCCURS (car-moving-to ?loc) I1))))
```

Using statements *CP1* and *CP2*, we can derive:

```
CP3)
  (INEV I0
    (IF (HOLDS (car-parked Wilson) I0)
      (IF (OCC not-move-car@i1)
        (HOLDS (car-parked Wilson) I1-2))))
```

Statement *CP3* says that it is inevitable at I_0 that if the car is parked on Wilson Boulevard during time I_0 , it will remain parked through interval I_1-2 if the plan instance *not-move-car@i1* occurs. In a typical situation, a plan instance corresponding to the non-occurrence of a set of events, such as *not-move-car@I1*, can be done at will. That is, there are no executability conditions associated with a non-

³ We can also talk about a plan instance that corresponds to the non-occurrence of a *car moving* event to a particular location during interval I_1 . The plan instance *not-move-car@I1* equals the composition of all these plan instances

occurrence plan instance. Thus, we assume that it is inevitable at *I0* that *not-move-car@I1* is executable:

CP4)
(INEV I0 (EXECUTABLE not-move-car@I1))

Taking CP3 and CP4 together, we can derive:

CP5)
(INEV I0
(IF (HOLDS (car parked Wilson) I0)
(IFTRIED not-move-car@I1
(HOLDS (car parked Wilson) I1-2))))))

Statement CP5 says that under all circumstances possible at *I0*, the property \lceil (car parked Wilson) \rceil can be maintained by performing *not-move-car@I1*.

In the above scenario, the property \lceil (car parked Wilson) \rceil may also be maintained through *I1-2* as a "side effect" of performing a plan instance that cannot occur simultaneously with any *car moving* event. For example, the performance of a plan instance corresponding to working in the library during interval *I1* precludes a *car moving* event from occurring:

SL1)
(INEV I0
(IF (OCC stay-in-library@I1)
($\forall ?loc$ (NOT (OCCURS (car-moving-to ?loc) I1))))))

Using the same line of reasoning as we did for *not-move-car@I1*, we can prove that \lceil (car parked Wilson) \rceil may be maintained through *I1-2* by performing *stay-in-library@I1*.

The next case to consider is where both the agent and the external environment can cause the car to be moved from Wilson Boulevard. Assume that there are two ways that the car may be moved from Wilson Boulevard. One way is if the agent moves the car and the other way is if the car is towed away. Thus, we have the following:

CT1)
(INEV I0
(IF (HOLDS (car-parked Wilson) I0)
(IF (AND (OCC not-move-car @I1)
(NOT (OCCURS (car towed) I1)))
(HOLDS (car-parked Wilson) I1-2))))))

CT1 says that it is inevitable at *I0* that if the car is parked on Wilson during *I0*, then it will remain on Wilson during *I1-2*, if the agent does not move the car during *I1* and the car is not towed during *I1*. We would like to show that \lceil (car-parked Wilson) \rceil can

be maintained through *I1-2* if it is not possible at *I0* that the car is towed during *I1*. That is, we would like to show the following:

```
CT2)
  (IF (NOT (POS I0 (OCCURS (car towed) I1)))
    (INEV I0
      (IF (HOLDS (car-parked Wilson) I0)
        (IFTRIED not-move-car@I1
          (HOLDS (car parked Wilson) I1-2))))))
```

CT2, however, does not logically follow from *CT1* because *CT1* does not rule out the case where the execution of *not-move-car@I1* causes $\lceil(\text{car towed})\rceil$ to occur during *I1*; statement *CT1* does not describe the relation between $\lceil(\text{car towed})\rceil$ and *not-move-car@I1*. For a realistic example, we would want to specify that *not-move-car@I1* has no affect on whether $\lceil(\text{car towed})\rceil$ occurs or not. This can be represented by:

```
CT3)
  (INEV I0
    (AND (IF (OCCURS (car towed) I1)
      (IFTRIED not-move-car@I1 (OCCURS (car towed) I1)))
      (IF (NOT (OCCURS (car towed) I1))
        (IFTRIED not-move-car@I1
          (NOT (OCCURS (car towed) I1))))))
```

Maintenance and Phantom Nodes

Phantom nodes, which are used in non-linear planning systems [Sacerdoti 77], relate to maintenance plan instances. In a non-linear planner, if an action *a*, that is introduced into the plan, has a precondition that holds at an earlier time and there is no intermediate action that negates this precondition, a phantom node is created. The presence of a phantom node indicates that it is not necessary (at least at this stage) to explicitly introduce another action to achieve this precondition. If the system finds another action *a2* in the plan whose effects negates *a1*'s precondition, the system tries to order this action to follow *a1*. If this ordering is not possible, the system must introduce a third action following *a2* and before *a1* that restores *a1*'s precondition, or remove either *a1* or *a2* from the plan. This strategy can be seen as an implementation of the STRIPS assumption.

In our system, plan instances that maintain some property take the place of phantom nodes, although there are some important differences. In our system, not every property can be maintained, while in a non-linear planning system, a phantom node may be created for any property. For example, if property *pr* is out of the agent's control, then *pr* cannot be maintained. In a state-based system, if property *pr* is asserted to hold in the initial state, then a phantom node may be created for *pr* appearing in any action's precondition list. This is true since we are assuming that

pr is out of the agent's control, and consequently there are no intermediate actions that can negate pr . As we mentioned in section 1.2, this leads to problems if we do not assume that all changes in the world are caused by the planning agent. In a system, such as DEVISER [Vere 81] that uses phantom nodes and represents external events, one must be careful in specifying the planning environment. If a property that the agent cannot affect is asserted to be initially true, then the planning environment must also include all external events that affect the value of this property. For example, if it is asserted that "the bank is open" holds in the initial state, it is necessary to include the external event that negates this property at the time when the bank is closing. If this is not done, we would get the spurious result that the precondition "the bank is open" is always satisfied.

One might suggest that a quick fix to a state-based system is to characterize properties as being either in the agent's control or not in the agent's control and to only create phantom nodes for properties in the agent's control. While this will handle some cases, there are other situations that cannot be treated. For instance, properties that are affected by both the agent and the external world cannot be handled by this simple scheme. Consider the example we gave in the last section where a parked car is moved iff the agent moves the car or if the car is towed. In this situation, we come up with the plan instance "do not move the car during time I " that conditionally maintains the property "the car is parked" during I . It is not clear how a state-based system can be modified to handle a conditional maintenance without moving to a system that treats phantom nodes just like actions, as we do in our system. By treating phantom nodes as maintenance actions, we could encode the above scenario by representing an action "maintain 'the car is parked'" whose executability conditions are that "the car is parked" holds just prior to execution and "the car is not being towed" holds during execution.

Maintenance and interference

The important feature of a phantom node is that it cannot be used to link a property true at some state to a later state if there is an intermediate action that negates the property. Analogously, in our system, a property cannot be maintained over some interval I if a plan instance occurs during I that negates this property. Consequently, if plan instance $pi2$ maintains the properties during I that are needed for plan instance pi to be executable, any earlier plan instance that would negate pi 's executability conditions would also conflict with $pi2$. Thus, the conflict where an earlier plan instance ruins a later one's executability conditions can be detected by only considering concurrent interactions. In the following chapter, we present an algorithm that exploits this relation.

In a system such as Wilkins' [Wilkins 83] that uses phantom nodes but allows concurrent actions, there is one mechanism for finding conflicts where one action's effects ruins another's preconditions and another for finding conflicts between concurrent actions, this being the resource mechanism. In our system, since both types of conflicts are detected as concurrent interactions, only one type of mechanism

is needed. In a paper describing resource management in planning, Bell suggests a similar idea in [Bell 85] where he says that the interaction between an earlier action's effects and a later one's preconditions can be thought of as "logical" resource conflicts. Thus, all conflicts can be detected by making sure that concurrent actions do not use up the available resources.

To help clarify the points made above, we present a simple "blocks world" example. We will refer to three intervals: I_p , which meets I_1 , which meets I_2 . Interval I_p refers to planning time. Consider the term $\lceil(\text{grasp blk1})@I_2\rceil$ which refers to the plan instance where the agent grasps block $blk1$ during interval I_2 . This plan instance is executable if the property $\lceil(\text{clear blk1})\rceil$ holds during interval I_1 . We assume that $\lceil(\text{clear blk1})\rceil$ holds at planning time I_p . Thus, the agent can successfully perform $\lceil(\text{grasp blk1})@I_2\rceil$ if it can maintain $\lceil(\text{clear blk1})\rceil$ through interval I_1 . We assume that this can be done and will let $\lceil(\text{keep-clear blk1})@I_1\rceil$ refer to the plan instance that maintains property $\lceil(\text{clear blk1})\rceil$ through interval I_1 . We also assume that it is inevitable at I_p that $\lceil(\text{keep-clear blk1})@I_1\rceil$ is executable. Consequently, it is inevitable at I_p that the composition of $\lceil(\text{keep-clear blk1})@I_1\rceil$ and $\lceil(\text{grasp blk1})@I_2\rceil$ is executable. This result follows from the theorem that is given below (and which is derivable from *SEQ-TH2* and *SEQ-TH3*, given in section 5.3):

SEQ-TH4)

```
(IF (AND (PRIOR  $I_p$  (TIME-OF  $\pi_1$ ))
          (PRIOR (TIME-OF  $\pi_1$ ) (TIME-OF  $\pi_2$ )))
    (IFF (AND (INEV  $I_p$  (EXECUTABLE  $\pi_1$ ))
              (INEV  $I_p$  (IF EC2 (EXECUTABLE  $\pi_2$ )))
              (INEV  $I_p$  (IF (OCC  $\pi_1$ ) EC2)))
          (INEV  $I_p$  (EXECUTABLE (COMP  $\pi_1$   $\pi_2$ ))))))
```

SEQ-TH4 says that if I_p is prior to π_1 which is prior to π_2 , then it is inevitable at I_p that the composition of π_1 and π_2 is executable iff i) it is inevitable at I_p that π_1 is executable and ii) it is inevitable at I_p that π_1 brings about π_2 's executability conditions. *SEQ-TH4* applies to our example by substituting $\lceil(\text{keep-clear blk1})@I_1\rceil$ for π_1 , $\lceil(\text{grasp blk1})@I_2\rceil$ for π_2 , and $\lceil(\text{HOLDS}(\text{clear blk1}) I_1)\rceil$ for *EC2*.

Let us now consider a third plan instance that is prior to $\lceil(\text{grasp blk1})@I_2\rceil$, one that might ruin $\lceil(\text{grasp blk1})@I_2\rceil$'s executability conditions. We assume that this plan instance's time of occurrence is I_1 and we will use $ev@I_1$ to denote it. Now, if $ev@I_1$ is introduced into the plan along with $\lceil(\text{keep-clear blk1})@I_1\rceil$ and $\lceil(\text{grasp blk1})@I_2\rceil$, we must make sure that it is inevitable at I_p that all three plan instances are executable together (We are ignoring the case where a fourth plan instance is introduced that enables the three plan instance to be executed together). Applying *SEQ-TH4*, substituting $\lceil(\text{COMP } ev@I_1 (\text{keep-clear blk1})@I_1)\rceil$ for π_1 , $\lceil(\text{grasp blk1})@I_2\rceil$ for π_2 , and $\lceil(\text{HOLDS}(\text{clear blk1}) I_1)\rceil$ for *EC2*, leads to:@

Chapter 6

A Planning Algorithm

6.1. Overview

In this chapter, we present a planning algorithm based on our formal logic. The input to the planning algorithm consists of:

- i) the goal conditions, which are given by a conjunction of interval logic statements which we will designate by G
- ii) a specification of the planning environment
- iii) the executability conditions for each simple plan instance
- iv) the effects produced by each simple plan instance
- v) the **non-interference conditions**, which relate pairs of simple plan instances that directly interfere with each other (which we will describe in section 6.2)

The specifications in ii) - v) induce constraints on the possible set of world models. These constraints can be captured by a set of sentences in our normal language, which we will designate by S . We will use I_p to denote the time of planning.

The algorithm searches for a set of simple plan instances, in the future of planning time I_p , such that their composition pi meets the relation: it is inevitable at I_p that pi is executable and inevitable at I_p that if pi occurs then G is true in any model where all the sentences in S are satisfied. Using our formal notation, this relation can be given by:

$$\begin{aligned} \text{SOL-PI)} \\ S \models (\text{AND } (\text{INEV } I_p (\text{EXECUTABLE } pi)) \\ (\text{INEV } I_p (\text{IF } (\text{OCC } pi) G))) \end{aligned}$$

Relation *SOL-PI* contains two of three necessary and sufficient conditions that we gave in the chapter 5 introduction for concluding that plan instance pi solves goal G with respect to S . We will not consider the third condition " $S \models (\text{POS } I_p (\text{NOT } G))$ " which is used to insure that the goal condition does not inevitably hold at I_p in all the models that satisfy S . Since this condition does not mention the argument pi , it can be checked separately, irrespective of the plan instance under consideration.

The procedure that we present is a non-linear backward chaining algorithm loosely based on the algorithm described by Allen and Koomen [Allen&Koomen 83b] which in turn is based on non-linear planning systems such as NOAH [Sacerdoti 77] and NONLIN [Tate 77]. At the beginning of each cycle, there is a conjunction of conditions to be achieved, which we call the **causal gap**, adopting the terminology from Allen and Koomen. Initially, the causal gap is equated with the goal conditions. During each cycle, an operation is chosen that removes one or more of the conjuncts from the causal gap, although new conditions may be added. A solution has been found if the last conjunct is removed from the causal gap and no

new ones added. We call the algorithm non-linear because at times the algorithm can be seen as solving two conjuncts separately and then checking whether the two plan instances that solve the two goals can be executed together.

There are two types of operations that can be performed to reduce the casual gap. One type of operation involves removing a conjunct from the causal gap that inevitably holds in the planning environment. The other type of operation involves the introduction of a simple plan instance "into the plan" to remove one or more conjuncts in the causal gap. By introducing plan instance pi , we can remove any conjunct that inevitably holds in the planning environment augmented by pi 's effects. We shortly give a precise characterization of "the planning environment augmented by pi 's effects". The introduction of plan instance pi also results in conditions being added to the causal gap; pi 's executability conditions are added along with the non-interference conditions relating pi and any plan instance that has previously been entered into the plan. Roughly, If the non-interference condition between pi and $pi2$ hold, then under all conditions possible at planning time, if both pi and $pi2$ are executable separately, then their composition is executable. In section 6.2, we discuss non-interference conditions in some detail.

At each cycle, there may be a number of different operations that can remove one or more conjuncts from the causal gap if applied. In this work, we will not analyze heuristics that may be used to choose among a set of applicable operations or analyze backtracking schemes that may be used when a causal gap is reached that cannot be solved, an example being where both P and $\neg(P)$ are in the causal gap. Instead, we describe a non-deterministic algorithm by just describing the set of planning operators, specifying when they are applicable and what their effects are. Each non-deterministic behavior corresponds to a finite sequence of operators, all of whose members are applicable when reached within the sequence. A sequence yields a solution if the casual gap is empty after all the operators in the sequence are applied. In this case, the solution consists of the composition of the simple plan instances that have been introduced by one of the operators in the sequence. Substituting this composition for pi in $SOL-PI$ makes $SOL-PI$ true.

We justify the algorithm by showing that the application of a sequence of operators can be mapped to a transformation from the initial problem state ST_0 through a succession of problem states ST_1, \dots, ST_f , where ST_i is true if ST_{i+1} is true for every i in the sequence (except $i = f$). The initial problem state is given by:

ST_0)

$$S \models (\exists pi (AND (PRIOR Ip (TIME-OF ?pi)) \\ (INEV Ip (EXECUTABLE ?pi)) \\ (INEV Ip (IF (OCC ?pi) G))))$$

Problem state ST_0 is true iff there exists a plan instance in the future of planning time Ip that meets the two conditions in $SOL-PI$. A solution sequence corresponds to a sequence of transformations from ST_0 to a problem state that is logically true.

One can think of the algorithm as a limited proof procedure that is sound with respect to the semantic theory. In this chapter, we will be using the proof theory that we developed in chapter 4 as a tool for showing that the algorithm is sound with respect to the semantics. We would say that our algorithm was complete, if it also had the following property: anytime that ST_0 is true, there exists a sequence of operators yielding a solution. One could develop a complete algorithm without developing a complete proof theory for the entire language. This is because the algorithm only involves the proof of a particular form of sentence (i.e. the sentence on the right of " \models " in ST_0) from a set of sentences describing the planning environment and action specifications, these sentences also having limited form.

Our algorithm differs from previous planning systems in the method used to handle action interactions and the use of maintenance plan instances, instead of using ghost nodes (see section 5.5). The interaction of two or more plan instances is computed by only considering the interaction of two plan instances that overlap in time. As we discussed in section 5.5, the conflict where an earlier plan instance ruins a later one's executability conditions can be detected by only considering concurrent interactions. In section 6.4, we present an example to show that the interaction of three or more plan instances can be detected by only looking at binary interactions.

To emphasize the novel features of our algorithm, we make some simplifications. For one, we only consider the introduction of grounded plan instance terms. Typically, planning systems such as NOAH [Sacerdoti 77], DEVISER [Vere 81], and SIPE [Wilkins 83] allow the introduction of a set of actions described by a function term that has one or more arguments that are constrained to meet some relations. As other constrained actions are introduced into the plan, new constraints may be imposed on the actions already in the plan.¹ The advantage to this approach is that, in some cases, the algorithm can decide if a whole class of actions conflict with other actions in the plan without trying out each instance of the class separately. Furthermore, the algorithm can delay deciding on the specific objects to be used as part of some action before knowing what other actions might use the same objects.

Another issue that we do not consider is hierarchical planning such as described by Sacerdoti [Sacerdoti 77]. This refers to planning initially at an abstract level, ignoring detail, and then successively planning at more and more detailed levels. Tenenberg [Tenenberg 86] describes a formal representation that may be used to support hierarchical planning. A problem may be characterized at different levels of detail by a set of theories which form an *abstraction hierarchy*. If there is a solution to a planning problem in some theory $t1$, then there is a corresponding solution in any theory that is an abstraction of $t1$. This is formalized using state-based actions.

¹ These systems deal with actions, instead of plan instances, that are successively ordered as planning goes on. In our framework, we can think of the ordering process as adding constraints to the "time of occurrence arguments" associated with the plan instances in the plan.

Future work may involve adapting this approach to be used with the representation of actions developed here.

In the next section we give a succinct description of the planning algorithm. We then discuss some simplifications made beyond what can be represented in our framework, as discussed in chapter 5. In the following section, we prove that the algorithm is sound with respect to our semantics, and in the last section, we present some simple examples.

6.2. Specification of the Planning Algorithm

In this section, we present the planning algorithm and then discuss the inputs, which characterize the planning problem. The algorithm is specified by describing i) the inputs that are required, ii) the two state variables that are transformed by the planning operators, and iii) for both types of planning operators, the states in which they are applicable and the transformations produced by their application. We then characterize applicable operator sequences and the result produced by these sequences. Each applicable sequence corresponds to a non-deterministic behavior of the planning algorithm. A solution corresponds to a sequence that when applied to the initial state, removes all the conjuncts from the causal gap.

The inputs to the planning algorithm are given by:

G	a conjunction of interval logic statements describing the goal conditions
PE	a set of interval logic statements describing the conditions that are inevitable at planning time

For each simple plan instance pi

EC(pi)	an interval logic statement describing pi 's executability conditions
EFF(pi)	an interval logic statement describing pi 's effects

For each of pair of plan instances $pi1$ and $pi2$ that overlap in time

NI($pi1, pi2$)	an interval logic statement describing the conditions under which $pi1$ and $pi2$ do not interfere which we will refer to as their non-interference conditions
------------------	--

notes: NI is symmetric in its arguments, so only NI($pi1, pi2$) or NI($pi2, pi1$) need be given. If $pi1$ and $pi2$ interfere under all conditions, NI($pi1, pi2$) is set to a false statement and if $pi1$ and $pi2$ do not interfere under any conditions NI($pi1, pi2$) is set to a tautology.

A state is an ordered pair having the form $\langle \text{INPLAN}_i, \text{CG}_i \rangle$ where:

INPLAN_i is the set of plan instances already entered at cycle i

CG_i is the conjunction of interval logic statements making up the causal gap at cycle i .

In the initial state, the plan is empty and the causal gap is equated with the goal conditions:

$$\text{INPLAN}_0 = \emptyset$$

$$\text{CG}_0 = G$$

The planning operators are given as partial functions from state to state. An operator's domain specifies the states in which it is applicable. A function applied to an applicable state captures the transformation produced by applying the operator. There are two types of planning operators, which we refer to by REMOVE_C and $\text{INTRO}_{pi,C}$, whose descriptions are given in figure 6.2-1.

A sequence of operators is applicable in state $\langle \text{INPLAN}_i, \text{CG}_i \rangle$ if each operator is applicable when reached in the sequence. More precisely, this can be given by the following recursive definition:

A sequence op_1 consisting of one operator is applicable in state $\langle \text{INPLAN}_i, \text{CG}_i \rangle$ if op_1 is applicable in this state

A sequence of two or more operators $op_1 \ op_2 \ \dots \ op_n$ is applicable in state $\langle \text{INPLAN}_i, \text{CG}_i \rangle$ if op_1 is applicable in this state and sequence $op_2 \ \dots \ op_n$ is applicable in state $op_1(\langle \text{INPLAN}_i, \text{CG}_i \rangle)$.

The result of applying a sequence of operators that are applicable in state $\langle \text{INPLAN}_i, \text{CG}_i \rangle$ is the final state reached after applying all operators in order starting from $\langle \text{INPLAN}_i, \text{CG}_i \rangle$.

A solution corresponds to sequence of operators that are applicable in the initial state $\langle \text{INPLAN}_0, \text{CG}_0 \rangle$ and whose result produces the state $\langle \text{INPLAN}_n, \text{CG}_n \rangle$ where CG_n is a tautology, which means that the causal gap is empty. The solution is given by the composition of simple plan instances in INPLAN_i .

We now examine some of the simplifications that we have made beyond what can be represented in our formalism as described in chapter 5. We also examine some relations to other non-linear planners.

The planning environment

The planning environment is given by PE which specifies conditions that are inevitably true at planning time. Thus, saying IL belongs to PE is tantamount to

$REMOVE_C$ removes an inevitably true conjunct C from the causal gap

Applicable in state $\langle INPLAN_i, CG_i \rangle$ iff

C is a conjunct in CG_i and $PE \vdash_{IL} C$

where " $PE \vdash_{IL} C$ " is true iff C is derivable from PE only using inference rules and axioms in the non-modal fragment

note: if " $PE \vdash_{IL} C$ " is true then we say that " C inevitably holds in the planning environment"

Transformation produced:

For any applicable state $\langle INPLAN_i, CG_i \rangle$,

$REMOVE_C(\langle INPLAN_i, CG_i \rangle) =_{def} \langle INPLAN_i, \lceil (CG_i - C) \rceil \rangle$

where $\lceil (CG_i - C) \rceil$ refers to the conjunction of all the conjuncts in CG_i with the exception of C ; if C is the only conjunct in CG_i , $\lceil (CG_i - C) \rceil$ stands for a tautology

$INTRO_{p,C}$ introduces plan instance pi into the plan in order to remove conjunct C

Applicable in state $\langle INPLAN_i, CG_i \rangle$ iff

C is a conjunct in CG_i and $PEU\{EFF(pi)\} \vdash_{IL} C$

note: if " $PEU\{E\} \vdash_{IL} C$ " is true we say that " C inevitably holds in the planning environment augmented by E "

Transformation produced:

For any applicable state $\langle INPLAN_i, CG_i \rangle$,

$INTRO_{p,C}(\langle INPLAN_i, CG_i \rangle) =_{def} \langle INPLAN_i \cup \{pi\}, CG_{i+1} \rangle$

where CG_{i+1} stands for:

$\lceil (AND (CG_i - C) EC(pi) NI(pi, pi_1) NI(pi, pi_2) \dots NI(pi, pi_n)) \rceil$

for all pi_j such that $pi_j \in INPLAN_i$ and pi_j overlaps (in time) with pi

note: The conjunction CG_{i+1} consists of the conjuncts in CG_i with the exception of C , pi 's executability conditions, and the non-interference conditions relating pi and all plan instances that have previously been entered into the plan that have overlapping times with pi

Figure 6.2-1

saying that $\lceil(INEV\ Ip\ IL)\rceil$ belongs to S , the set of sentences that describe the planning problem discussed at the beginning of this chapter.

We do not put any restrictions on the interval logic statements belonging to PE . They may include statements describing the temporal relation between interval constants and statements describing the times when events occur and properties hold, both relative and absolute. In Vere's system [Vere 81], one can only use absolute dates to specify when some external event occurs. In our system, absolute dates roughly correspond to assertions about conditions that hold over some interval constant. We can also describe relative relations such as event $ev2$ occurs after some occurrence of $ev1$. This would be given by:

$$(\exists ?i1 ?i2 (AND (PRIOR ?i1 ?i2) (OCCURS\ ev1 ?i1) (OCCURS\ ev2 ?i2)))$$

Disjunctive statements may also belong to PE . In a system such as Vere's, that implicitly uses the STRIPS assumption, disjunctions cannot be used. In section 6.4, we present a simple example showing that disjunctions, which are harmless in our system would lead to problems if the STRIPS assumption were used. Moreover, we do not need to assume that the planning environment completely describes all the external events that can affect any property that is asserted to be initially true. In section 1.2, we showed that this assumption is necessary in Vere's system since he uses the STRIPS assumption.

The simplification that we have made beyond what can be represented in our language is the omission of conditions that are possible but not inevitable from the planning environment. For our purposes here, it does not matter whether conditions that are possible but not inevitable are omitted or whether they are asserted to be possible. This is because we have only characterized when a sequence of operators yields a solution plan instance, i.e. a plan instance pi that meets the two conditions in $SOL-PI$, $\lceil(INEV\ Ip\ (EXECUTABLE\ pi))\rceil$ and $\lceil(INEV\ Ip\ (IF\ (OCC\ pi)\ G))\rceil$. These two statements refer to conditions that are inevitably true. One can only prove the truthhood of inevitable conditions from other inevitable conditions. Possible conditions would only be needed if we wanted to prove the falsehood of an inevitable statement, such as proving that there is no plan instance that meets the two conditions in $SOL-PI$. Thus, conditions that are possibly true may be needed if we wanted to prove that a goal is unsolvable or if we wanted to investigate search techniques for rejecting, in the midst of the planning process, a set of simple plan instance that cannot be part of any solution.

In this work, we have not considered these issues. This is typical of current planning systems. They have characterized when a solution is found, but have not precisely characterized when a goal is unsolvable. One might suggest that a planning system can conclude that the goal is unsolvable if the set of planning operators supplied cannot be used to find a solution. If this technique is adopted, however, one cannot distinguish between the case where the system knows that a solution does not exist from the case where the system knows that it cannot solve the problem, but does not know whether a solution exists.

The Goal Conditions and the Causal Gap

The goal conditions and the causal gap are taken to be conjunctions of interval logic statements. Each conjunct may be a complex formula, but is treated as a single unit. That is, each conjunct is either removed as a whole or not removed during the planning cycles. For example, if there are two formulas that mention the same existential variable, they must be treated as a single unit. This restriction precludes the use of a search strategy that is employed by the non-linear planners such as NOAH [Sacerdoti 77]. In these systems, two formulas that share an existential variable may be solved separately. When the solution for one of the formulas involves binding a constant or imposing constraints on one of the shared variables, the other solution must be checked to see if the bound constant or constraints imposed are compatible.

The criteria that we use to check if a condition can be removed from the causal gap is more general than the test typically employed by the non-linear planners. Using the operation REMOVE_C , a conjunct C can be removed if " $PE \vdash_{IL} C$ " where \vdash_{IL} refers to derivability only using the interval logic (i.e non-modal) axioms and inference rules. Similarly, conjunct C can be removed using $\text{INTRO}_{pi,C}$ if " $PE \cup \{\text{EFF}(pi)\} \vdash_{IL} C$ " holds. In a non-linear planner, an action can be used to remove a condition C only if the action's effects syntactically entail the condition. This is a special case of our relation; if " $\text{EFF}(pi) \vdash_{IL} C$ " is true then " $PE \cup \{\text{EFF}(pi)\} \vdash_{IL} C$ " is true. The relation between C and $\text{EFF}(pi)$ might, however, depend on the sentences that are in PE . For example, suppose that C stands for " $(\text{HOLDS } pr \ I1)$ " and $\text{EFF}(pi)$ stands for " $(\text{HOLDS } pr \ I2)$ ". We can only establish that " $PE \cup \{\text{EFF}(pi)\} \vdash_{IL} C$ " is true if " $(\text{IN } I1 \ I2)$ ", meaning that interval $I1$ is contained in or equals $I2$, is derivable from PE . Other type of relations between C and $\text{EFF}(pi)$ are associated with indirect effects (see section 5.4). For example, suppose that " $(\text{INEV } Ip \ (\text{IF } A \ B))$ " is in PE . In this case, one might say that a plan instance pi whose effects include A , indirectly brings about B , and thus pi may be introduced to remove B if it is in the causal gap.

For simplicity, we have not provided for the case where a conjunct can only be removed by using the effects of two plan instances taken together and not by either of the plan instances taken alone. This condition could easily be provided for by modifying the relation " $PE \cup \{\text{EFF}(pi)\} \vdash_{IL} C$ " so that the left hand side also mentions the occurrences of all the plan instances that have previously been entered into the plan. Another simple extension that we can make to the relation " $PE \cup \{\text{EFF}(pi)\} \vdash_{IL} C$ " is to add " $(\text{OCC } pi)$ " on the left hand side. This would allow us to take advantage of *behavioral constraints* (see [Lansky 85] and sections 5.2 and 5.3) where one specifies the executability conditions for some plan instance $pi2$ using a form such as " $(\text{OCC } pi)$ " to indicate the the occurrence of pi enables $pi2$. This type of relation allows one to bypass the use of properties that must be used in state-based systems to indirectly relate two actions where one enables the other.

Action Specifications

The relation between a plan instance and its executability conditions and between a plan instance and its effects were given in sections 5.2 and 5.4, respectively. Specifying $EC(pi)$ as pi 's executability conditions implies that the following sentence holds in S , the set of sentences that describe the planning problem:

$(INEV Ip (IF EC(pi) (EXECUTABLE pi)))$

Specifying $EFF(pi)$ as pi 's effects implies that the following sentence holds in S :

$(INEV Ip (IF (OCC pi) EFF(pi)))$

We place no restrictions on the form of $EC(pi)$ (other than it is an interval logic statement). A disjunction in $EC(pi)$ can be used to specify that there are different ways to enable plan instance pi . For example, if we employed behavioral constraints, as we just discussed in the last sub-section, we could set $EC(pi)$ to $\lceil (OR (OCC pi1) (OCC pi2)) \rceil$ to specify that both $pi1$ and $pi2$ enable pi .

We also do not place any restrictions on the form of $EFF(pi)$. Thus, we allow disjunctive effects. If we were using the STRIPS assumption, disjunctive effects, just like disjunctions in the planning environment, would lead to problems. Moreover, omitting an effect would lead to problems since STRIPS works under the assumption that unless otherwise noted an action does not affect a property. This restriction does not have to be made in our system. Consequently, $EC(pi)$ may only include some of pi 's effects.

Non-Interference Conditions

We distinguish between direct and indirect interference. Non-interference conditions describe the conditions under which two plan instances do not directly interfere. Direct interference refers to the case where the attempt of a plan instance $pi1$ precludes another plan instance $pi2$ from occurring, although the attempt of $pi1$ would not ruin $pi2$'s executability conditions. Two plan instance can directly interfere only if they overlap in time. An example of direct interference is where two simultaneous plan instances share the same resource type and only one resource is available. Another example is where two simultaneous actions are alternatives to each other, such as "move forward at time i " and move backward at time i ".

Indirect interference refers to the case where one plan instance $pi1$ interferes with another one $pi2$ by interfering with a third plan instance that enables $pi2$'s executability conditions. The interaction where an earlier plan instance ruins a later one's executability conditions is an example of indirect interference. As we described in section 5.5, this type of interaction between sequential actions is the principal conflict that must be detected in a state-based system. In our framework, indirect

interference plays a secondary role because it can be computed from direct interference relations.

The non-interference conditions $NI(pi1, pi2)$ are conditions that meet the following relation:

```

NI-RS)
  (INEV Ip (IF NI(pi1, pi2) NI-CND))
  where NI-CND =def
    (IF (AND (EXECUTABLE pi1) (EXECUTABLE pi2))
      (AND (IF (IFTRIED pi1 (EXECUTABLE pi2))
        (IF (OCC pi2) (IFTRIED pi1 (OCC pi2))))
        (IF (IFTRIED pi2 (EXECUTABLE pi1))
          (IF (OCC pi1) (IFTRIED pi2 (OCC pi1)))))))

```

Saying that $NI(pi1, pi2)$ is the non-interference conditions for $pi1$ and $pi2$ is tantamount to saying that $NI-RS$ is in S , the set of sentences that describe the planning problem. Now, we are only interested in whether two plan instances, which are executable individually, interfere with each other. Thus, the conditions $\lceil (EXECUTABLE pi1) \rceil$ and $\lceil (EXECUTABLE pi2) \rceil$ are included in the antecedent of $NI-CND$. This means that non-interference conditions do not need to include the conditions under which the two plan instances are executable individually.

$NI-CND$'s consequent is false only if $pi1$ and $pi2$ directly interfere. The first conjunct in $NI-CND$'s consequent is false if the attempt of $pi1$ would ruin $pi2$'s occurrence but would not ruin $pi2$'s executability conditions. Similarly, the second conjunct is false if the attempt of $pi2$ would ruin $pi1$'s occurrence but would not ruin $pi1$'s executability conditions. If both these conditions are true, then $pi1$ and $pi2$ can only interfere indirectly. Since non-overlapping plan instances can only indirectly interfere, it is not necessary to specify their non-interference conditions; $NI-CND$ holds in all possible branches. This is reflected by the following theorem which we prove in appendix H:

```

NON-OVRLP)
  (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2)) (INEV Ip NI-CND))

```

6.3. Proving the Algorithm is Sound

We prove that our algorithm is sound by showing that the application of a sequence of operators can be mapped to a transformation from the **initial problem state** ST_0 through a succession of problem states ST_1, \dots, ST_n , where ST_i is true if ST_{i+1} is true for every i in the sequence (except $i=n$). The initial problem state is given by:

ST_0)
 $S \models (\exists ?pi (AND (PRIOR Ip (TIME-OF ?pi))$
 $(INEV Ip (EXECUTABLE ?pi))$
 $(INEV Ip (IF (OCC ?pi) G))))$

where S is the set of sentences describing the planning problem

If problem state ST_0 is true then there exists a plan instance in the future of planning time Ip that solves the goal G with respect to description S (see 6.1). A solution sequence corresponds to a transformation from ST_0 to any state where the causal gap is empty, these being states that we will show to be necessarily true. Thus, if there is an operator sequence corresponds to a transformation from ST_0 to a state where the causal gap is empty, ST_0 is true.

To give an overview showing the relation between the planning operators and the problem state transformations, we work through the application of three particular operators starting from the initial problem state. In particular, we describe the problem state ST_1 that would result from applying $INTRO_{pi,C}$ in state ST_0 and the state ST_2 that would result from applying $REMOVE_C$ in ST_1 . To give a sketch of the proofs that we use to prove that the transformations are sound, we present some intermediate steps that are needed to show that if ST_1 holds then ST_0 holds, and to show that if ST_2 is true then ST_1 is true. The intermediate steps are labeled by $INT1, INT2$, etc. We then present the transformation that is produced by introducing a second plan instance which serves to illustrate where the non-interference conditions fit in. Following this, we show that the transformations produced by application of any operator applied in any state ST_i yields a state ST_{i+1} having the property if ST_{i+1} is true then ST_i is true.

An Overview Showing the relation Between the Logic and the Algorithm

The initial casual gap CG_0 is equated with the goal conditions. Thus, we can specify the initial problem state by:

ST_0)

$$S \models (\exists ?pi \text{ (AND (PRIOR } Ip \text{ (TIME-OF } ?pi)) \\ \text{(INEV } Ip \text{ (EXECUTABLE } ?pi))} \\ \text{(INEV } Ip \text{ (IF (OCC } ?pi \text{) } CG_0))))))$$

The introduction of plan instance $pi1$ to remove conjunct $C1$ (i.e. $INTRO_{pi1,C1}$) may be performed iff $C1$ is a conjunct in CG_0 and " $PE \cup \{EFF(pi1)\} \vdash_{IL} C1$ " is true, in which case we say that $C1$ inevitably holds in the planning environment augmented by $pi1$'s effects. The important property of the relation " $PE \cup \{EFF(pi1)\} \vdash_{IL} C1$ " is that if it holds then " $S \models (INEV Ip (IF (OCC pi) C1))$ " holds. The reason for this is as follows. First of all, saying that interval logic statement IL belongs to PE is equivalent to saying that " $(INEV Ip IL)$ " belongs to S . Saying that $EFF(pi1)$ is $pi1$'s effects is equivalent to saying that " $(INEV (IF (OCC pi1) EFF(pi1)))$ " belongs to S . Secondly, our proof theory (of which \vdash_{IL} refers to a limited part) is sound with respect to our semantics. Thus, if " $PE \cup \{EFF(pi1)\} \vdash_{IL} C1$ " holds then " $PE \cup \{EFF(pi1)\} \models C1$ " holds. Finally, we make use of the relation: if " $A \models B$ " is true, then " $\{(INEV Ip Q) \mid Q \in A\} \models (INEV Ip B)$ " is true.

By introducing $pi1$ into the plan, we are transforming ST_0 into a problem state that is true iff there exists a plan instance containing $pi1$ that solves the goal. Thus, we are committing to a particular form to substitute for $?pi$ in S_0 , namely " $(COMP ?pi2 pi1)$ ". Our problem now becomes solving:

$INT1$)

$$S \models (\exists ?pi2 \text{ (AND (PRIOR } Ip \text{ (COMP } ?pi2 \text{ } pi1))} \\ \text{(INEV } Ip \text{ (EXECUTABLE (COMP } ?pi2 \text{ } pi1))} \\ \text{(INEV } Ip \text{ (IF (OCC (COMP } ?pi2 \text{ } pi1)) } CG_0))))))$$

The last conjunct in $INT1$ is true iff it is inevitable at Ip that CG_0 is true if both $?pi2$ and $pi1$ occur together. Thus, the last conjunct is true, if $?pi2$ brings about all the conjuncts in CG_0 that are not brought about by $pi1$. Since, it is inevitable that $C1$ holds if $pi1$ occurs, $INT1$ is true if the following relation holds:

$INT2$)

$$S \models (\exists ?pi2 \text{ (AND (PRIOR } Ip \text{ (TIME-OF (COMP } ?pi2 \text{ } pi1)))} \\ \text{(INEV } Ip \text{ (EXECUTABLE (COMP } ?pi2 \text{ } pi1)))} \\ \text{(INEV } Ip \text{ (IF (OCC } ?pi2 \text{) (} CG_0 - C1))))))$$

In $INT2$, we have used the notation that we presented in the last section. The construct " $(CG_0 - C1)$ " stands for the conjunction of conditions in CG_0 with the exception of $C1$.

The second conjunct in *INT2*, $\lceil (\text{INEV } I_p (\text{EXECUTABLE } (\text{COMP } ?pi2 \text{ } pi1))) \rceil$, may be transformed by making use of a theorem stating that it is inevitable at I_p that $\lceil (\text{COMP } pi2 \text{ } pi1) \rceil$ is executable if i) it is inevitable at I_p that $pi2$ is executable, ii) it is inevitable at I_p if $pi2$ occurs then $pi1$ is executable, and iii) it is inevitable at I_p that if $pi2$ occurs then the attempt of $pi1$ would not ruin $pi2$'s occurrence. Thus, *INT2* is true if the following is

INT3)

$$S \models (\exists ?pi2 (\text{AND} (\text{PRIOR } I_p (\text{TIME-OF } (\text{COMP } ?pi2 \text{ } pi1))) \\ (\text{INEV } I_p (\text{EXECUTABLE } ?pi2)) \\ (\text{INEV } I_p (\text{IF } (\text{OCC } ?pi2) (\text{EXECUTABLE } pi1))) \\ (\text{INEV } I_p (\text{IF } (\text{OCC } ?pi2) (\text{IFTRIED } pi1 (\text{OCC } ?pi2)))) \\ (\text{INEV } I_p (\text{IF } (\text{OCC } ?pi2) (\text{CG}_0 - C1))))))$$

The next step is to replace $\lceil (\text{EXECUTABLE } pi1) \rceil$ in *INT3* with the executability conditions for $pi1$, which we refer to by $\text{EC}(pi1)$. In all models in which all the sentences in S are satisfied, it is inevitable at I_p if $\text{EC}(pi1)$ holds then $pi1$ is executable. Consequently, *INT2* is true if the following holds:

INT4)

$$S \models (\exists ?pi2 (\text{AND} (\text{PRIOR } I_p (\text{TIME-OF } (\text{COMP } ?pi2 \text{ } pi1))) \\ (\text{INEV } I_p (\text{EXECUTABLE } ?pi2)) \\ (\text{INEV } I_p (\text{IF } (\text{OCC } ?pi2) \text{EC}(pi1))) \\ (\text{INEV } I_p (\text{IF } (\text{OCC } ?pi2) (\text{IFTRIED } pi1 (\text{OCC } ?pi2)))) \\ (\text{INEV } I_p (\text{IF } (\text{OCC } ?pi2) (\text{CG}_0 - C1))))))$$

Rearranging the conjuncts in *INT4*, distributing conjunction into *INEV* and making use of the equivalence between sentences of the form $\lceil (\text{INEV } I_p (\text{AND } (\text{IF } P \text{ } Q) (\text{IF } P \text{ } R))) \rceil$ and $\lceil (\text{INEV } I_p (\text{IF } P (\text{AND } Q \text{ } R))) \rceil$, gives us:

INT5)

$$S \models (\exists ?pi2 (\text{AND} (\text{PRIOR } I_p (\text{TIME-OF } (\text{COMP } ?pi2 \text{ } pi1))) \\ (\text{INEV } I_p (\text{EXECUTABLE } ?pi2)) \\ (\text{INEV } I_p (\text{IF } (\text{OCC } ?pi2) (\text{AND } \text{EC}(pi1) (\text{CG}_0 - C1)))) \\ (\text{INEV } I_p (\text{IF } (\text{OCC } ?pi2) (\text{IFTRIED } pi1 (\text{OCC } ?pi2))))))$$

We can replace the last conjunct in *INT5* by $\lceil (\text{INEV } I_p (\text{IF } (\text{EXECUTABLE } pi1) (\text{IF } (\text{OCC } ?pi2) (\text{IFTRIED } pi1 (\text{OCC } ?pi2)))) \rceil$ since if the third conjunct holds, it is inevitable that $pi1$ is executable if $?pi2$ occurs. We also replace $\lceil (\text{PRIOR } I_p (\text{TIME-OF } (\text{COMP } ?pi2 \text{ } pi1))) \rceil$ by $\lceil (\text{PRIOR } I_p (\text{TIME-OF } ?pi2)) \rceil$ because i) I_p is prior to the composition of $?pi2$ and $pi1$ iff I_p is prior to both $?pi2$ and $pi1$, and ii) we assume that that each plan instance that we introduce into the plan, such as $pi1$, is in the future of I_p (thus, we assume " $S \models (\text{PRIOR } I_p (\text{TIME-OF } pi1))$ " is true). Consequently, *INT5* is true if the following is true:

ST₁)

$$S \models (\exists ?pi2 \text{ (AND (PRIOR } Ip \text{ (TIME-OF } ?pi2)) \\ \text{(INEV } Ip \text{ (EXECUTABLE } ?pi2)) \\ \text{(INEV } Ip \text{ (IF (OCC } ?pi2) (AND EC(pi1) (CG_0 - C1)))) \\ \text{(INEV } Ip \text{ (IF (EXECUTABLE } pi1) \\ \text{(IF (OCC } ?pi2) \\ \text{(IFTRIED } pi1 \text{ (OCC } ?pi2)))))))$$

Problem state ST₁ is true if there exists a plan instance *?pi2* in the future of *Ip* such that i) it is inevitable at *Ip* that *?pi2* is executable, ii) it is inevitable at *Ip* that if *?pi2* occurs then both *pi1*'s executability conditions and $\lceil (G - C1) \rceil$ hold, and iii) it is inevitable at *Ip*, if *pi1* is executable, then the attempt of *pi1* would not ruin *?pi2*'s occurrence. The application of the operation $INTRO_{pi,C}$, can be seen as a transformation from problem state ST₀ to problem state ST₁; the intermediate steps *INT1* - *INT5* are just used to show this connection. In the end of this section, we rigorously prove that if ST₁ is true, then ST₀ is true, thereby justifying the transformation from ST₀ to ST₁.

At the completion of cycle 1, the new causal gap CG₁ is set to $\lceil (AND EC(pi1) (G - C1)) \rceil$. Using CG₁ to denote this causal gap, we can rewrite problem state ST₁ as:

ST₁'

$$S \models (\exists ?pi2 \text{ (AND (PRIOR } Ip \text{ (TIME-OF } ?pi2)) \\ \text{(INEV } Ip \text{ (EXECUTABLE } ?pi2)) \\ \text{(INEV } Ip \text{ (IF (OCC } ?pi2) CG_1)) \\ \text{(INEV } Ip \text{ (IF (EXECUTABLE } pi1) \\ \text{(IF (OCC } ?pi2) \\ \text{(IFTRIED } pi1 \text{ (OCC } ?pi2)))))))$$

The second type of operation that may be performed is the removal of a conjunct in CG₁ (i.e. REMOVE_{C2}) that inevitably holds in the planning environment. This operation can be performed iff C₂ is a conjunct in CG₁ and " $PE \vdash_{IL} C2$ " is true, in which case we say that C₂ inevitably holds in the planning environment. Using a similar line of reasoning that we used in going from ST₀ to *INT2* above, we can transform ST₁' into:

ST₂)

$$S \models (\exists ?pi2 \text{ (AND (PRIOR } Ip \text{ (TIME-OF } ?pi2)) \\ \text{(INEV } Ip \text{ (EXECUTABLE } ?pi2)) \\ \text{(INEV } Ip \text{ (IF (OCC } ?pi2) (CG_1 - C2)))) \\ \text{(INEV } Ip \text{ (IF (EXECUTABLE } pi1) \\ \text{(IF (OCC } ?pi2) \\ \text{(IFTRIED } pi1 \text{ (OCC } ?pi2)))))))$$

The formal justification that if ST_2 holds then ST_1' holds is given at the end of this section.

If we remove the last conjunct from CG_1 , then we have found a solution (ie. pi_1). This happens if pi_1 's effects brings about the goal condition G and pi_1 's executability conditions inevitably holds in the planning environment. In this case CG_1 is equal to $EC(\neg 1)$, and $\lceil (CG_1 - C_2) \rceil$ is a tautology, signifying that the causal gap is empty. In the end of this section, we prove a general result that can be used to show that if $\lceil (CG_1 - C_2) \rceil$ is tautology, then ST_2 is vacuously true for any set of sentences S .

To illustrate where non-interference conditions come in, let us investigate the effect of introducing a second plan instance at problem state ST_2 corresponding to operator $INTRO_{pi_2, C_3}$. Substituting CG_2 for $\lceil (CG_1 - C) \rceil$ in ST_2 (the new causal gap) and $\lceil (COMP ?pi_3 pi_2) \rceil$ for $?pi_2$ gives us:

INTS1)

$$S \models (\exists ?pi_3 (AND (PRIOR Ip (TIME-OF (COMP ?pi_3 pi_2))) \\ (INEV Ip (EXECUTABLE (COMP ?pi_3 pi_2))) \\ (INEV Ip (IF (OCC (COMP ?pi_3 pi_2)) CG_2)) \\ (INEV Ip (IF (EXECUTABLE pi_1) \\ (IF (OCC (COMP ?pi_3 pi_2)) \\ (IFTRIED pi_1 (OCC (COMP ?pi_3 pi_2))))))))))$$

The second and third conjuncts in $INT4$ have the same syntactic forms as the second and third conjuncts in $INTS1$. Applying the same transformation to $INTS1$ as we did to get from $INT1$ to $INT4$ gives us:

INTS2)

$$S \models (\exists ?pi_3 (AND (PRIOR Ip (TIME-OF (COMP ?pi_3 pi_2))) \\ (INEV Ip (EXECUTABLE ?pi_3)) \\ (INEV Ip (IF (OCC ?pi_3) (AND EC(pi_2) (CG_2 - C_3)))) \\ (INEV Ip (IF (EXECUTABLE pi_1) \\ (IF (OCC (COMP ?pi_3 pi_2)) \\ (IFTRIED pi_1 (OCC (COMP ?pi_3 pi_2))))))))))$$

The last conjunct in $INTS2$ can be transformed to the following form by using the equivalence between $\lceil (OCC (COMP ?pi_3 pi_2)) \rceil$ and $\lceil (AND (OCC ?pi_3) (OCC pi_2)) \rceil$ and the theorem that conjunction distributes into $INEV$ and into $IFTRIED$:

INTS3)

$$S \models (\exists ?pi3 (AND (PRIOR Ip (TIME-OF (COMP ?pi3 pi2))) \\ (INEV Ip (EXECUTABLE ?pi3)) \\ (INEV Ip (IF (OCC ?pi3) (AND EC(pi2) (CG_2 - C3))) \\ (INEV Ip (IF (OCC ?pi3) (IFTRIED pi2 (OCC ?pi3)))) \\ (INEV Ip (IF (OCC pi2) \\ (IF (EXECUTABLE pi1) \\ (IF (OCC ?pi3) (IFTRIED pi1 (OCC ?pi3)))))) \\ (INEV Ip (IF (OCC ?pi3) \\ (IF (EXECUTABLE pi1) \\ (IF (OCC pi2) (IFTRIED pi1 (OCC pi2))))))))))$$

The last conjunct in *INTS3* is true iff it is inevitable at *Ip* that if *?pi3* occurs then if *pi1* is executable then the attempt of *pi1* would not ruin *pi2*'s occurrence. This is where the non-interference conditions come in. If it is inevitable at *Ip* that the occurrence of *?pi3* brings about the non-interference conditions between *pi1* and *pi2*, then it is inevitable at *Ip* if *?pi3* occurs and *pi1* is executable then *pi1* would not ruin *pi2*'s occurrence. We use *NI(pi1,pi2)* to refer to the non-interference conditions between *pi1* and *pi2*. Using this relation we can transform *INTS3* to form:

INTS4)

$$S \models (\exists ?pi3 (AND (PRIOR Ip (TIME-OF (COMP ?pi3 pi2))) \\ (INEV Ip (EXECUTABLE ?pi3)) \\ (INEV Ip (IF (OCC ?pi3) (AND EC(pi2) (CG_2 - C3))) \\ (INEV Ip (IF (OCC ?pi3) (IFTRIED pi2 (OCC ?pi3)))) \\ (INEV Ip (IF (OCC pi2) \\ (IF (EXECUTABLE pi1) \\ (IF (OCC ?pi3) (IFTRIED pi1 (OCC ?pi3)))))) \\ (INEV Ip (IF (OCC ?pi3) NI(pi1,pi2))))))$$

In the end of the section, we prove that if *pi1* and *pi2* do not overlap in time then the last conjunct in *INTS4* can be omitted.

The next transformation is to replace $\lceil (PRIOR Ip (TIME-OF (COMP ?pi3 pi2))) \rceil$ with $\lceil (PRIOR (TME-OF ?pi3)) \rceil$ (since we are assuming that " $S \models (PRIOR Ip (TIME-OF pi2))$ " is true) and to replace the fifth conjunct in *INTS4* with the stronger statement $\lceil (INEV Ip (IF (EXECUTABLE pi1) (IF (OCC ?pi3) (IFTRIED pi1 (OCC ?pi3)))) \rceil$ to get a simpler form:

INTS5)

$$S \models (\exists ?pi3 (AND (PRIOR Ip (TIME-OF (COMP ?pi3 pi2))) \\ (INEV Ip (EXECUTABLE ?pi3)) \\ (INEV Ip (IF (OCC ?pi3) (AND EC(pi2) (CG_2 - C3))) \\ (INEV Ip (IF (OCC ?pi3) (IFTRIED pi2 (OCC ?pi3)))) \\ (INEV Ip (IF (EXECUTABLE pi1) \\ (IF (OCC ?pi3) (IFTRIED pi1 (OCC ?pi3)))))) \\ (INEV Ip (IF (OCC ?pi3) NI(pi1,pi2))))))$$

Finally, we replace the fourth conjunct by $\lceil (\text{INEV } I_p (\text{IF } (\text{EXECUTABLE } p_{i2}) (\text{IF } (\text{OCC } ?p_{i3}) (\text{IFTRIED } p_{i2} (\text{OCC } ?p_{i2})))) \rceil$ (see step from INT_5 to ST_1 for an analogous justification), and we group together the third and last conjunct in $INTS_5$ to get:

ST_3)

$$S \models (\exists ?p_{i3} (\text{AND}(\text{PRIOR } I_p (\text{TIME-OF } ?p_{i3})) \\ (\text{INEV } I_p (\text{EXECUTABLE } ?p_{i3})) \\ (\text{INEV } I_p (\text{IF } (\text{OCC } ?p_{i3}) \\ (\text{ANDEC}(p_{i2}) (\text{CG}_2 - \text{C3}) \text{NI}(p_{i1}, p_{i2})))) \\ (\text{INEV } I_p (\text{IF } (\text{EXECUTABLE } p_{i2}) \\ (\text{IF } (\text{OCC } ?p_{i3}) (\text{IFTRIED } p_{i2} (\text{OCC } ?p_{i3})))) \\ (\text{INEV } I_p (\text{IF } (\text{EXECUTABLE } p_{i1}) \\ (\text{IF } (\text{OCC } ?p_{i3}) (\text{IFTRIED } p_{i1} (\text{OCC } ?p_{i3})))))))$$

The new causal gap CG_3 is equated with $\lceil (\text{AND} (\text{CG}_2 - \text{EFF}(p_{i2})) \text{NI}(p_{i1}, p_{i2})) \rceil$. This transformation from ST_2 to ST_3 is just like the one for introducing p_{i1} (ST_0 to ST_1) with the exception that we include the non-interference conditions between p_{i1} and p_{i2} in the causal gap (if they overlap). If there were more than one plan instance entered into the plan when we introduced p_{i2} , we would also have to include the non-interference conditions between p_{i2} and each plan instance that overlapped with p_{i2} .

Soundness Proofs for the General Transformations

In section 6.2, we presented the algorithm in terms of the transformations of two state variables INPLAN_i and CG_i . Each pair of variables corresponds to a problem state. This relation is given in figure 6.3-1. The input to the planning problem, given by PE , EC , EFF , and NI , are related to the problem states by constraints that they impose on the set of sentences S appearing on the left side of \models . The constraints imposed by each input is given in figure 6.3-2.

In section 6.2, the two planning operators, REMOVE_C and $\text{INTRO}_{p_i, C}$, were given in terms of variable transformations. The problem state transformations corresponding to REMOVE_C and $\text{INTRO}_{p_i, C}$ are given in figures 6.3-3 and 6.3-4. To show that these transformations are sound, we show that for any S meeting the constraints corresponding to the input specification in 6.3-2, if ST_{i+1} is true then ST_i is true as long as the operator applicability conditions are met. The proofs that these transformations are sound are given in appendix H.

A solution operator sequence corresponds to a transformation from ST_0 to any state where the causal gap is empty, signified by CG_i being a tautology. The solution consists of the composition of the simple plan instances that have been introduced by one of the operators in the sequence. To prove that the initial problem state is true if a state is reached where the causal gap is a empty, we show that any problems state ST_i , where CG_i is a tautology, is true for any set of sentences S . We also show that the composition of simple plan instances that have been entered as part of a solution

operator sequence makes ST_0 true when substituted for $?pi$. These proofs are given in appendix H.

$\langle INPLAN_i, CG_i \rangle$ corresponds to:

GEN- ST_i

$$S \models (\exists ?pi \ (AND(PRIOR \ Ip \ (TIME-OF \ ?pi)) \\
\quad (INEV \ Ip \ (EXECUTABLE \ ?pi)) \\
\quad (INEV \ Ip \ (IF \ (OCC \ ?pi) \ CG_i)) \\
\quad (INEV \ Ip \ (IF \ (EXECUTABLE \ pi1) \\
\quad \quad (IF \ (OCC \ ?pi) \ (IFTRIED \ pi1 \ (OCC \ ?pi)))))) \\
\quad \vdots \\
\quad (INEV \ Ip \ (IF \ (EXECUTABLE \ pii) \\
\quad \quad (IF \ (OCC \ ?pi) \ (IFTRIED \ pii \ (OCC \ ?pi))))))$$

for $pi1, pi2, \dots, pii \in INPLAN_i$

The initial problems state is given by $\langle INPLAN_0, CG_0 \rangle$, where $INPLAN_0$ is empty and CG_0 is set to the goal condition G. This problem state can be written as:

ST_0

$$S \models (\exists ?pi \ (AND(PRIOR \ Ip \ (TIME-OF \ ?pi)) \\
\quad (INEV \ Ip \ (EXECUTABLE \ ?pi)) \\
\quad (INEV \ Ip \ (IF \ (OCC \ ?pi) \ G))))$$

Figure 6.3-1

Constraints imposed on S by the input specifications

PE the planning environment

For every statement IL such that $IL \in PE$, $\neg(INEV Ip IL) \in S$

EC(pi) the executability conditions for each simple plan instance pi

$\neg(INEV Ip (IF EC(pi) (EXECUTABLE pi))) \in S$

EFF(pi) the effects for each simple plan instance pi

$\neg(INEV Ip (IF (OCC pi) EFF)) \in S$

NI(pi1,pi2) the non-interference conditions between two overlapping simple plan instances, pi1 and pi2

$\neg(INEV Ip$
 $(IF NI(pi1,pi2)$
 $(IF (AND (EXECUTABLE pi1) (EXECUTABLE pi2))$
 $(AND (IF (IFTRIED pi1 (EXECUTABLE pi2))$
 $(IF (OCC pi2) (IFTRIED pi1 (OCC pi2))))$
 $(IF (IFTRIED pi2 (EXECUTABLE pi1))$
 $(IF (OCC pi1) (IFTRIED pi2 (OCC pi1))))))$
 $)))))) \in S$

We also assume that every plan instance pi that may be introduced into the plan is in the future of Ip:

$\neg(PRIOR Ip (TIME-OF pi)) \in S$

Figure 6.3-2

Transformation produced by applying $INTRO_{pix,C}$ in problem state ST_i

From
 ST_i)

$$S \models (\exists ?pi \text{ (AND(PRIOR } Ip \text{ (TIME-OF ?pi))} \\ \text{(INEV } Ip \text{ (EXECUTABLE ?pi))} \\ \text{(INEV } Ip \text{ (IF (OCC ?pi) (AND C REST-CG,)))} \\ \text{(INEV } Ip \text{ (IF (EXECUTABLE pi 1)} \\ \text{(IF (OCC ?pi) (IFTRIED pi 1 (OCC ?pi))))))$$

$$\text{(INEV } Ip \text{ (IF (EXECUTABLE pii)} \\ \text{(IF (OCC ?pi) (IFTRIED pii (OCC ?pi))))))$$

for $pi_1, \dots, pii \in INPLAN$.

To

ST_{i+1})

$$S \models (\exists ?pi \text{ (AND(PRIOR } Ip \text{ (TIME-OF ?pi))} \\ \text{(INEV } Ip \text{ (EXECUTABLE ?pi))} \\ \text{(INEV } Ip \text{ (IF (OCC ?pi) (AND REST-CG, EC(pix) } NI_1 \dots NI_k))} \\ \text{(INEV } Ip \text{ (IF (EXECUTABLE pi 1)} \\ \text{(IF (OCC ?pi) (IFTRIED pi 1 (OCC ?pi))))))$$

$$\text{(INEV } Ip \text{ (IF (EXECUTABLE pii)} \\ \text{(IF (OCC ?pi) (IFTRIED pii (OCC ?pi))))))$$

$$\text{(INEV } Ip \text{ (IF (EXECUTABLE pix)} \\ \text{(IF (OCC ?pi) (IFTRIED pix (OCC ?pi))))))$$

for $NI_1 \dots NI_k \in \{NI(pix, pi) \mid pi \in INPLAN, \text{ and } pix \text{ and } pi \text{ overlap in time}\}$
and $pi_1, \dots, pii, pix \in INPLAN$.

where $PEU\{EFF(pix)\} \vdash_{IL} C$

Figure 6.3-4

6.4. Planning Examples

In this section a number of simple planning problems are presented to demonstrate the basic features of our planning algorithm. In particular, we illustrate the use of the two planning operators *REMOVE* and *INTRO*, the role played by interval logic reasoning, and the treatment of both sequential and concurrent interactions. We also comment on the novel aspects of our approach, showing how it enables us to relax some of the restrictions that had to be imposed using other methods.

Both "relative" temporal descriptions, such as "event *ev* occurs after planning time" and temporal relations that are given by referring to intervals forming a date-line structure may be used in our simple algorithm. If a date-line structure is used, the temporal relation between the intervals forming this structure must be specified in the set *PE* (i.e., the planning environment description). For the examples in this section, we make use of a simple date-line structure that is indexed by the terms *I0*, *I1*, *I2*, ..., where *I0* refers to the time of planning, *I1* meets *I0* to the right, *I1* meets *I2* to the right, etc.. We will assume then that the set $\{\overline{(\text{MEETS } I0 \ I1)}, \overline{(\text{MEETS } I1 \ I2)}, \dots\}$, which captures these temporal relations, belongs to *PE* in each of our planning examples.

6.4.1. The Two Planning Operators and Interval Logic Reasoning

To start with, we work through a simple planning example that illustrates the use of the two planning operators, *REMOVE* and *INTRO*, and the role played by a procedure that can compute the derivability relation in interval logic. Consider a safe which is locked at the time of planning (*I0*). At this time also, the key that opens this safe rests on a nearby table. The agent's goal is to have the safe opened by the end of interval *I2*. We show that this goal can be solved by a plan instance composed of the two simple plan instances "grasp the key during *I1*" and "open the safe with the key during *I2*". We will use the term *gr-ky@I1* to refer to this first plan instance and *op-sf@I2* to refer to the second. We assume that *gr-ky@I1* is executable as long as the key is on the table just prior to its execution and that its effects are that the agent is grasping the key by its completion. Secondly, we assume that plan instance *op-sf@I2* is executable if the agent is grasping the key just prior to its execution and its effects are that the safe is open by its completion.

The input specification capturing this planning problem is given by an interval logic statement describing the goal (*G*), a set of interval logic statements describing the planning environment (*PE*), and four interval logic statements describing the executability conditions and effects for both plan instances (*EC*(*gr-ky@I1*), *EFF*(*gr-ky@I1*), *EC*(*op-sf@I2*), and *EFF*(*op-sf@I2*)). This specification, which is listed below, does not include non-interference conditions between *gr-ky@I1* and *op-sf@I2* since these plan instances do not overlap in time.

```

G:      (EXISTS ?i (AND (HOLDS sf-opn ?i)
                        (ENDS≤ ?i I2))))

PE:      {(HOLDS sf-lkd I0), (HOLDS ky-on-tbl I0),
          (FORALL ?i (NOT (AND (HOLDS sf-lkd ?i)
                              (HOLDS sf-opn ?i))))
          (MEETS I0 I1), (MEETS I1 I2), ...}

gr-ky@I1

EC:      (EXISTS ?i (AND (HOLDS ky-on-tbl ?i)
                        (MEETS ?i I1)))

EFF:     (EXISTS ?i (AND (HOLDS ky-grspd ?i)
                        (ENDS= ?i I1)))

op-sf@I2

EC:      (EXISTS ?i (AND (HOLDS ky-grspd ?i)
                        (MEETS ?i I2)))

EFF:     (EXISTS ?i (AND (HOLDS sf-opn ?i)
                        (ENDS= ?i I2)))

```

The sentence assigned to G (the goal conditions) is true if the property "the safe is open" holds during an interval that ends before or at the same time as $I2$, i.e., by the end of $I2$. The planning environment set PE contains sentences describing the date-line structure and sentences capturing that the safe is locked during planning time $I0$, the key is on the table during $I0$, and the safe cannot be both locked and open at the same time. Plan instance $gr-ky@I1$'s executability conditions are true if the property "the key is on the table" holds during an interval that meets $I1$, i.e., a time just prior to $gr-ky@I1$'s execution. Plan instance $gr-ky@I1$'s effects are true if the property "the key is grasped" holds during an interval that ends at the same time as $I1$, i.e., a time that holds during $gr-ky@I1$'s completion. Finally, $op-sf@I2$'s executability conditions are true if the property "the key is grasped" holds just prior to $op-sf@I2$'s execution, and its effects are true if the property "the safe is open" holds at its completion.

During each cycle of the planning algorithm, the state variables $INPLAN$ and CG are updated. Before execution, $INPLAN$ is set to the null set indicating that initially there are no simple plan instances in the plan, and the causal gap CG is set to G , the conjunction of goal conditions. In our example, the goal is given by one condition, and consequently we will say that there is one conjunct forming the causal gap (which is a slight abuse of the language). The operator $REMOVE$ can be used to remove a conjunct C from the causal gap if it is derivable (in the interval logic fragment) from PE . The operator $INTRO$ using a plan instance pi can be used to remove a conjunct C from the causal gap if C is derivable (in the interval logic fragment) from the set containing the members of PE and the sentence $EFF(pi)$. In our example, $INTRO$ applied to $op-sf@I2$ is applicable, while $REMOVE$ and $INTRO$ using $gr-ky@I1$ are not. $INTRO$ using $op-sf@I2$ is applicable because the goal condition is derivable from $\{ \ulcorner (EXISTS ?i (AND (HOLDS sf-opn ?i) (ENDS= ?i I2)) \urcorner \}$, a subset of $PE \cup \{ EFF(op-sf@I2) \}$.

One way to mechanize the process of detecting operator applicability is to employ a theorem prover that forms proofs in interval logic. A resolution theorem prover can be used

since interval logic is cast as a first order theory. Unfortunately, finding an efficient implementation can be quite difficult since potentially there may be any logical relation between a condition in the causal gap and the set formed by the planning environment augmented by a plan instance's effects. In this work, we have not yet investigated efficiency issues, although in the conclusion we sketch some approaches that can be taken. What we have done, however, is to isolate the role of interval logic reasoning and have not confounded it, as Allen and Koomen [Allen&Koomen 83b] did, with non-deductive reasoning to handle the frame problem (persistence) or reasoning to detect action interactions.

A second problem that must be faced is that we want any procedure used to determine operator applicability to halt whether or not the operator is applicable. If we do not restrict the form of our inputs it is impossible to find a procedure that says "no" in all cases when an operator is not applicable. The reason for this is that this problem is equivalent to detecting the lack of derivability in any arbitrary first order theory, an undecidable task. Thus, we may take two approaches (which are not mutually exclusive): we can restrict the form of our inputs, or we can employ a mechanism that may say "no" in cases where a more complete procedure would find a derivation and consequently say "yes". The only ramification of this second limitation is that there may exist a solution to a planning problem that is not detected. Erroneous results are not produced where the planner concludes that a composite plan instance solves a goal, while it could be derived that it does not.

Now, back to our planning example. The effect of introducing a plan instance *pi* to remove a condition *C* from the causal gap is that *C* is removed while *pi*'s executability conditions are added along with the non-interference conditions between *pi* and any plan instance already in the plan (i.e., belonging to *INPLAN*) that overlaps with *pi*. In our example, since *INPLAN* is empty, there are no non-interference conditions to be added. Thus, *CG* at the end of the first cycle is set to:

(EXISTS ?i (AND (HOLDS ky-grspd ?i) (MEETS ?i I2)))

and *INPLAN* is set to {*op-sf@I2*}.

At the second cycle, *INTRO* using *gr-ky@I1* is applicable since the condition making up the causal gap is derivable from { \uparrow (EXISTS ?i (AND (HOLDS ky-grspd ?i) (ENDS= ?i I1)) \uparrow , \uparrow (MEETS I1 I2) \uparrow }, a subset of $PE \cup \{EFF(gr-ky@I1)\}$. Notice that this derivation mentions the formula \uparrow (MEETS I1 I2) \uparrow , which captures part of the date-line structure. This formula is needed because the condition to be removed is in terms of *I2*, while the plan instance's effects are in terms of *I1*.

The effect of introducing *gr-ky@I1* is that *INPLAN* is set to {*op-sf@I2*, *gr-ky@I1*}, and *CG* is set to:

(EXISTS ?i (AND (HOLDS ky-on-tbl ?i) (MEETS ?i I1)))

In forming this new causal gap, only *gr-ky@I1*'s executability conditions are added, and not any non-interference conditions, because *op-sf@I2* is the only member of *INPLAN* and it does not overlap with *gr-ky@I1*.

At cycle three, *REMOVE* can be used to remove the condition making up the causal gap since this condition is derivable from { \uparrow (MEETS I0 I1) \uparrow , \uparrow (HOLDS ky-on-tbl I0) \uparrow }, a subset of *PE*. In this example, after applying *REMOVE* an empty causal gap is produced because this operator removes the single condition forming the causal gap, and *REMOVE* never adds any new conditions. Consequently, the algorithm terminates indicating that the plan instance

$\overline{[COMP\ gr-ky@I1\ op-sf@I2]}$, which is the composition of the members of *INPLAN*, solves the goal *G*.

If we modified the above example so that the sentence $\overline{[HOLDS\ ky-on-tbl\ I0]}$ was not included in *PE*, we would not be able to remove the condition making up the causal gap at cycle three. In general, the only ramification that can result from an incomplete planning environment specification is that there might be a goal that could be solved, but this cannot be detected from the information provided in the input specification. This contrasts with other planning systems that handle external events, such as *DEVISER* [Vere 81] and Allen's and Koomen's algorithm [Allen&Koomen 83b], which can produce erroneous results if an incomplete planning description is used. In particular, these systems can act as if a property persists that actually does not (a conclusion reached from the omission of (relevant) information). For example, if the above planning environment was encoded in either of these systems, and we omitted a description capturing that a safe cannot both be locked and open at the same time, these system can act as if the safe remains locked even after *op-sf@I2*, which causes the safe to be opened, occurs.

Similarly, the only ramification of using imprecise descriptions, such as "either event *ev1* or *ev2* occurs during *I1*", (a disjunctive description), or "*pr* holds sometimes during *I1*", is that there might be a goal that could be solved, but this cannot be detected. In other planning systems, either these type of descriptions are precluded or erroneous results can be produced. Allen's and Koomen's algorithm is an example of the later. For instance, if they use a planning environment description capturing that "*pr* holds during *I0*" and "the negation of *pr* holds sometimes during interval *I1-2* (the concatenation of *I1* and *I2*)", their system can act if *pr* holds during *I1*, which is a conclusion that does not hold under the possible outcome "the negation of *pr* holds during *I1*".

6.4.2. Sequential Interactions and Maintenance

We now consider a situation where an earlier plan instance may ruin a later one's executability conditions, this being the principal conflict detected in state-based systems. To illustrate this situation, we complicate our first example by adding a second condition that must hold in order for *op-sf@I2* to be executable: the agent must be within arms reach of the safe just prior to execution (which implies that *op-sf@I2* does not involve moving to the location of the safe if moving is necessary). Thus, *op-sf@I2*'s executable conditions are now taken to be:

EC(*op-sf@I2*)

(AND (EXISTS ?i (AND (HOLDS *ky-grspd* ?i) (MEETS ?i *I2*)))
(EXISTS ?i (AND (HOLDS *sf-arms-ln* ?i) (MEETS ?i *I2*))))

where the term *sf-arms-ln* refers to the property "the safe is within arms length". We also assume that this property holds at planning time, and thus modify *PE* from the above example to include $\overline{[HOLDS\ sf-arms-ln\ I0]}$.

Let us now examine the operation of the planning algorithm with these two modifications. We first iterate through three cycles of the planning algorithm as before. The only differences from our earlier example is that there are two conjuncts, not a single one, in the causal gap after *INTRO* using *op-sf@I2* is applied. Consequently, there are two conjuncts after *gr-ky@I1* is introduced and one after *REMOVE* is applied. In particular, after *INTRO* using *op-sf@I2* is applied, *CG* is set to:

(AND (EXISTS ?i (AND (HOLDS ky-grspd ?i) (MEETS ?i I2)))
 (EXISTS ?i (AND (HOLDS sf-arms-ln ?i) (MEETS ?i I2))))

After applying *INTRO* using *gr-ky@I1* to remove the first conjunct, *CG* is set to:

(AND (EXISTS ?i (AND (HOLDS ky-on-tbl ?i) (MEETS ?i I1)))
 (EXISTS ?i (AND (HOLDS sf-arms-ln ?i) (MEETS ?i I2))))

Finally, after applying *REMOVE* to remove the first conjunct, *CG* is set to:

(EXISTS ?i (AND (HOLDS sf-arms-ln ?i) (MEETS ?i I2)))

At this stage, the planner is at an impasse. The condition making up the causal gap cannot be removed using *INTRO* applied to any plan instance presented so far. Neither can *REMOVE* be used; from the planning environment description, one can infer only that the property *sf-arms-ln* holds during *I0*. This specification is noncommittal as to whether this property holds at any later times, such as a time just prior to *I2*. Unlike existing systems, we do not operate as if a property remains true unless it can be determined otherwise. Instead, a plan instance must be introduced into the plan to guarantee that a property remains true over some interval; i.e., the property must be maintained. We now extend our example by introducing such a plan instance.

Whether there exists a plan instance to maintain a property depends on whether the agent can affect this property: a property cannot be maintained if it is totally out of the agent's control. In this example, we assume that the property "the safe is within arms length" is not totally out of the agent's control, and we will use the term *(mtn sf-arms-ln)@I1* to refer to a plan instance that maintains this property over interval *I1*. The specification for this plan instance is given by:

(mtn sf-arms-ln)@I1

EC: (EXISTS ?i (AND (HOLDS sf-arms-ln ?i) (MEETS ?i I1)))

EFF: (HOLDS sf-arms-ln I1)

The above specification captures that *(mtn sf-arms-ln)@I1* is executable if the safe is within arms length at a time just prior to execution, and its effect is that this property holds during execution. It is important to note that this specification implies not only that the agent can influence the property "the safe is within arms reach" during time *I1*, but furthermore, that there are no external events that can prevent the agent from maintaining this property during *I1*. If for example, it were possible that another agent could perform an action that would result in the safe being moved sometimes during *I1*, *(mtn sf-arms-ln)@I1*'s executability conditions would have to mention that this action does not occur.

Since *gr-ky@I1* and *(mtn sf-arms-ln)@I1* overlap in time, their non-interference conditions must be specified. If the performance of *gr-ky@I1* does not involve a change in location, then *gr-ky@I1* and *(mtn sf-arms-ln)@I1* do not interfere under any conditions. In this case, we can set their non-interference conditions to TRUE. On the other hand, if grasping the key necessarily involves moving out of arms reach from the safe, *gr-ky@I1* and *(mtn sf-arms-ln)@I1* would interfere under all conditions, and consequently we would set their non-interference conditions to FALSE. The third alternative is where the two plan instances interfere only under certain conditions. As an example, suppose that grasping the key involves

moving out of arms distance from the safe only if the key is not resting on the side of the table closest to the safe. In this case, we could set their non-interference conditions to $\neg(\text{AND}(\text{EXISTS } ?i (\text{HOLDS } \text{ky-cls } ?i) (\text{MEETS } ?i \text{ I1})))$, which we assume to be true if and only if the key is on the side of the table closest to the safe at the time just before the key may be grasped. We now investigate the operation of our planning algorithm in all three cases.

To begin with, what gr-ky@I1 's and $(\text{mtn sf-arms-ln})@I1$'s non-interference conditions is set to does not affect whether *INTRO* using $(\text{mtn sf-arms-ln})@I1$ is applicable. For the causal gap considered above, this operator is applicable since $\neg(\text{EXISTS } ?i (\text{AND}(\text{HOLDS } \text{sf-arms-ln } ?i) (\text{MEETS } ?i \text{ I2})))$ is derivable from $\{\neg(\text{HOLDS } \text{sf-arms-ln } \text{I1}), \neg(\text{MEETS } \text{I1 } \text{I2})\}$, a subset of $\text{PEU}\{\text{EFF}((\text{mtn sf-arms-ln})@I1)\}$.

The result of introducing $(\text{mtn sf-arms-ln})@I1$ into the plan when *CG* is set to $\neg(\text{EXISTS } ?i (\text{AND}(\text{HOLDS } \text{sf-arms-ln } ?i) (\text{MEETS } ?i \text{ I2})))$ and *INPLAN* is set to $\{\text{op-sf@I2}, \text{gr-key@I1}\}$ is that the condition making up the causal gap is removed, while $(\text{mtn sf-arms-ln})@I1$'s executability conditions and the non-interference conditions between $(\text{mtn sf-arms-ln})@I1$ and gr-ky@I1 are added. In the case where $(\text{mtn sf-arms-ln})@I1$ and gr-ky@I1 do not interfere (under any conditions), the following causal gap would be produced:

(AND (EXISTS ?i (AND (HOLDS sf-arms-ln ?i) (MEETS ?i I1)))
TRUE)

During the next two cycles both these conditions could be removed by applying *REMOVE*. The first conjunct can be removed since it follows from $\{\neg(\text{HOLDS } \text{sf-arms-ln } \text{I0}), \neg(\text{MEETS } \text{I0 } \text{I1})\}$, a subset of *PE*. TRUE can be removed since it is derivable from any set. Thus, to gain efficiency, one can modify our planning algorithm so that TRUE is ignored (i.e., not added to the causal gap) in situations where it would have been added using our simple algorithm.

In the case where the two plan instances interfere under all conditions, the result of introducing $(\text{mtn sf-arms-ln})@I1$ into the plan is that *CG* is set to:

(AND (EXISTS ?i (AND (HOLDS sf-arms-ln ?i) (MEETS ?i I1)))
FALSE)

This new causal gap is unsolvable since FALSE is not derivable from any set. Thus, to gain efficiency, one can modify our planning algorithm by considering any operator that would enter FALSE into the causal gap (using our simple algorithm) as not being applicable.

The last case to consider is where the two plan instances interfere only if the key is not resting on the side of the table closest to the safe, and thus we set $(\text{mtn sf-arms-ln})@I1$'s and gr-ky@I1 's non-interference conditions to $\neg(\text{AND}(\text{EXISTS } ?i (\text{HOLDS } \text{ky-cls } ?i) (\text{MEETS } ?i \text{ I1})))$. In this case, introducing $(\text{mtn sf-arms-ln})@I1$ into the plan produces the new causal gap:

(AND (EXISTS ?i (AND (HOLDS sf-arms-ln ?i) (MEETS ?i I1)))
(EXISTS ?i (HOLDS ky-cls ?i) (MEETS ?i I1)))

Like the cases above, the first conjunct in this causal gap can be removed since it is derivable from *PE*. To remove the second condition, we would need to show that this condition is derivable from the planning environment. The second condition cannot be removed using *INTRO* since this condition is not in the future of planning time and we can only introduce plan instances that occur after planning time.

In these examples, whether *gr-ky@I1* ruins *op-sf@I2*'s executability conditions is detected through the interaction between *gr-ky@I1* and *(mtn sf-arms-ln)@I1*, which is captured by their executability conditions. There are a number of advantages gained by detecting interactions in this fashion, as opposed to the standard approach where interactions are detected by looking at the relation between preconditions and effects and using phantom nodes. For one, our method is more expressive in the type of sequential interactions that can be treated. In other systems, either an earlier action ruins a later one's preconditions, or does not. In our framework, we can also model the situation where an earlier plan instance conditionally ruins a later one's precondition. Our last example is one of these cases.

Secondly, we have more flexibility in how we describe plan instance (action) effects because they are used in our system only to determine if an action can achieve a condition, rather than for determining both achievement relations and action conflicts, as is typically done. For example, our algorithm does not require a complete and non-disjunctive specification of a plan instance's (action's) effects. This is in contrast to the standard methods which must work under the assumption that each action's effect list is complete and must preclude the use disjunctive effects.¹ Using the standard methods, if *EFF* is an actual effect of *a1* that is omitted from *a1*'s effect list, one would not detect the harmful interaction where *a1* conflicts with a later action that has a precondition mentioning the negation of *EFF*. In our system, the only ramification of an incomplete effect list is that there may be a goal that can actually be achieved, but this cannot be determined from the incomplete description given as input.

Lastly, the use of maintenance plan instance allows us to make a distinction that cannot be captured using phantom nodes. In particular, we are able to distinguish between three cases:

- 1) The agent cannot affect property *pr* during time *ix*; in which case, there does not exist a plan instance that maintains *pr* any time during *ix*.
- 2) Property *pr* is completely in the agent's control during interval *ix*; in which case, the only executability conditions for maintaining *pr* during *ix* is that *pr* holds just prior to *ix*.
- 3) Property *pr* is affected by both the agent and the external world during interval *ix*; in which case, the executability conditions for a plan instance that maintains *pr* during *ix* must mention the external events or conditions that could prevent this property from holding.

Being able to make the above distinctions is critical if we permit external events and do not assume that the planning environment description is complete.

¹ In our system, if we have a non-deterministic plan instance that brought about either property *p* at time *ix* or property *q* at time *ix*, we could set its effect list to this disjunction. We must also set to FALSE the non-interference conditions between *pi* and any plan instance that either maintains the negation of *p* during a time overlapping with *ix* and between *pi* and any plan instance that maintains the negation of *q* during a time overlapping with *ix*.

6.4.3. Concurrent Interactions

We now examine plan instance interactions that would be detected as concurrent interactions in other systems. We first present a simple example showing how we can model the conflict between two plan instances that share the same type of resource. We then complicate this example to take into account the possibility where an external event is also competing for this resource type.

Consider two plan instances $ev1@I1$ and $ev2@I1$ that share the same type of resource. If there is just one resource available then one but not both can be executed, while if there are two or more resources available then they both can be executed together. We will use the function term $[rsrc-avl\ n]$ to denote the property " n resources are available". To focus on the interaction between $ev1@I1$ and $ev2@I1$, we set up the following simple planning problem:

G:	(AND (HOLDS $pr1\ I1$) (HOLDS $pr2\ I1$))
PE:	{(FORALL $?i$ (IF (HOLDS ($rsrc-avl\ 2$) $?i$) (HOLDS ($rsrc-avl\ 1$) $?i$))) (FORALL $?i$ (IF (HOLDS ($rsrc-avl\ 3$) $?i$) (HOLDS ($rsrc-avl\ 2$) $?i$))) (MEETS $I0\ I1$), (MEETS $I1\ I2$), ...}
$ev1@I1$	
EC:	(HOLDS ($rsrc-avl\ 1$) $I1$)
EFF:	(HOLDS $pr1\ I1$)
$ev2@I1$	
EC:	(HOLDS ($rsrc-avl\ 1$) $I1$)
EFF:	(HOLDS $pr2\ I1$)
{ $ev1@I1, ev2@I1$ }	
NI:	(HOLDS ($rsrc-avl\ 2$) $I1$)

In the planning environment set PE , we include the two instances of a general relation, that we will make use of, capturing that if there is at least n resources available during any time i , then there is also at least $n-1$ resources available during i . The executability conditions specifications for $ev1@I1$ and $ev2@I1$ capture that both plan instances can be executed individually as long as one resource is available. The non-interference conditions specification captures that the two plan instances do not interfere if at least two resources are available. Implicit in the executability and non-interference specifications is that there are no external events or other plan instances that can compete for the same resource. In our second example, we show how the specification can be modified to provide for this possibility.

Using our planning algorithm, the interaction between $ev1@I1$ and $ev2@I1$ would first be considered when one of them is already in the plan and we are considering introducing the other. One can get to such a state by first introducing either $ev1@I1$ or $ev2@I1$ into the plan, these both being applicable at the first cycle when the causal gap is set to the conjunction assigned to G . Suppose we first introduce $ev1@I1$ into the plan, which can be used to remove

the first conjunct in G . The result of applying this operator is that CG is set to:

```
(AND (HOLDS pr2 I1)
      (HOLDS (rsrc-avl 1) I1))
```

and $INPLAN$ is set to $\{ev1@I1\}$.

At the second cycle, $ev2@I1$ can be introduced to remove $\lceil \text{HOLDS pr2 I2} \rceil$ from the causal gap. The new causal gap produced by this operator is formed from the previous one by removing $\lceil \text{HOLDS pr2 I2} \rceil$ while adding $ev2@I1$'s executability conditions and the non-interference conditions between $ev1@I1$ and $ev2@I1$ yielding:

```
(AND (HOLDS (rsrc-avl 1) I1)
      (HOLDS (rsrc-avl 1) I1)
      (HOLDS (rsrc-avl 2) I1))
```

This new causal gap contains the conditions that must hold or be brought about in order for the composition of $ev1@I1$ and $ev2@I1$ to be executable. It can be deduced that all these conditions hold if $\lceil \text{HOLDS (rsrc-avl 2) I1} \rceil$ is derivable from PE . In this case, the planning operator $REMOVE$ can be applied three times to remove these three conjuncts from the causal gap. The reason that $\lceil \text{HOLDS (rsrc-avl 1) I1} \rceil$ can be removed (twice) is because of the presence of $\lceil \text{FORALL ?i (IF (HOLDS (rsrc-avl 2) ?i) (HOLDS (rsrc-avl 1) ?i))} \rceil$ in PE . For similar reasons, if some plan instance can be introduced to remove $\lceil \text{HOLDS (rsrc-avl 2) I1} \rceil$, then this plan instance can be introduced three times to remove the three conditions. While introducing the same plan instance more than once might be wasteful, it does not produce erroneous results.

One can modify our simple algorithm to avoid this wasteful behavior where an operator is applied more than once to remove the same condition from a causal gap. A simple solution is to avoid this situation by only adding new conditions to the causal gap if they are not already present (and not allowing duplicates in the goal). A more encompassing solution is to only add a condition C to the causal gap, if it is not derivable from the planning environment augmented with the conditions in the causal gap. Thus, for example, if condition $C1$ were in the causal gap and $\lceil \text{IF } C1 \ C2 \rceil$ were in PE , one would not add condition $C2$ to the causal gap.

Let us now complicate the above example by taking into account an external event that competes for the same type of resource used by $ev1@I1$ and $ev2@I1$. For simplicity, we will assume that only one external event, which we will designate by $ext-ev$, occurring during $I1$ competes for this resource type during $I1$. In this case, each plan instance is executable only if either there are at least two resources available during $I1$ (so that whether or not $ext-ev$ occurs during $I1$ there will still be enough resource available), or $ext-ev$ does not occur during $I1$ and there is at least one resource available. We therefore modify the above example so that $ev1@I1$'s and $ev2@I1$'s executability conditions are both given by:

```
(OR (HOLDS (rsrc-avl 2) I1)
     (AND (HOLDS (rsrc-avl 1) I1) (NOT (OCCURS ext-ev I1)))
```

We must also modify the non-interference conditions between $ev1@I1$ and $ev2@I1$ to provide for the possibility where $ext-ev$ occurs during $I1$; in which case they would interfere if at least three resources were not available. Thus, we assume that their non-interference conditions are now given by:

```
(OR (HOLDS (rsrc-avl 3) I1)
    (AND (HOLDS (rsrc-avl 2) I1) (NOT (OCCURS ext-ev I1)))
```

A similar specification could be used if instead a third plan instance was competing for the resource.

Using these modifications, the causal gap produced by introducing both *ev1@I1* and *ev2@I1* is given by:

```
(AND (OR (HOLDS (rsrc-avl 2) I1)
          (AND (HOLDS (rsrc-avl 1) I1) (NOT (OCCURS ext-ev I1))))
    (OR (HOLDS (rsrc-avl 2) I1)
        (AND (HOLDS (rsrc-avl 1) I1) (NOT (OCCURS ext-ev I1))))
    (OR (HOLDS (rsrc-avl 3) I1)
        (AND (HOLDS (rsrc-avl 2) I1) (NOT (OCCURS ext-ev I1)))))
```

If $\neg(\text{HOLDS (rsrc-avl 3) I1})$ is derivable from *PE*, then we could remove all three conjuncts from the causal gap using *REMOVE*. The last condition could be removed since its left disjunct would be derivable from *PE*. The first two conditions could be removed since their left disjuncts (i.e., both $\neg(\text{HOLDS (rsrc-avl 2) I1})$) would be derivable from *PE* because of the presence of $\neg(\text{FORALL ?i (IF (HOLDS (rsrc-avl 3) ?i) (HOLDS (rsrc-avl 2) ?i))})$ in *PE*. Similarly, if some plan instance could be introduced to remove $\neg(\text{HOLDS (rsrc-avl 3) I1})$, then this plan instance could be introduced three times to remove these three conditions.

We could also remove all three conditions if both $\neg(\text{NOT (OCCURS ext-ev I1)})$ and $\neg(\text{HOLDS (rsrc-avl 2) I1})$ are derivable from *PE*, meaning that it is deducible that *ext-ev* will not occur during *I1* and that there will be at least two resources available at this time. We could not introduce a plan instance, however, to achieve both conditions since *ext-ev* is an external event, and thus we are assuming that no plan instance can affect it and consequently $\neg(\text{NOT (OCCURS ext-ev I1)})$ cannot be achieved. Instead, if $\neg(\text{NOT (OCCURS ext-ev I1)})$ is derivable from *PE*, and plan instance *pi* achieves $\neg(\text{HOLDS (rsrc-avl 2) I1})$, we could introduce *pi* to remove all three conditions from the causal gap.

Chapter 7

Conclusion

7.1. Summary

In this dissertation an interpreted deductive logic was developed to describe and reason about planning problems that may involve external events and concurrent actions. We also developed a simple planning algorithm that can handle these features. A rigorous construction was provided demonstrating that this algorithm can be viewed as a sound (but limited) proof procedure.

We motivated our development by first describing the inadequacies of state-based systems and their underlying formalism, situation calculus, which is the context in which planning was originally formulated. While situation calculus is suitable for describing the effects of executing different actions, both individually or in a sequence, from some instantaneous state, it cannot directly model simultaneous events or conditions that may hold while an action is to be executed. As a result, this formalism is insufficient for describing planning problems that mention either concurrent actions or external events that may be taking place while the agent is to be executing its plan. There has been, however, work in extending the capabilities of state-based planners to handle some of these features, while keeping close to the state-based framework. In this work, we discussed SIPE [Wilkins 83] and DEVISER [Vere 81], which are two of the more sophisticated types. These systems, however, have many restrictions. We illustrated that some of these restrictions stem from their use of the STRIPS assumption which is an approach intimately tied to the state-based framework, which is used to determine whether a property remains true.

In response to the deficiencies of the state-based framework, Allen [Allen 84] and McDermott [McDermott 82] put forth temporal models that can be used to describe simultaneous events and conditions that hold or are changing while an event is occurring. We demonstrated, however, that both Allen's model (which is a linear time model) and McDermott's model (which is a branching time model) lacked the structure necessary to describe the different ways the agent can affect the world by executing the different actions at its disposal. That is, although these models are suitable for capturing the perspective of an "outside observer", they are not suitable from the perspective of an agent that is an active participant, one who can influence the world.

More specifically, we demonstrated that neither model can be used to express "temporally rich" planning problems having the form: given a description of the world in which planning is to be done that may specify conditions that either possibly or inevitably hold in the future of planning time, find a collection of actions at specified times (i.e., a plan instance) that achieves a desired future condition (i.e., the goal). To capture such problems, for example, it is necessary to distinguish possibilities that are out of the agent's control from ones that the agent can influence. We demonstrated that neither Allen's or McDermott's model can make this distinction. Secondly, we showed that neither model can express that some action can be executed only under certain conditions, where these conditions may refer to times that hold while the action is to be executed.

To remedy these problems, we developed a model of action and time that can be viewed as a branching time model (indexed by temporal intervals) extended with a function analogous to the result function in situation calculus. This function captures the different ways in which the

agent can affect the world. In our models, **world-histories** and **basic action instances** take the place of situations and actions. World-histories refer to complete worlds over time, rather than to instantaneous snapshots. Basic action instances refer to primitive actions at a specified times and are used to construct **plan instances**, our analog to plans which can be thought of as collections of actions at specified times. A world-history serves as the context in which the execution of a basic action instance is specified. For each world-history h and basic action instance $ba@i$, the model specifies the world-histories that differ from h solely on the account of $ba@i$'s occurrence. This function is similar to ones found in the semantic models for conditionals developed by Stalnaker [Stalnaker 68] and Lewis [Lewis 73]. This structure enables us to model the influence of conditions that may hold during the time that a plan instance is to be executed, and provides a simple basis for modeling concurrent interactions and for composing plan instances to form more complex ones.

The models we developed served as the semantic structures used to construct and interpret a deductive logic. To describe these models, we extended Allen's language, which is a first order language, with two modal operators *INEV* and *IFTRIED*. Statements formed using the *INEV* operator are used to describe the branching structure; that is, *INEV* is used to describe conditions that are inevitable at a specified time and ones that are possible. Statements formed by the *IFTRIED* operator are used to describe the conditions that would be affected if the agent were to attempt a specified plan instance and the ones that would not be affected.

After specifying the language and semantic theory, we provided an axiomatization and showed that the resulting proof theory is sound with respect to the semantics. We then illustrated how temporally rich planning problems could be expressed in the logic and provided derivations showing the relation between composite plan instances and their constituent parts. We also investigated plan instance interactions, paying particular attention to concurrent interactions.

Using our framework, we re-examined the STRIPS assumption and its analog "the persistence assumption" [McDermott 82] which is a non-deductive scheme used for detecting whether a property remains true when simultaneous events are allowed. We demonstrated that these assumptions are inappropriate when planning with simultaneous events. As an alternative, we introduced plan instances that maintain properties over intervals and investigated their relation to the STRIPS assumption.

Lastly, we presented a simple planning algorithm that handles concurrent actions and external events. We proved that this algorithm is sound with respect to our semantics, using our general proof theory as a tool. This algorithm is novel in that all action interactions, both sequential and concurrent, derive from a conditional *interference* relation specified by the user that relates only concurrent plan instances. Secondly, we employ maintenance plan instances, rather than resorting to the STRIPS assumption. This enables us to relax some pressing restrictions that had to be imposed when using the STRIPS assumption, such as requiring a description of all the (relevant) external events that will occur during the time that plan execution is to take place.

7.2. Limitations and Future Directions

We describe extensions of the work presented in this dissertation along two different dimensions. We first discuss implementing and extending our simple planning algorithm to gain efficiency. We then discuss some issues that are outside the scope of "the deductive planning problem", which we have formulated, and focus on the role that would be played by our deductive logic when considering these more encompassing issues.

7.2.1. Limitations of the Planning Algorithm

When presenting our simple planning algorithm in chapter six, we did not consider efficiency issues. In a number of places the algorithm is abstract in the sense that there are many implementations, affording various degrees of efficiency, that meet the specification. Thus, we must fill in the details, trying to find an efficient implementation. In particular, we would like i) an efficient mechanism for determining which planning operators are applicable at each cycle, ii) a mechanism that wisely chooses the appropriate operator from the set of applicable ones at each cycle, and iii) a mechanism that detects that an unsolvable state has been reached and backtracking should be performed. In this sub-section, we discuss i) and iii) in some detail.

Efficiency can also be improved by adapting techniques employed by some current state based planning systems, such as constraints [Stefik 81], abstract actions [Sacerdoti 74] [Tenenbergs 86], and decomposable actions [Sacerdoti 77]. Unlike the issues mentioned in the paragraph above, to incorporate these features it is necessary to modify our algorithm rather than just plugging in the details. In this sub-section, we describe planning with constraints in more detail.

Lastly, we discuss the problem of finding alternate input specifications that may be more readily available than those required by our simple algorithm.

Determining Applicable Operators (Mechanizing Interval Logic)

In this work, we have not investigated efficient techniques for determining which planning operators are applicable at each cycle; we have just given "correctness conditions" that must be met by any procedure used to perform this task. As we have seen, detecting operator applicability involves determining whether an interval logic statement is derivable from a set of interval logic statements. This does not necessitate, however, that the mechanism for determining operator applicability must be a theorem prover that forms proofs in interval logic. Any procedure can be used as long as the procedure concludes that an operator is applicable only when the condition being removed from the causal gap is derivable from the set of sentences associated with the operator (and that it halts on all inputs).

One approach that can be taken to gain efficiency is to restrict the form of the inputs to the planning algorithm. We can look at existing planning systems to find suitable restrictions, such as NOAH [Sacerdoti 77], SIPE [Wilkins 83], or DEVISER [Vere 81]. In these systems, simple and efficient mechanisms have been found to check whether an action achieves some condition because these systems can express only simple relations between an action's effects and conditions to be achieved. Moreover, these mechanisms can be interpreted as meeting our criteria for correctness. Thus, one may build an efficient mechanism by first restricting our planner's inputs to guarantee that the relation between any plan instance's effects and any condition in the causal gap only corresponds to a relation found in one of these planners chosen

as our model. We then can implement a mechanism for detecting operator applicability that is analogous to the one used by the model planner, thereby affording the same efficiency.

The limitation of this approach is in the loss in expressability. Current planning systems do not describe the variety of examples that we wish to handle. For example, they cannot describe actions (plan instances) that have disjunctive effects or describe the various temporal relations we would like to consider. One might then try to extend a mechanism for detecting operator applicability so that we can relax some of the more imposing restrictions placed on the form of the inputs.

If we are striving for expressability, an interval logic theorem prover might be a more appropriate mechanism for detecting operator applicability. In a theorem prover framework, one typically strives for efficiency by implementing an appropriate control strategy. This control strategy may be domain dependent, driven by information about the specific problem being considered.

We may also improve the efficiency of an interval logic theorem prover by trying to separate interval relation reasoning (i.e., reasoning about temporal relations) from reasoning about the causal and definitional relations between events and properties. We can then exploit an efficient procedure tuned for interval relation reasoning to gain efficiency. Stickel [Stickel 85] has developed a technique, called theory resolution, that may be used for this purpose. His method enables one to augment a resolution theorem prover, given a particular theory, with special purpose procedures tuned for efficiency that reason about sub-theories. A special purpose procedure can be employed for a sub-theory when a mechanism can be found that detects unsatisfiability in this sub-theory.

In our case, we would naturally look at Allen's interval relation reasoner [Allen 83a], which can detect an unsatisfiable set of ground interval relation statements. Thus, to employ theory resolution, we may restrict the inputs to our planning algorithm so that only ground interval relation statements need be considered, or we may extend Allen's mechanism to handle quantified statements. The former approach, however, may be overly restrictive; for example, it would preclude the use of the statement "event *ev* occurs after planning time" in the planning environment description. In general, this approach precludes the use of any statement in the planning environment or effect lists that would contain a free variable in an interval relation formula when put in clause form; it also precludes the use of any statement in the goal, executability, or non-interference conditions (these being the conditions that can be added to the causal gap) that would contain a free variable in an interval relation formula when its negation is put in clause form.

Backtracking and Detecting when the Causal Gap is Unsolvable

We want to provide for the situation where during the planning process a bad choice, or a bad set of choices, has been made. In this case some of the earlier choices made (and their ramifications) must be retracted before proceeding. We would like a mechanism that implements this process to recognize when a bad state has been reached in as many cases as possible and as early as possible. We would also like a retraction mechanism that only retracts the choices that are causing the problem.

In our planning algorithm, a bad state corresponds to the situation where the causal gap is unsolvable, that is, when there is no applicable sequence of operators that can be used to remove all the conditions from the causal gap. This situation is immediately detectable when there exists a condition in the causal gap that cannot be removed by any applicable operator.

One can also determine that the causal gap is unsolvable by detecting that there exists a condition in the causal gap that is impossible to achieve. We must clarify by noting that "being impossible" is a stronger (i.e., more specific) condition than "being unsolvable". The causal gap would be unsolvable but not impossible in cases where there are plan instances that could be performed to achieve the conditions in the causal gap, but this cannot be determined because the (incomplete) description of the planning problem either does not mention these plan instances or does not mention all their effects.

In section 6.4.2 we encountered a simple case where we detected that the causal gap is impossible, this being the case when the causal gap contained the condition FALSE (which stands for an inconsistent statement). More generally, it can be shown that the causal gap is impossible to achieve if the conditions in the causal gap taken with the conditions in the planning environment are inconsistent. There are also additional cases where it could be detected that the causal gap is unsolvable if we extended the type of information that could be provided as input. For example, we might include information about conditions that are both possibly false and out of the agent's control. If condition C has this property, then any causal gap containing C is impossible. We must note that we could not simply conclude that a causal gap containing a condition that is possibly false (and not necessarily out of the agent's control) is impossible; it might be the case that a condition is possibly false because of a course of action that the agent can take but is not going to take in the plan under consideration.

Let us now turn to the problem of isolating the choices that produce a bad state. In order to narrow down the choices to retract when the causal gap is unsolvable, it is necessary to find the conditions in the causal gap that are the source of the problem. If it is detected that the causal gap is unsolvable because there are no applicable operators, then each condition that cannot be removed is a source of the problem. If we detected that the causal gap is unsolvable because it is impossible, then there may be conditions that individually cause a problem along with ones that jointly cause a problem. For example, if two conditions are inconsistent with each other, then they jointly cause a problem.

For each condition that individually causes a problem, we must retract the operator that resulted in this condition being added to the causal gap.¹ For each set of conditions that jointly cause a problem, we must retract one of the operators that resulted in adding one of these conditions to the causal gap. One must also retract any decision where plan instance pi was introduced into the plan if the only reason that pi is in the plan is to achieve the executability conditions of another plan instance being removed during retraction. Secondly, we must remove any plan instance that achieves the non-interference conditions between a pair of plan instances, where one or both of the pair is being removed.

Partially Specified Plan Instances (Planning with Constraints)

One restrictive aspect of our simple planning algorithm is that each plan instance must be completely specified when it is entered into the plan. By "completely specified" we mean that the plan instance is represented by a ground term. This is in contrast to current planning systems, such as MOLGEN [Stefik 81], SIPE [Wilkins 83], and DEVISER [Vere 81], that permit features of an action, such as the objects manipulated by the action or the action's time of occurrence, to be only partially specified when the action is first entered into the plan. A

¹ If there is a condition causing a problem that is not entered by any operator, then it was entered as part of the goal, and consequently there is no solution to the planning problem

partially described action can be represented by a function term containing arguments whose values are just constrained to meet some property or relation, rather than arguments that are ground terms. As the planning process continues, additional constraints may be imposed on these values for such reasons as preventing the plan instances in the solution from conflicting with each other.

The use of partially described plan instances permits more flexible control strategies than those that can be employed using our simple algorithm since more choices are available at each decision point. This added flexibility allows one to better approximate the "least commitment control strategy" where one commits to detail only when necessary, trying to avoid premature decisions that are made before other requirements needed to solve the problem are known.

In order to extend and modify our algorithm to allow partially described plan instances, a number of issues must be considered. We keep in mind that we want to provide a mapping (as we did with our simple algorithm) that interprets the new algorithm as reaching conclusions that follow (in our logic) from knowledge about the problem given as input. Mapping the algorithm to the logic forces us to be precise as to what the algorithm is doing. For example, if the algorithm returns a partially described plan instance, we need to decide the relation between the solution and the goal. For example, does the mechanism return a partially described plan having the property that each of its completions solves the goal (and at least one completion exists), or does it return a partially described plan instance having the property that there exists a completion that solves the goal.

We must provide a syntax to describe partially described plan instances and must modify our action specifications so that executability conditions, effects, and non-interference conditions are given for partially described plan instances rather than for specific ones (which is currently done). For example, we may want an executability specification that captures: for any interval i that has duration D , the plan instance $ev@i$ is executable if property pr holds just prior to i . As another example, we may want a non-interference specification that captures that for any objects $o1$ and $o2$, and any intervals $i1$ and $i2$, $(grasp\ o1)@i1$ and $(grasp\ o2)@i2$ interfere only if $o1$ and $o2$ denote the same object and $i1$ and $i2$ overlap in time.

A syntax is also needed to describe partially described conditions appearing in the causal gap. Partially described conditions may be entered into the causal gap as the result of introducing a partially described plan instance. For example, the result of introducing the partially described plan instance $ev@i$ (described above) can be that the partially described condition " pr holds during an interval ix that is constrained to meet i to the right" is added to the causal gap.

The presence of partially described plan instances and conditions forces us to modify the *INTRO* and *REMOVE* planning operators. We must come up with a new characterization of when these operators are applicable. For example, we may say that *INTRO* is applicable using partially described plan instance *PI-PD* to remove partially described condition *C-PD* iff there is a completion of *PI-PD* (pi) and a completion of *C-PD* (C) such that C is derivable from the planning environment augmented with pi 's effects. This modified *INTRO* operator will require a more sophisticated mechanism to determine operator applicability. The use of partially described operators also complicates the process of backtracking from bad planning choices.

Finally, a third type of planning operator must be provided that adds additional constraints to arguments appearing in partially described plan instances and to arguments appearing in partially described conditions appearing in the causal gap. This operator may be performed when a partially described plan instance is being entered. For example, when introducing a

partially described plan instance *PI-PD* to remove a partially described condition *C-PD*, we may add constraints to *PI-PD* and *C-PD* so that any completion of *PI-PD* with the added constraints achieves any completion of *C-PD* with the added constraints.

A mechanism must also be provided that checks that the result of adding a new constraint does not lead to some partially described plan instance or condition having no completions. Typically, this has been determined by seeing if the set of constraints are satisfiable. In our system, however, satisfiability may not be sufficient because we are mapping our algorithm to a deductive logic, interpreting the property "there exists a completion", as conclusion with an existential quantifier. For example, while sentences capturing that "grasp object *o1* during *i*" and "grasp object *o2* during *i*" both occur, and *o1* and *o2* are constrained to be distinct may be satisfiable, this does not guarantee that there is a completion for each partially described plan instance meeting the constraint. In order to prove that such a completion exists, we would have to prove that at least two objects exist that can be manipulated by a grasping action.

Using Different Types of Input Specifications

The algorithm that we presented required that one specify the planning environment, the executability conditions and effects for each action, and the non-interference conditions for each pair of non-overlapping plan instances as the input to the planning algorithm. Some of these descriptions, however, may not be readily available or may require the user to perform complex computations to be determined. As an alternate, we might put the burden on the computer system and derive these relations from descriptions that may be more readily available.

Coming up with non-interference conditions may be particularly difficult. For example, determining whether a plan instance *pi* interferes with a plan instance that maintains property *pr* over interval *i* involves determining whether the occurrence of *pi* leads to *pr* not holding any time during interval *i*.

Instead of requiring the user to compute non-interference conditions, we may try to find other types of information from which these conditions can be derived. For example, one could compute that plan instances *pi1* and *pi2* interfere under all conditions (and thus, their non-interference conditions would be set to FALSE) if $\neg(\text{NOT}(\text{OCC } pi1) (\text{OCC } pi2))$ is derivable from the planning environment description. One however, could not use the planning environment alone to deductively derive all non-interference relations. For one, there is no information in the planning environment description (which consists only of interval logic statements in our current incarnation) enabling us to deductively derive that two overlapping plan instances do not interfere under any conditions.

To remedy the above problem, one might propose that one concludes that *pi1* and *pi2* interfere if and only if $\neg(\text{NOT}(\text{OCC } pi1) (\text{OCC } pi2))$ is not derivable from the planning environment description.² There are problems, however, with this approach. It might be the case that *pi1* and *pi2* interfere although $\neg(\text{NOT}(\text{OCC } pi1) (\text{OCC } pi2))$ is not derivable from the planning environment description. This case arises when the planning environment description is incomplete. Thus, to exploit this approach, we would have to make assumptions about the completeness of our planning environment description.

² This type of relation is not deductive, it could, however, be formulated using a non-monotonic logic

A more serious flaw is that this approach does not take into account plan instances that just conditionally interfere. In fact, if we do not extend the planning environment to include statements other than interval logic statements, it is impossible to detect conditional interference. For example, suppose that plan instances $pi1$ and $pi2$ interfere under conditions C . As we have shown in section 2.3, this is not detectable by seeing if $\neg(\text{IF } C (\text{NOT } (\text{OCC } pi1) (\text{OCC } pi2)))$ is derivable from the planning environment (even if we assume that the planning environment description is complete). This statement does not distinguish between the case where C is a condition that must hold in order for $pi1$ and $pi2$ not to interfere from the case where $\neg C$ is an effect of executing $pi1$ and $pi2$ together. Thus, if we want to derive non-interference conditions, we have to find a convenient way to augment the planning environment description to capture these type of distinctions.

7.2.2. Issues Outside the Scope of the Deductive Logic

In this section, we briefly look at some issues that are not fully addressed by developing a deductive logic to express planning problems. We first discuss the problem of deciding what possibilities to take into account when forming a plan. We then discuss planning with an incomplete description where it may be appropriate to suspend planning to obtain relevant information. Lastly, we mention the problem of planning with incorrect descriptions where planned actions might fail or not produce desired results. These discussions will elicit the role that the deductive logic can play in these reasoning tasks, demonstrating that the logic is not rendered obsolete when considering these more encompassing issues.

The Deductive planning Problem and a Simplified World Model

We have formulated the "deductive planning problem"; given a description of the world S and a goal G , we are looking for a plan instance pi such that it deductively follows from S that pi is executable and if pi occurs then G holds under all possible circumstances. One might argue that this framework is inappropriate or useless since an agent never comes up with a plan that works under all conceivable possibilities. This, however, is not the problem that is necessarily faced in this framework; the possibilities referred to are just the ones that are captured by our world description S . This description can represent a simplified view of the world that takes into account only the few possibilities that the agent is actively considering.

This is in line with our more encompassing view of planning: the task of finding a plan that is airtight with respect to the possibilities that are actively being considered. For example, if you tell me that my plan would not work if so and so happens, a typical response would be "Oh, I was not taking that into account" and would either argue that it should not be taken into account or would modify my plan to take it into account.

Identifying a deductive logic is essential for formalizing this more encompassing planning process since this formalism provides the notion of "being airtight". The other problem, that of determining the possibilities to take into account, however, is outside the scope of this work.

There has been some work that is relevant to this problem of coming up with a simplified world description. Of particular relevance, is any framework for handling the *qualification problem* [McCarthy 77]. This is the problem that typically, action precondition specifications and causal statements only take into account certain possibilities. For example, one might include a precondition specification for starting a car that mentions such things as the key

being in the ignition and there being gas in the tank. Typically, however, one would not mention the condition that there is no potato in the tailpipe, although this situation could conceivably arrive and prevent the car from starting. This suggests how we may describe simplified scenarios in our framework: our specifications of executability conditions, non-interference conditions, and causal statements (which we have been capturing using a material implication) need only mention conditions that one is actively considering.

There has been a number of schemes addressing the qualification problem that employ non-monotonic logics, such as [Ginsberg&Smith 87] and [Shoam 86]. In both these approaches, when describing a general domain theory that is to be used for a number of different problems, one explicitly mentions only the conditions that are considered *normal*, that is, conditions that are to be taken into account in all given problems. For a particular problem, one can also specify *abnormal conditions* that should be taken into account when solving this particular problem. When solving a problem, these systems take into account the normal conditions plus the specified abnormal conditions, ignoring all other conditions. Unfortunately, deciding which conditions to be considered as normal is a difficult problem which is outside the scope of these approaches.

Planning with an Incomplete Description

While planning at a particular time, the agent may be unable to find a plan to solve a given goal because it is missing relevant pieces of information about the external world. In our framework, this situation manifests when neither the truthhood or falsehood of a relevant condition follows from the description of a planning problem. Since our logic is deductive, the only ramification resulting from the presence of an incomplete description is that results may not be derived that could be if a more complete description were supplied. Thus, this framework is quite suitable for reasoning with incomplete information, as opposed to ones that mix in non-deductive schemes such as the persistence assumption [McDermott 82], which can lead to incorrect results when incomplete descriptions are used.

When one's knowledge is incomplete, it is necessary to determine which additional pieces of information are relevant and how or whether that information can be obtained before needed. For example, if there are two routes that can be taken to reach a destination, and one of them may be blocked, one would like to determine which route is blocked before reaching the point where the two routes diverge.

In order to reason about such examples in a static planning framework, it is necessary to represent the agent's knowledge of the world and actions that change the agent's knowledge state (along with representing the external world and actions that affect the external world). We can then introduce a "conditional plan instance" such as $\langle \text{if-else } \text{cnd } i \text{ } \pi i1 \text{ } \pi i2 \rangle$ which refers to executing $\pi i1$ if the agent determines that property cnd holds during interval i , while executing $\pi i2$ if the agent determines that cnd is false during i . This plan instance is executable only if the agent either knows, at a time prior to $\pi i1$'s and $\pi i2$'s execution times, that cnd is true during i or knows that cnd is false during i . Typically, this conditional plan instance would be used only if the agent does not know at planning time whether or not this condition holds. Thus, in order to make sure that the conditional plan instance $\langle \text{if-else } \text{cnd } i \text{ } \pi i1 \text{ } \pi i2 \rangle$ is executable, one would have to introduce a prior plan instance whose effect is that the agent knows whether cnd is true or false during i . For instance, in the example above, there may be sign where the routes diverge indicating which route is blocked. In this case, one would introduce a plan instance referring to reading the sign to determine which route is blocked.

An alternative to coming up with a conditional plan (that contains "branches" for the different possibilities that can result) is to suspend planning to gather relevant information. This option clearly would be favored in the case when a conditional plan would require many branches. In order to use this approach, however, one must determine which relevant information should be gathered. This reasoning task is similar to the one that would be encountered when using the conditional constructs. This is because relevant information refers to information that allows one to choose between different options (or determine that all of our options will not work) for achieving the current goal. Thus, we feel that a good starting point for dealing with incomplete knowledge is to extend our logic to incorporate i) information about the agent's knowledge state, ii) actions that produce knowledge, and iii) conditional plan instance constructs. A mechanism then would be needed that decided when to suspend planning to gather relevant information.

Relevant to the above issues is work presented by Moore [Moore 80], which describes a rigorous and principled framework for relating actions and knowledge. He integrated situation calculus and a possible worlds approach to knowledge enabling him to describe both the external world and one's knowledge of the world at each situation, along with actions that change the agent's state of knowledge. One may try to do adapt his method in our framework by extending our semantic structure to include an accessibility relation to interpret an epistemic modal operator (i.e., a modal operator for describing an agent's knowledge state). Some complications that would have to be considered are the interplay between epistemic possibility and temporal possibility (as captured by the *POS* modal operator) and the introduction of non-rigid designators (i.e., terms that can denote different objects in different possible worlds (world-histories)), which have a central role in Moore's theory.

Planning with Incorrect Descriptions

During planning time, one's view of the world may be incorrect, a fact that may be verified by later observations indicating such things as a planned action that could not be executed or an expected result which was not produced. A deductive logic, such as ours, can be used to provide a precise characterization of this conflict between one's view at planning time and later observations. Each of these conflicts can be formalized as an inconsistency between sentences describing one's view during planning time and sentences describing later observations.

One must be careful, however, when choosing an appropriate language for describing the world at planning time and expressing later observations. For example, suppose that a simplified world description is used (as discussed earlier) that does not take into account a condition *C* that actually turns out to ruin plan instance *pi*'s executability conditions. Since this world view does not even mention *C*, this description would not be inconsistent with the observation that *C* is true. Thus, to detect an inconsistency, one would either need an observation that *pi* was attempted but failed in conjunction with the simplified description, or an observation that *C* does not hold in conjunction with a more detailed world view that takes into account the relation between *C* and *pi*.

Along with detecting that a conflict exists, one wants to isolate the cause of the conflict and possibly use this information to patch up a plan to compensate for the unexpected occurrence or result. Isolating a conflict and patching up a plan, however, are tasks outside the scope of our logic. For example, suppose that we derive that condition *C* holds using a planning description and description about the plan that is going to be executed, but later determine that it does not or will not hold. In this case, it would be natural to look for the events or plan

instances that *cause* C , possibly under some expected circumstances, and then try to determine whether the event (plan instance) did not occur or whether the expected circumstances did not result. If we want to revise the plan to provide for the fact that C does not hold or will not hold, we would need to find the plan instances whose executability conditions or non-interference conditions depend on C being true. Thus, to isolate a conflict and patch plans, information is needed such as links between effects and their causes in conjunction with circumstances setting the context, and links between plan instances and their executability and non-interference conditions. These links cannot be expressed in the logic we have presented.

There are techniques that have been developed relevant to this task, such as Doyle's TMS system [Doyle 79] used to resolve inconsistencies and Ginsberg's belief revision scheme [Ginsberg 86] used to evaluate counterfactuals. A deficiency of these approaches, however, is that the results produced are sensitive to syntactic differences that may not reflect semantic ones. For example, consider the logically equivalent sets: $S1$, which consists of our non-modal axioms plus the set $\{\neg(\text{OCC}(\text{COMP } p1 \text{ } p2))\}$, and $S2$, which consists of our non-modal axioms plus the set $\{\neg(\text{OCC } p1), \neg(\text{OCC } p2)\}$. If we used either method to resolve the conflict produced by $\neg(\text{NOT}(\text{OCC } p1))$, we would get a different results depending on whether $S1$ or $S2$ is used. As a second example, both systems would produce different results depending on whether theory $\{A, B\}$ or theory $\{A, \neg(\text{IF } A \text{ } B)\}$ is used in a problem where we find that $\neg(\text{NOT } B)$ holds. Ginsberg mentions this result, but suggests that it is not a problem. We, however, favor an approach where the ontology is extended to capture the intended differences for cases where different results are warranted.¹

¹ We think that the two examples presented are manifestations of two different problems, the first dealing with the "granularity" of the assertions and the second with causal connections implicitly intended by the use of the material implication conditional

Bibliography

[Allen 83a]

Allen, J. F., Maintaining Knowledge about Temporal Intervals, *Communications of the ACM* 26.11 (1983), 832-843.

[Allen&Koomen 83b]

Allen, J. F. and Koomen, J. A., Planning Using a Temporal World Model, *8th International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, August 1983, 711-714.

[Allen 84]

Allen, J. F., Towards a General Theory of Action and Time, *Artificial Intelligence* 29,2 (1984), 123-154.

[Allen&Hayes 85]

Allen, J. F. and Hayes, P. J., A Common-Sense Theory of Time, *9th International Joint Conference on Artificial Intelligence*, Los Angeles, USA, August 1985.

[Bell 85]

Bell, C., Resource Management in Automated Planning, Working Paper Series No. 85-33, University of Iowa, Iowa City, August 1985.

[Chapman 85]

Chapman, D., Planning for Conjunctive Goals, *Artificial Intelligence* 32.3 (1987), 333-377.

[Doyle 79]

Doyle, J., A Truth Maintenance System, *Artificial Intelligence* 12,3 (1979), 231-272.

[Fikes&Nilsson 71]

Fikes, R. E. and Nilsson, N. J., STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving, *Artificial Intelligence* 2,3/4 (1971).

[Georgeff 86]

Georgeff, M., The Representation of Events in Multiagent Domains, *Proceedings of the National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, 70-75.

[Ginsberg 86]

Ginsberg, M. L., Counterfactuals, *Artificial Intelligence* 30(1986), 35-80.

[Ginsberg&Smith 87]

Ginsberg, M. L. and Smith, D. E., Possible Worlds and the Qualification Problem, *Proceedings of the National Conference on Artificial Intelligence*, Seattle, WA, August 1987, 212-217.

[Goldman 70]

Goldman, A. I., *A Theory of Human Action*. Prentice Hall, Englewood Cliffs, NJ, 1970.

[Haas 85]

Haas, A., Possible Events, Actual Events, and Robots, *Computational Intelligence* 1,2 (1985), 59-70.

[Hanks&McDermott 85]

Hanks, S. and McDermott, D., Temporal Reasoning and Default Logics, Computer Science Research Report No. 430, Yale University, October 1985.

[Hintikka 62]

Hintikka, J., *Knowledge and Belief*, Cornell University Press, Ithica, NY, 1962.

[Hughes&Cresswell 68]

Hughes, G. E. and Cresswell, M. J., *An Introduction to Modal Logic*. Methuen and Co. Ltd., London UK, 1968.

[Lansky 85]

Lansky, A. L., Behaviorial Specifications and Planning for Multiagent Domains, Technical Report 360, SRI International, Menlo Park, CA, August 1985.

[Lewis 73]

Lewis, D. K., *Counterfactuals*, Harvard University Press, Cambridge, MA, 1973.

[McCarthy&Hayes 69]

McCarthy, J. and Hayes, P., Some Philosophical Problems from the Standpoint of Artificial Intelligence, in *Machine Intelligence*, vol. 4, Michie, B. M. D. (editor), 1969, 463-502.

[McCarthy 77]

McCarthy, J., Epistemological Problems of Artificial Intelligence, *5th International Joint Conference on Artificial Intelligence*, Cambridge, USA, August 1977, 1038-1044.

[McCarthy 80]

McCarthy, J., Circumscription—A Form of Nonmonotonic Reasoning, *Artificial Intelligence* 19(1980), 27-39.

[McDermott&Doyle 80]

McDermott, D. and Doyle, J., Non-Monotonic Logic I, *Artificial Intelligence* 19(1980), 41-72.

[McDermott 82]

McDermott, D., A Temporal Logic for Reasoning about Process and Plans, *Cognitive Science* 6,2 (1982), 101-155.

[Moore 80]

Moore, R., Reasoning about Knowledge and Action, Technical Report 191, SRI International, Menlo Park, CA, August 1980.

[Pelavin&Allen 1986]

Pelavin, R. N. and Allen, J. F., A Formal Logic of Plans in Temporally Rich Domains, *Proceedings of the IEEE* 74.10 (October 1986), 1364-1382.

[Pollack 86]

Pollack, M. E., Inferring Domain Plans in Question-Answering, Ph.D. Dissertation, University of Pennsylvania, Philadelphia, PA, 1986.

[Prior 67]

Prior, A., *Past, Present, and Future*, Oxford University Press, Oxford UK, 1967.

[Reiter 80]

Reiter, R., A Logic for Default Reasoning, *Artificial Intelligence* 19(1980), 81-132.

[Sacerdoti 74]

Sacerdoti, E. D., Planning in a Hierarchy of Abstraction Spaces, *Artificial Intelligence* 5.2 (1974).

[Sacerdoti 77]

Sacerdoti, E. D., *A Structure for Plans and Behavior*, American Elsevier, 1977.

[Shoam 86]

Shoam, Y., Chronological Ignorance, *Proceedings of the National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, 389-393.

[Stalnaker 68]

Stalnaker, R., A Theory of Conditionals, in *Studies in Logical Theory*, Rescher, N. (editor), Basil Blackwell, Oxford, 1968, 98-112.

[Stefik 81]

Stefik, M., Planning with Constraints (MOLGEN: Part 1), *Artificial Intelligence* 16(1981), 111-140.

[Stickel 85]

Stickel, M. E., Automated Deduction by Theory Resolution, *9th International Joint Conference on Artificial Intelligence*, Los Angeles, USA, August 1985, 1181-1186.

[Stuart 86]

Stuart, C. J., A New View of Parallel Activity for Conflict Resolution, *Proceedings of 1986 Workshop on Reasoning about Actions and Plans*, Timberline, OR, 1986, 92-115.

[Tate 77]

Tate, A., Generating Project Networks, *5th International Joint Conference on Artificial Intelligence*, Cambridge, USA, August 1977.

[Tenenberq 86]

Tenenberg, J. T., Planning with Abstraction, *Proceedings of the National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, 76-80.

[Thomason 70]

Thomason, R. H., Indeterminist Time and Truth Value Gaps, *Theoria* 30(1970). 264-281.

[Vere 81]

Vere, S. A., Planning in Time: Windows and Durations for Activities and Goals. Research Report, Jet Propulsion Laboratory, Pasadena, CA, November 1981.

[Wilkins 83]

Wilkins, D. E., Domain Independent Planning: Representation and Plan Generation, Technical Note 266R, SRI International, Menlo Park, CA, May 1983.

Appendix A

The Primitive Language

Primitive symbols

The set of individual variables $VAR_{ol} = \{?v_1, ?v_2, \dots\}$

The set of function symbols $FN_{ol} = \{f_1, f_2, \dots\}$

The function $DEG: FN_{ol} \rightarrow Z$ (the set of non-negative integers). DEG yields the arity for each function symbol. If $DEG(f_i)=0$, then f_i is a constant.

The set of types $TYPES_{ol} = \{OBJ_{ol}, INT_{ol}, PROP_{ol}, EV_{ol}, PI_{ol}\}$

The function $TYPE-OF: VAR_{ol} \cup FN_{ol} \rightarrow TYPES_{ol}$. $TYPE-OF$ yields the type associated with each function symbol and variable.

The set of terms $TERMS_{ol}$ and the function $TYPE-OF^*$ that yields the type associated with each term are defined by:

For every variable $(?v_i)$, $?v_i \in TERMS_{ol}$
and $TYPE-OF^*(?v_i) =_{def} TYPE-OF(?v_i)$

For every function symbol (f_i) such that $DEG(f_i)=0$,
 $f_i \in TERMS_{ol}$
and $TYPE-OF^*(f_i) =_{def} TYPE-OF(f_i)$

For every function symbol (f_i) such that $DEG(f_i)=n>0$
and terms (t_1, t_2, \dots, t_n) , $\overline{(f_i t_1 t_2 \dots t_n)} \in TERMS_{ol}$
and $TYPE-OF^*(\overline{(f_i t_1 t_2 \dots t_n)}) =_{def} TYPE-OF(f_i)$

For all terms (t_i) and (t_j) such that $TYPE-OF^*(t_i)=PI_{ol}$
and $TYPE-OF^*(t_j)=PI_{ol}$, $\overline{(COMP t_i t_j)} \in TERMS_{ol}$
and $TYPE-OF^*(\overline{(COMP t_i t_j)}) =_{def} PI_{ol}$

For every term (t_i) such that $TYPE-OF^*(t_i)=PI_{ol}$,
 $\overline{(TIME-OF t_i)} \in TERMS_{ol}$
and $TYPE-OF^*(\overline{(TIME-OF t_i)}) =_{def} INT_{ol}$

The set of atomic formulas AF_{ol} :

For all terms (t_i and t_j), $\overline{[(= t_i t_j)]} \in AF_{ol}$

For all terms (t_i and t_j) such that $TYPE-OF^*(t_i) = INT_{ol}$
and $TYPE-OF^*(t_j) = INT_{ol}$, $\overline{[(MEETS t_i t_j)]} \in AF_{ol}$

For all terms (t_i and t_j) such that $TYPE-OF^*(t_i) = PROP_{ol}$
and $TYPE-OF^*(t_j) = INT_{ol}$, $\overline{[(HOLDS t_i t_j)]} \in AF_{ol}$

For all terms (t_i and t_j) such that $TYPE-OF^*(t_i) = EV_{ol}$
and $TYPE-OF^*(t_j) = INT_{ol}$, $\overline{[(OCCURS t_i t_j)]} \in AF_{ol}$

For every term (t_i) such that $TYPE-OF^*(t_i) = PI_{ol}$,
 $\overline{[(OCC t_i)]} \in AF_{ol}$

The set of well formed formulas WFF_{ol} :

For every atomic formula (af_i),
 $af_i \in WFF_{ol}$

For every well formed formula (P),
 $\overline{[(NOT P)]} \in WFF_{ol}$

For all well formed formulas (P and Q),
 $\overline{[(OR P Q)]} \in WFF_{ol}$

For every variable ($?v_i$) and well formed formula (P),
 $\overline{[(\forall ?v_i P)]} \in WFF_{ol}$

For every term (t_i) such that $TYPE-OF^*(t_i) = INT_{ol}$
and well formed formula (P), $\overline{[(INEV t_i P)]} \in WFF_{ol}$

For every term (t_i) such that $TYPE-OF^*(t_i) = PI_{ol}$
and well formed formula (P), $\overline{[(IFTRIED t_i P)]} \in WFF_{ol}$

Appendix B

Defined Symbols

Logical Symbols

$$(\text{AND } P \ Q) =_{\text{def}} (\text{NOT } (\text{OR } (\text{NOT } P) (\text{NOT } Q)))$$

$$(\exists v \ P) =_{\text{def}} (\text{NOT } (\forall v \ (\text{NOT } P)))$$

$$(\text{POS } i \ P) =_{\text{def}} (\text{NOT } (\text{INEV } i \ (\text{NOT } P)))$$

$$(\text{P-IFTRIED } i \ P) =_{\text{def}} (\text{NOT } (\text{IFTRIED } pi \ (\text{NOT } P)))$$

Interval Relation Predicates

$$\begin{aligned} (\text{BEFORE } i1 \ i2) =_{\text{def}} \\ (\text{EXISTS } ?i3 \\ (\text{AND } (\text{MEETS } i1 \ ?i3) (\text{MEETS } ?i3 \ i2))) \end{aligned}$$

$$(\text{AFTER } i1 \ i2) =_{\text{def}} (\text{BEFORE } i2 \ i1)$$

$$\begin{aligned} (\text{EQUAL } i1 \ i2) =_{\text{def}} \\ (\text{EXISTS } ?i3 \ ?i4 \\ (\text{AND } (\text{MEETS } ?i3 \ i1) (\text{MEETS } i1 \ ?i4) \\ (\text{MEETS } ?i3 \ i2) (\text{MEETS } i2 \ ?i4)))) \end{aligned}$$

$$\begin{aligned} (\text{OVERLAPS } i1 \ i2) =_{\text{def}} \\ (\text{EXISTS } ?i3 \ ?i4 \ ?i5 \ ?i6 \ ?i7 \\ (\text{AND } (\text{MEETS } ?i3 \ i1) (\text{MEETS } i1 \ ?i6) (\text{MEETS } ?i6 \ ?i7) \\ (\text{MEETS } ?i3 \ ?i4) (\text{MEETS } ?i4 \ i2) (\text{MEETS } i2 \ ?i7) \\ (\text{MEETS } ?i4 \ ?i5) (\text{MEETS } ?i5 \ ?i6)))) \end{aligned}$$

$$(\text{OVERLAPPED-BY } i1 \ i2) =_{\text{def}} (\text{OVERLAPS } i2 \ i1)$$

$$\begin{aligned} (\text{STARTS } i1 \ i2) =_{\text{def}} \\ (\text{EXISTS } ?i3 \ ?i4 \ ?i5 \\ (\text{AND } (\text{MEETS } ?i3 \ i1) (\text{MEETS } i1 \ ?i4) (\text{MEETS } ?i4 \ ?i5) \\ (\text{MEETS } ?i3 \ i2) (\text{MEETS } i2 \ ?i5)))) \end{aligned}$$

Interval Relation Predicates (Cont.)

(STARTED-BY i1 i2) =_{def} (STARTS i2 i1)

(FINISHES i1 i2) =_{def}
 (EXISTS ?i3 ?i4 ?i5
 (AND (MEETS ?i3 ?i4) (MEETS ?i4 i1) (MEETS i1 ?i5)
 (MEETS ?i3 i2) (MEETS i2 ?i5)))

(FINISHED-BY i1 i2) =_{def} (FINISHES i2 i1)

(DURING i1 i2) =_{def}
 (EXISTS ?i3 ?i4 ?i5 ?i6
 (AND (MEETS ?i3 ?i4) (MEETS ?i4 i1)
 (MEETS i1 ?i5)(MEETS ?i5 ?i6)
 (MEETS ?i3 i2) (MEETS i2 ?i6)))

(CONTAINS i1 i2) =_{def} (DURING i2 i1)

(ENDS= i1 i2) =_{def}
 (EXISTS ?i3
 (AND (MEETS i1 ?i3) (MEETS i2 i3)))

(ENDS< i1 i2) =_{def}
 (EXISTS ?i3 ?i4
 (AND (MEETS i1 ?i3) (MEETS ?i3 ?i4) (MEETS i2 i3)))

(ENDS≤ i1 i2) =_{def}
 (OR (ENDS= i1 i2) (ENDS< i1 i2))

(IN i1 i2) =_{def}
 (OR (STARTS i1 i2) (EQUALS i1 i2)
 (DURING i1 i2) (FINISHES i1 i2))

(PRIOR i1 i2) =_{def}
 (OR (MEETS i1 i2) (BEFORE i1 i2))

(DISJOINT i1 i2) =_{def}
 (OR (PRIOR i1 i2) (PRIOR i2 i1))

Appendix C

The Semantic Model

A model is a tuple $\langle H, I, OBJ, PROP, EV, PI, MTS, R, BA, BAEV, V_s, V_{vf} \rangle$ where

- H** is a non-empty set designating the world-histories (i.e., complete worlds over time)
- I** is a non-empty set designating the temporal intervals (i.e., stretches of time in a date line)
- OBJ** is a non-empty set of physical objects
- PROP** is a non-empty subset of $2^{I \times H}$ designating the properties (i.e., static conditions). A property is equated with the interval/world-history pairs over which it holds.
- EV** is a non-empty subset of $2^{I \times H}$ designating the events. An event is equated with the interval/world-history pairs over which it occurs.
- PI** is a non-empty subset of $2^{EV \times I} \times 2^{BA \times I}$ designating the plan instances. A plan instance pi is associated with a set of event instances (i.e., event/interval pairs) ei -set and a set of basic action instances (i.e., basic action/interval pairs) bai -set; attempting pi refers to executing the basic action instances in bai -set; pi is successfully executed (i.e., pi occurs) iff all the basic action instances in bai -set are executed and all the event instances in ei -set occur.
- MTS** is a relation defined on $I \times I$ specifying the *meets* interval relation; $MTS(i1, i2)$ means that interval $i1$ immediately precedes interval $i2$.
- R** is a relation defined on $I \times H \times H$; $R(i, h1, h2)$ means that world-histories $h1$ and $h2$ are possible with respect to each other and share a common past through the end of interval i .
- BA** is a non-empty set of basic actions where each element is a function from $I \times H$ to 2^H ; for a basic action ba , the members of $ba(i, h)$ constitute all the world-histories that minimally differ from h on the account of executing basic action ba during interval i .
- BAEV** is a function from BA to EV specifying the event associated with each basic action; basic action ba is executed during interval i in world-history h iff event $BAEV(ba)$ occurs during interval i in world-history h .
- V_s** is a function from $WFF_{ol} \times H$ to $\{TRUE, FALSE\}$ specifying the truth-value of each well formed formula at each world-history.
- V_{vf}** is a function from $TERMS_{ol}$ to functions with domain D^n and range D specifying the denotation of each variable term, constant term, and function symbol.

where $D =_{def} I \cup OBJ \cup PROP \cup EV \cup PI$ and n is a non-negative integer

We make use the following definitions in specifying the constraints on the model structure

$BEG-BEF(i1,i2) =_{def}$ there exists intervals ($i0$ and $i0'$)
such that $MTS(i0,i1)$ and $MTS(i0,i0')$ and $MTS(i0',i2)$

$BEG-SAME(i1,i2) =_{def}$ there exists an interval ($i0$)
such that $MTS(i0,i1)$ and $MTS(i0,i2)$

$ENDS-BEF(i1,i2) =_{def}$ there exists intervals ($i3$ and $i3'$)
such that $MTS(i2,i3')$ and $MTS(i3,i3')$ and $MTS(i1,i3)$

$ENDS-SAME(i1,i2) =_{def}$ there exists an interval ($i3$)
such that $MTS(i1,i3)$ and $MTS(i2,i3)$

$IN(i1,i2) =_{def}$ ($BEG-BEF(i2,i1)$ or $BEG-SAME(i2,i1)$) and
($ENDS-BEF(i1,i2)$ or $ENDS-SAME(i1,i2)$)

$STARTS(i1,i2) =_{def}$ $BEG-SAME(i1,i2)$ and $ENDS-BEF(i1,i2)$

$EQUAL(i1,i2) =_{def}$ $BEG-SAME(i1,i2)$ and $ENDS-SAME(i1,i2)$

$COVER(i-set) =_{def}$ the interval (i) having the property
For every interval (ix), if $ix \in i-set$ then $IN(i,ix)$
and there exists intervals (ix and iy) such that
 $iy \in i-set$ and $BEG-SAME(i,iy)$ and
 $iz \in i-set$ and $ENDS-SAME(i,iy)$

Constraints on the model structure

MT1)

For all intervals ($i1$, $i2$, $i3$, and $i4$)
if $MTS(i1,i3)$ and $MTS(i2,i3)$
then $MTS(i1,i4)$ iff $MTS(i2,i4)$

MTS2)

For all intervals ($i1$, $i2$, $i3$, and $i4$)
if $MTS(i1,i2)$ and $MTS(i1,i3)$
then $MTS(i0,i2)$ iff $MTS(i0,i3)$

MTS3)

For all intervals ($i1$, $i2$, $i3$, and $i4$)
if $MTS(i1,i2)$ and $MTS(i3,i4)$
then one of the following is true
 $MTS(i1,i4)$ or
there exists an interval (ix)
such that $MTS(i1,ix)$ and $MTS(ix,i4)$ or
there exists an interval (iy)
such that $MTS(i3,iy)$ and $MTS(iy,i2)$

MTS4)

For all intervals (i1)
there exists intervals (i0 and i2)
such that MTS(i0,i1) and MTS(i1,i2)

MTS5)

For all intervals (i1 and i2)
if MTS(i1,i2) then there exists intervals (ix, iy, and iz)
such that MTS(ix,i1) and MTS(i2,iy) and MTS(ix,iz) and MTS(iz,iy)

PROP1)

For all properties (pr), intervals (i1 and i2),
and world-histories (h),
If IN(i1,i2) and $\langle i2, h \rangle \in pr$ then $\langle i1, h \rangle \in pr$

R1)

For all world-histories (h1 and h2) and intervals (i1 and i2),
if ENDS-SAME(i1,i2) then $R(i1, h1, h2)$ iff $R(i2, h1, h2)$

R2)

For all world-histories (h1 and h2) and intervals (i),
if $R(i, h1, h2)$ then $R(i, h2, h1)$

R3)

For all world-histories (h1, h2, and h3) and intervals (i),
if $R(i, h1, h2)$ and $R(i, h2, h3)$ then $R(i, h1, h3)$

R4)

For all world-histories (h1 and h2) and intervals (i1 and i2),
if ENDS-BEF(i1,i2) and $R(i2, h1, h2)$ then $R(i1, h1, h2)$

R5)

For all world-histories (h1 and h2), properties (pr),
and intervals (i1 and i2),
if (ENDS-SAME(i1,i2) or ENDS-BEF(i1,i2)) and $R(i2, h1, h2)$ then
 $\langle i1, h1 \rangle \in pr$ iff $\langle i1, h2 \rangle \in pr$

R6)

For all world-histories (h1 and h2), events (ev),
and intervals (i1 and i2),
if (ENDS-SAME(i1,i2) or ENDS-BEF(i1,i2)) and $R(i2, h1, h2)$ then
 $\langle i1, h1 \rangle \in ev$ iff $\langle i1, h2 \rangle \in ev$

PI1)

For all plan instances ($\langle eiS, baiS \rangle$),
STARTS(bai-time,ei-time) or EQUAL(bai-time,ei-time)
where ei-time =_{def} COVER($\{i \mid \langle ev, i \rangle \in eiS\}$)
bai-time =_{def} COVER($\{i \mid \langle ba, i \rangle \in baiS\}$)

PI2)

For all plan instances ($\langle eiS1, baiS1 \rangle$ and $\langle eiS2, baiS2 \rangle$),
 $\langle eiS1 \cup eiS2, baiS1 \cup baiS2 \rangle \in PI$

BA0)

For every basic action (ba), interval (i), and world-history (h),
 $ba(i, h) \neq \emptyset$

BA1)

For every basic action (ba), interval (i), and world-history (h),
 if $\langle i, h \rangle \in BAEV(ba)$ then $ba(i, h) = \{h\}$

BA2)

For every basic action (ba), interval (i),
 and world-histories (h and h2),
 if $h \neq h2$ and $h2 \in ba(i, h)$ then $\langle i, h2 \rangle \in BAEV(ba)$

BA-R1)

For all world-histories (h and h2), basic actions (ba),
 and intervals (i0 and i),
 if $h2 \in ba(i, h)$ and $MTS(i0, i)$ then $R(i0, h, h2)$

BA-R2)

For all world-histories (h1 and h2), intervals (i and ir),
 and basic actions (ba),
 if $R(ir, h1, h2)$ then for all world-histories (hcl1) such that $hcl1 \in ba(i, h1)$, then there exists a
 world-history (hcl2) such that $hcl2 \in ba(i, h2)$ and $R(ir, hcl1, hcl2)$ are true

We make use the following definitions:

$$fx;fy(h) =_{def} \bigcup_{hx \in fx(h)} fy(hx)$$

$$SEQF(\{\langle ba, i \rangle\}) =_{def} \{\lambda h. ba(i, h)\}$$

$$SEQF(baiS \cup \{\langle ba2, i \rangle\}) =_{def}$$

$$\{seq; \lambda h. ba2(i, h) \mid seq \in SEQF(baiS)\}$$

where $|baiS| \geq 1$

$$ALL-OC(h, seq, baiS) =_{def}$$

$$seq \in SEQF(baiS) \text{ and }$$

$$seq(h) \subseteq \{h2 \mid \langle i, h2 \rangle \in BAEV(ba) \text{ for all } \langle ba, i \rangle \in baiS\}$$

FCL-DEF)

If there exists a sequence function (seq) such that $ALL-OC(h, seq, baiS)$,
 then $F_{cl}(baiS, h) =_{def} seq(h)$

Otherwise, $F_{cl}(baiS, h) =_{def} \{h\}$

BA-CMP1)

For all world-histories (h), basic action instance sets (baiS),
and sequence functions (seq1 and seq2)
if ALL-OC(h,seq1.baiS) and ALL-OC(h,seq2.baiS)
then seq1(h)=seq2(h)

BA-CMP2)

For all world-histories (h), basic action instance sets
(baiS1 and baiS2) and sequence functions (seq1 and seq2).
if ALL-OC(h,seq2.baiS2) and
ALL-OC(h,seq1:seq2.baiS1 \cup baiS2)
then ALL-OC(h,seq2;seq1.baiS1 \cup baiS2)

The interpretation functions

We make use of the following defined function:

TYP-MAP(OBJ_{ol}) =_{def} OBJ
TYP-MAP(INT_{ol}) =_{def} I
TYP-MAP(PROP_{ol}) =_{def} PROP
TYP-MAP(EV_{ol}) =_{def} EV
TYP-MAP(PI_{ol}) =_{def} PI

The denotation function for primitive symbols (V_{vf})

For every variable (?v), $V_{vf}(?v) \in \text{TYP-MAP}(\text{TYPE-OF}(?v))$

For every function symbol (f) such that DEG(f)=0,
 $V_{vf}(?vi) \in \text{TYP-MAP}(\text{TYPE-OF}(f))$

For every function symbol (f) such that DEG(f)>0,
 $V_{vf}(f)$ is a function with doman $D^{\text{DEG}(f)}$ and
range TYP-MAP(TYPE-OF(f))

where $D =_{\text{def}} I \cup \text{OBJ} \cup \text{PROP} \cup \text{EV} \cup \text{PI}$

The definition of the the denotation function for terms (V_t)

For every variable (?v), $V_t(?v) =_{\text{def}} V_{vf}(?v)$

For every function symbol (f) such that DEG(f)=0,
 $V_t(f) =_{\text{def}} V_{vf}(f)$

For every function symbol (f) such that DEG(f)=n>0
and terms (t_1, t_2, \dots, t_n),
 $V_t(\overline{(f \ t_1 \ t_2 \ \dots \ t_n)}) =_{\text{def}} V_{vf}(f)(V_t(t_1), V_t(t_2), \dots, V_t(t_n))$

The interpretation function for well formed formulas (V_s)

For all wffs of the form $\overline{[(= t_1 t_2)]}$ and world-histories (h),
 $V_s(\overline{[(= t_1 t_2)]}, h) = \text{TRUE}$ iff $V_t(t_1) = V_t(t_2)$

For all wffs of the form $\overline{[(OR P1 P2)]}$ and world-histories (h),
 $V_s(\overline{[(OR P1 P2)]}, h) = \text{TRUE}$ iff $V_s(P1, h) = \text{TRUE}$ or $V_s(P2, h) = \text{TRUE}$

For all wffs of the form $\overline{[(NOT P)]}$ and world-histories (h),
 $V_s(\overline{[(NOT P)]}, h) = \text{TRUE}$ iff $V_s(P, h) \neq \text{TRUE}$

For all wffs of the form $\overline{[(\forall ?v P)]}$ and world-histories (h),
 $V_s(\overline{[(\forall ?v P)]}, h) = \text{TRUE}$ iff for all objects (x)
 if $x \in \text{TYP-MAP}(\text{TYPE-OF}(?v))$, then $V_{s[\overline{?v, x}]}(P, h) = \text{TRUE}$
 where $V_{s[\overline{?v, x}]}$ is identical to V_s with the exception that $V_{t[\overline{?v, x}]}(?v) = x$

For all wffs of the form $\overline{[(\text{MEETS } t_{int1} t_{int2})]}$ and world-histories (h),
 $V_s(\overline{[(\text{MEETS } t_{int1} t_{int2})]}, h) = \text{TRUE}$ iff $\text{MTS}(V_t(t_{int1}), V_t(t_{int2}))$

For all wffs of the form $\overline{[(\text{HOLDS } t_{pr} t_{int})]}$ and world-histories (h),
 $V_s(\overline{[(\text{HOLDS } t_{pr} t_{int})]}, h) = \text{TRUE}$ iff $\langle V_t(t_{int}), h \rangle \in V_t(t_{pr})$

For all wffs of the form $\overline{[(\text{OCCURS } t_{ev} t_{int})]}$
 and world-histories (h),
 $V_s(\overline{[(\text{OCCURS } t_{ev} t_{int})]}, h) = \text{TRUE}$ iff $\langle V_t(t_{int}), h \rangle \in V_t(t_{ev})$

For all wffs of the form $\overline{[(\text{OCC } t_{pi})]}$ and all world-histories (h),
 $V_s(\overline{[(\text{OCC } t_{pi})]}, h) = \text{TRUE}$
 iff for all events (ev), intervals (i) and basic actions (ba)
 if $\langle ev, i \rangle \in V_t(t_{pi})|_1$, then $\langle i, h \rangle \in ev$ and
 if $\langle ba, i \rangle \in V_t(t_{pi})|_2$, then $\langle i, h \rangle \in \text{BAEV}(ba)$

For all wffs of the form $\overline{[(\text{INEV } t_{int} P)]}$ and world-histories (h),
 $V_s(\overline{[(\text{INEV } t_{int} P)]}, h) = \text{TRUE}$ iff for all world-histories (h2)
 if $R(V_t(t_{int}), h, h2)$ then $V_s(P, h2) = \text{TRUE}$

For all wffs of the form $\overline{[(\text{IFTRIED } t_{pi} P)]}$ and world-histories (h),
 $V_s(\overline{[(\text{IFTRIED } t_{pi} P)]}, h) = \text{TRUE}$ iff for all world-histories (h2)
 if $h2 \in F_{cl}(V_t(t_{pi})|_2, h)$ then $V_s(P, h2) = \text{TRUE}$

For all plan instance terms (t_{pi}),
 $V_t(\overline{[(\text{TIME-OF } t_{pi})]}) = \text{COVER}(\{i \mid \langle ev, i \rangle \in V_t(t_{pi})|_1\})$

For all plan instance terms (t_{pi1} and t_{pi2}),
 $V_t(\overline{[(\text{COM" } P t_{pi1} t_{pi2})]}) = \langle V_t(t_{pi1})|_1 \cup V_t(t_{pi2})|_1, V_t(t_{pi1})|_2 \cup V_t(t_{pi2})|_2 \rangle$

Appendix D

The Axiomatics

Axioms:

AX-FO1)

$\vdash (\text{IF } (\text{OR } P \ P) \ P)$

AX-I O2)

$\vdash (\text{IF } Q \ (\text{OR } P \ Q))$

AX-FO3)

$\vdash (\text{IF } (\text{OR } P \ Q) \ (\text{OR } Q \ P))$

AX-FO4)

$\vdash (\text{IF } (\text{IF } Q \ R) \ (\text{IF } (\text{OR } P \ Q) \ (\text{OR } P \ R)))$

AX-FO5)

$\vdash (\text{IF } (\forall ?v \ P1) \ P2)$

where P2 differs from P1 in having all free occurrences of ?v in P1 replaced by some term t that has the same type as variable ?v, and if term t has any variables in it, then they must not become bound by the substitution

AX-FO6)

$\vdash (= t \ t)$

AX-FO7)

$\vdash (\text{IF } (= t1 \ t2) \ (\text{IF } P1 \ P2))$

where P2 differs from P1 in having one or more free occurrences of t1 in P1 replaced by t2, and if term t2 has any variables in it, then they must not become bound

AX-FO8)

$\vdash (\text{NOT } (= t1 \ t2))$

where t1 and t2 have different types

AX-IL1)

$\vdash (\text{IF } (\text{IN } i1 \ i2) (\text{IF } (\text{HOLDS } pr \ i2) (\text{HOLDS } pr \ i1))))$

AX-IL2)

$\vdash (= (\text{COMP } pi1 \ pi1) \ pi1)$

AX-IL3)

$\vdash (= (\text{COMP } pi1 \ pi2) (\text{COMP } pi2 \ pi1))$

AX-IL4)

$\vdash (= (\text{COMP } pi1 (\text{COMP } pi2 \ pi3)) (\text{COMP } (\text{COMP } pi1 \ pi2) \ pi3))$

AX-IL5)

$\vdash (\text{IFF } (\text{OCC } (\text{COMP } pi1 \ pi2)) (\text{AND } (\text{OCC } pi1) (\text{OCC } pi2))))$

AX-IL6)

$\vdash (\text{IFF } (\text{AND } (\text{IN } (\text{TIME-OF } pi1) \ i) (\text{IN } (\text{TIME-OF } pi2) \ i))$
 $(\text{IN } (\text{TIME-OF } (\text{COMP } pi1 \ pi2)) \ i))$

AX-IR1)

$\vdash (\text{IF } (\text{AND } (\text{MEETS } i1 \ i3) (\text{MEETS } i2 \ i3))$
 $(\text{IFF } (\text{MEETS } i1 \ i4) (\text{MEETS } i2 \ i4))))$

AX-IR2)

$\vdash (\text{IF } (\text{AND } (\text{MEETS } i1 \ i2) (\text{MEETS } i1 \ i3))$
 $(\text{IFF } (\text{MEETS } i0 \ i2) (\text{MEETS } i0 \ i3))))$

AX-IR3)

$\vdash (\text{IF } (\text{AND } (\text{MEETS } i1 \ i2) (\text{MEETS } i3 \ i4))$
 $(\text{XOR } (\text{MEETS } i1 \ i4)$
 $(\exists ?ix (\text{AND } (\text{MEETS } i1 \ ?ix) (\text{MEETS } ?ix \ i4))$
 $(\exists ?iy (\text{AND } (\text{MEETS } i3 \ ?iy) (\text{MEETS } ?iy \ i2))))))$

where $(\text{XOR } P \ Q \ R) =_{\text{def}} (\text{OR } (\text{AND } (\text{NOT } P) \ Q \ R))$
 $(\text{AND } P \ (\text{NOT } Q) \ R)$
 $(\text{AND } P \ Q \ (\text{NOT } R)))$

AX-IR4)

$\vdash (\exists ?i0 \ ?i2 (\text{AND } (\text{MEETS } ?i0 \ i1) (\text{MEETS } i1 \ ?i2))))$

AX-IR5)

$\vdash (\text{IF } (\text{MEETS } i1 \ i2)$
 $(\exists ?ix \ ?iy \ ?iz$
 $(\text{AND } (\text{MEETS } ?ix \ i1) (\text{MEETS } i2 \ ?iy)$
 $(\text{MEETS } ?ix \ ?iz) (\text{MEETS } ?iz \ ?iy))))$

AX-INV1)

⊢ (IF (INEV i P)
P)

AX-INV2)

⊢ (IF (INEV i (IF P Q))
(IF (INEV i P) (INEV i Q)))

AX-INV3)

⊢ (IF (INEV i P)
(INEV i (INEV i P)))

AX-INV4)

⊢ (IF (POS i P)
(INEV i (POS i P)))

AX-INV5)

⊢ (IF (ENDS ≤ i1 i2)
(IF (INEV i1 P)
(INEV i2 P)))

AX-INV6)

⊢ (IF (ENDS ≤ i1 i2)
(IF (POS i2 (HOLDS pr i1))
(INEV i2 (HOLDS pr i1))))

AX-INV7)

⊢ (IF (ENDS ≤ i1 i2)
(IF (POS i2 (OCCURS ev i1))
(INEV i2 (OCCURS ev i1))))

AX-INV8)

⊢ (IF (ENDS ≤ (TIME-OF pi) i)
(IF (POS i (OCC pi))
(INEV i (OCC pi))))

AX-INV9)

⊢ (IF (POS i (MEETS i1 i2))
(INEV i (MEETS i1 i2)))

AX-IFTR1)

⊢ (IF (IFTRIED pi P)
(NOT (IFTRIED pi (NOT P))))

AX-IFTR2)

⊢ (IF (IFTRIED pi (IF P Q))
(IF (IFTRIED pi P) (IFTRIED i Q)))

AX-IFTR3)

⊢ (IF (OCC pi)
(IFF (IFTRIED pi P) P))

AX-IFTR4)

⊢ (IFF (IFTRIED pi (IFTRIED pi P))
(IFTRIED pi P))

AX-IFTR5)

⊢ (IF (PRIOR i (TIME-OF pi))
(IF (INEV i P)
(IFTRIED pi P)))

AX-IFTR6)

⊢ (IF (IFTRIED pi (INEV i P)))
(INEV i (IFTRIED pi P)))

AX-IFTR7)

⊢ (IF (IFTRIED pi1 (IFTRIED pi2 (AND (OCC pi1) (OCC pi2))))
(IFF (IFTRIED (COMP pi1 pi2) P)
(IFTRIED pi1 (IFTRIED pi2 P))))

AX-IFTR8)

⊢ (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
(IF (EXECUTABLE (COMP pi1 pi2))
(AND (EXECUTABLE pi1)
(IFTRIED pi (EXECUTABLE pi2)))))

AX-IFTR9)

⊢ (IF (EXECUTABLE pi1)
(IF (IFTRIED pi2 (IFTRIED pi1 (AND (OCC pi1)
(OCC pi2)))))
(IFTRIED pi1 (IFTRIED pi2 (AND (OCC pi1)
(OCC pi2)))))

Inference Rules:

MP)

From: $S1 \vdash P$ and $S2 \vdash (IF P Q)$

To: $S1 \cup S2 \vdash Q$

where $S1$ and $S2$ are any finite set of sentences, and P and Q are any sentences

UNV-INTRO)

From: $S \vdash (IF P Q)$

To: $S \vdash (IF P (\forall ?v Q))$

where S is a finite set of sentences, P and Q are any sentences, $?v$ is a variable, and there does not exist any free occurrences of $?v$ in P or any member of S

RL-INV)

From: $\vdash P$

To: $\vdash (INEV i P)$

RL-IFTR)

From: $\vdash P$

To: $\vdash (IFTRIED pi P)$

Appendix E

Auxiliary Theorems and Derived Rules

This appendix contains first order logic theorems and derived rules and interval relation theorems that are used in the proofs in appendices F, G, and H.

First order logic theorems and derived rules

Discharging single assumption

From: $S \cup \{A\} \vdash P$

To: $S \vdash (\text{IF } A \text{ } P)$

where S is a finite set of sentences, and A and P are sentences

Discharging multiple assumptions

From: $S \cup \{A_1, A_2, \dots, A_n\} \vdash P$

To: $S \vdash (\text{IF } (\text{AND } A_1 A_2 \dots A_n) P)$

where S is a finite set of sentences, and each A_i is a sentence

TRNSP)

$\vdash (\text{IF } (\text{IF } P \text{ } Q) (\text{IF } (\text{NOT } Q) (\text{NOT } P)))$

NNP-P)

$\vdash (\text{IFF } (\text{NOT } (\text{NOT } P)) P)$

MP-TRNS)

From: $S1 \vdash (\text{IF } P \text{ } Q)$ and $S2 \vdash (\text{IF } Q \text{ } R)$

To: $S1 \cup S2 \vdash (\text{IF } P \text{ } R)$

AND-ELIM-1)

$\vdash (\text{IF } (\text{AND } P \text{ } Q) Q)$

AND-ELIM-2)

$\vdash (\text{IF } (\text{AND } P \text{ } Q) P)$

EXT-INTRO)

From: $S \vdash (\text{IF } P \text{ } Q)$

To: $S \vdash (\text{IF } (\exists ?v P) Q)$

where $?v$ does not appear free in Q or any member of S

AND-MI)

From: $S1 \vdash (\text{IF } P1 \text{ } Q1)$ and $S2 \vdash (\text{IF } P2 \text{ } Q2)$

To: $S1 \cup S2 \vdash (\text{IF } (\text{AND } P1 \text{ } P2) (\text{AND } (Q1 \text{ } Q2)))$

First order logic theorems and derived rules (Cont.)

AND-INTRO)

From: $S1 \vdash P$ and $S2 \vdash Q$
 To: $S1 \cup S2 \vdash (AND P Q)$

RCS)

From: $S1 \vdash (OR P1 P2)$ and $S2 \vdash (IF P1 Q1)$ and $S3 \vdash (IF P2 Q2)$
 To: $S1 \cup S2 \cup S3 \vdash (OR Q1 Q2)$

SUBST)

From: $\vdash (IFF P Q)$
 To: $\vdash (IFF R1 R2)$
 where $R2$ differs from $R1$ in that one or more occurrences of P in $R1$ are replaced by Q .
 and no occurrences of P in $R1$ are under the scope of a modal operator

ANTCD-INTRO)

From: $S1 \vdash P$
 To: $S2s-2 \cup S1 \vdash P$

DEM1)

$\vdash (IFF (NOT (AND P Q)) (OR (NOT P) (NOT Q)))$

DEM2)

$\vdash (IFF (NOT (OR P Q)) (AND (NOT P) (NOT Q)))$

UNV-MI)

$\vdash (IF (IF P Q) (IF (\forall^?x P) (\forall^?x Q)))$

EXT-MI)

$\vdash (IF (IF P Q) (IF (\exists^?x P) (\exists^?x Q)))$

UNV-EXT)

$\vdash (IFF (\forall^?x (NOT P)) (NOT (\exists^?x P)))$

Interval relation theorems

TH-IR1)

$$\vdash (\text{IF } (\text{MEETS } i1 \ i2) (\text{PRIOR } i1 \ i2))$$

TH-IR2)

$$\vdash (\text{IF } (\text{PRIOR } i1 \ iy) (\text{IF } (\text{MEETS } i \ iy) (\text{ENDS} \leq i1 \ i)))$$

TH-IR3)

$$\{(\text{PRIOR } i1 \ iy), \dots, (\text{PRIOR } in \ iy)\} \\ \vdash (\text{IF } (\text{MEETS } i \ iy) \\ (\text{AND } (\text{ENDS} \leq i1 \ i) \dots (\text{ENDS} \leq in \ i)))$$

TH-IR4)

$$\vdash (\text{IF } (\text{AND } (\text{IN } ip \ i2) (\text{MEETS } i \ i2)) (\text{PRIOR } i \ ip))$$

TH-IR5)

$$\vdash (\text{IF } (\text{AND } (\text{PRIOR } i \ ip1) (\text{PRIOR } i \ ip2)) \\ (\exists \ ?i2 (\text{AND } (\text{MEETS } i \ ?i2) (\text{IN } ip1 \ ?i2) (\text{IN } ip2 \ ?i2))))$$

TH-IR6)

$$\vdash (\text{ENDS} \leq i \ i)$$

TH-IR7)

$$\vdash (\text{IN } i \ i)$$

TH-IR8)

$$\vdash (\text{IF } (\text{AND } (\text{PRIOR } i1 \ i2) (\text{IN } i3 \ i2)) (\text{PRIOR } i1 \ i3))$$

Appendix F

Proof of Theorems in Chapter 4

We use the following conventions in this appendix and in appendices G and H to present the proof of theorems and derived rules.

Proof form for theorems

A proof of theorem TH is given in the following form:

1) PL_1
 $\langle JS_1 \rangle$

2) PL_2
 $\langle JS_2 \rangle$

n) TH
 $\langle JS_n \rangle$

We refer to each PL_i as a *proof line* and each JS_i as the *justification for PL_i* .

Each proof line PL_i refers to a theorem or axiom in the logic. Each justification JS_i specifies the name of the axiom or theorem (if it has been proven elsewhere) that PL_i corresponds to, or specifies the inference rule(s) (possibly derived) along with the axioms and theorems that it is applied to yielding the theorem associated with PL_i . JS_i may mention theorems that are on earlier lines in the current proof; these theorems are identified by their line numbers in the proof.

The different forms of a proof line PL_i and the theorem that each form stands for is given as follow:

P_i	stands for	$\{P_i\} \vdash P_i$
$\vdash P_i$	" "	$\vdash P_i$
$S \vdash P_i$	" "	$S \vdash P_i$
$LN_1, LN_2, \dots, LN_j \vdash P_i$	" "	$\{P_x \mid x \in \{LN_1, LN_2, \dots, LN_j\}\} \vdash P_i$
$S \cup LN_1, LN_2, \dots, LN_j \vdash P_i$	" "	$S \cup \{P_x \mid x \in \{LN_1, LN_2, \dots, LN_j\}\} \vdash P_i$

where P_i is a sentence in the logic, LN_1, LN_2, \dots, LN_j are positive integers, each less than i (referring to proof line numbers), S is a non-empty finite set of sentences in the logic, and each P_x refers to the sentence in proof line x appearing on the right side of the turnstile operator (or the proof line itself, if proof line x has no turnstile)

Proof Form for Derived Rules

The proofs for derived inference rules have the same form as proofs for theorems with the exception that

- i) There are proof lines of the form " $\vdash P$ " and " $S \vdash P$ " that are not theorems: they are just assumptions made for the proof of the rule. If this is the case, the proof line's justification specifies "assumption for proof of rule".
- ii) A proof line for a rule may have any of the five forms for theorems plus the additional form:

$$LN_1, LN_2, \dots, LN_j \rightarrow RPL_i$$

where LN_1, LN_2, \dots, LN_j are earlier line numbers, and RPL_i is one of the five proof line forms for theorems.

The meaning of the above proof line is that if the statements in lines LN_1, LN_2, \dots, LN_j are theorems, then RPL_i corresponds to a theorem.

DRL-IFTR1)

From: $\vdash (\text{IF } P \text{ } Q)$ To: $\vdash (\text{IF IFT-MC}(P) \text{ IFT-MC}(Q))$

where IFT-MC(P) is a modal chain formed by INEV, POS, IFTRIED, and P-IFTRIED operators with P embedded on the inside

Lemma1)

From: $\vdash (\text{IF } P \text{ } Q)$ To: $\vdash (\text{IF } (\text{IFTRIED } \pi \text{ } P) (\text{IFTRIED } \pi \text{ } Q))$

Proof of Lemma1

- 1) $\vdash (\text{IF } P \text{ } Q)$
<assumption for proof of rule>
- 2) $\vdash (\text{IFTRIED } \pi (\text{IF } P \text{ } Q))$
<RL-IFTR applied to 1>
- 3) $1 \rightarrow \vdash (\text{IF } (\text{IFTRIED } \pi (\text{IF } P \text{ } Q))$
 $(\text{IF } (\text{IFTRIED } \pi \text{ } P) (\text{IFTRIED } \pi \text{ } Q)))$
<AX-IFTR?>
- 4) $1 \rightarrow \vdash (\text{IF } (\text{IFTRIED } \pi \text{ } P) (\text{IFTRIED } \pi \text{ } Q)))$
<MP applied to 2,3>

QED (Lemma1)

Lemma2)

From: $\vdash (\text{IF } P \text{ } Q)$ To: $\vdash (\text{IF } (\text{P-FTRIED } \pi \text{ } P) (\text{P-FTRIED } \pi \text{ } Q))$

Proof of Lemma2

- 1) $\vdash (\text{IF } P \text{ } Q)$
<assumption for proof of rule>
- 2) $1 \rightarrow \vdash (\text{IF } (\text{NOT } Q) (\text{NOT } P))$
<MP applied to 1, TRANSP>
- 3) $1 \rightarrow \vdash (\text{IF } (\text{IFTRIED } \pi (\text{NOT } Q)) (\text{IFTRIED } \pi (\text{NOT } P)))$
<Lemma1 with substitutions $[(\text{NOT } Q)/P, (\text{NOT } P)/Q]$ applied to 2>
- 4) $1 \rightarrow \vdash (\text{IF } (\text{NOT } (\text{IFTRIED } \pi (\text{NOT } P)))$
 $(\text{NOT } (\text{IFTRIED } \pi (\text{NOT } Q))))$
<MP applied to 3, TRANSP>
- 5) $1 \rightarrow \vdash (\text{IF } (\text{P-FTRIED } \pi \text{ } P) (\text{P-FTRIED } \pi \text{ } Q))$
<(P-IFTRIED π P) and (P-IFTRIED π Q) substituted for their definitions appearing in 4>

QED (Lemma2)

Proof of DRL-IFTR1 continued on next page

Continuation of proof of DRL-IFTR1

Proof of DRL-IFTR1 by induction on the length of the modal chain

Proof of Base case: length = 1

If IFT-MC has length 1, then IFT-MC(P) has form $\neg(\text{INEV } i P)$, $\neg(\text{POS } i P)$, $\neg(\text{IFTRIED } \text{pi } P)$, or $\neg(\text{P-IFTRIED } \text{pi } P)$

The proof that DRL-IFTR1 is true when IFT-MC(P) has form $\neg(\text{INEV } i P)$ or $\neg(\text{POS } i P)$ follows from DRL-INV1

The proof that DRL-IFTR1 is true when IFT-MC(P) has form $\neg(\text{IFTRIED } \text{pi } P)$ or $\neg(\text{P-IFTRIED } \text{pi } P)$ follows from Lemma1 and Lemma2, respectively

QED (Base case)

Proof of Inductive step

Assume that DRL-IFTR1 holds for any IFT-MC_n that has length n. We prove that DRL-IFTR1 holds for any modal chain with length n+1.

If the modal chain IFT-MC has length n+1, then IFT-MC(P) has form $\neg(\text{INEV } i \text{ IFT-MC}_n(P))$, $\neg(\text{POS } i \text{ IFT-MC}_n(P))$, $\neg(\text{IFTRIED } \text{pi } \text{ IFT-MC}_n(P))$ or $\neg(\text{P-IFTRIED } \text{pi } \text{ IFT-MC}_n(P))$ where IFT-MC_n is a modal chain with length n

Proof of DRL-IFTR1 for $\neg(\text{INEV } i \text{ IFT-MC}_n(P))$

- 1) $\vdash (\text{IF } P \ Q)$
<assumption for proof of rule>
- 2) $1 \rightarrow \vdash (\text{IF IFT-MC}_n(P) \ \text{IFT-MC}_n(Q))$
<inductive hypothesis rule applied to 1>
- 3) $1 \rightarrow \vdash (\text{IF } (\text{INEV } i \text{ IFT-MC}_n(P)) \ (\text{INEV } i \text{ IFT-MC}_n(Q)))$
<DRL-INV1 applied to 2>

QED

Proof of inductive step continued on next page

Continuation of proof of DRL-IFTR1

Continuation of proof of inductive step

Proof of DRL-IFTR1 for $\overline{[(\text{POS } i \text{ MC}_n(P))]}$

- 1) $\vdash (\text{IF } P \text{ } Q)$
<assumption for proof of rule>
- 2) $1 \rightarrow \vdash (\text{IF IFT-MC}_n(P) \text{ IFT-MC}_n(Q))$
<inductive hypothesis applied to 1>
- 3) $1 \rightarrow \vdash (\text{IF } (\text{POS } i \text{ IFT-MC}_n(P))$
 $(\text{POS } i \text{ IFT-MC}_n(Q)))$
<DRL-INV1 applied to 2>

QED

Proof of DRL-IFTR1 for $\overline{[(\text{IFTRIED } \pi \text{ IFT-MC}_n(P))]}$

- 1) $\vdash (\text{IF } P \text{ } Q)$
<assumption for proof of rule>
- 2) $1 \rightarrow \vdash (\text{IF IFT-MC}_n(P) \text{ IFT-MC}_n(Q))$
<inductive hypothesis rule applied to 1>
- 3) $1 \rightarrow \vdash (\text{IF } (\text{IFTRIED } \pi \text{ IFT-MC}_n(P))$
 $(\text{IFTRIED } \pi \text{ IFT-MC}_n(Q)))$
<Lemma1 applied to 2>

QED

Proof of DRL-IFTR1 holds for $\overline{[(\text{P-FTRIED } \pi \text{ MC}_n(P))]}$

- 1) $\vdash (\text{IF } P \text{ } Q)$
<assumption for proof of rule>
- 2) $1 \rightarrow \vdash (\text{IF IFT-MC}_n(P) \text{ IFT-MC}_n(Q))$
<inductive hypothesis applied to 1>
- 3) $1 \rightarrow \vdash (\text{IF } (\text{P-FTRIED } \pi \text{ IFT-MC}_n(P))$
 $(\text{P-FTRIED } \pi \text{ IFT-MC}_n(Q)))$
<Lemma2 applied to 2>

QED

QED (Inductive step)

QED (DRL-IFTR1)

DRL-IFTR2)

From: $\vdash (\text{IFF } P \ Q)$

To: $\vdash (\text{IFF } S1 \ S2)$

where $S1$ differs from $S2$ by replacing one or more occurrences of P with Q , and $S1$ and $S2$ are any sentences in the complete language

Lemma1)

From: $\vdash (\text{IFF } P \ Q)$

To: $\vdash (\text{IFF } (\text{IFTRIED } \pi \ P) \ (\text{IFTRIED } \pi \ Q))$

Proof of Lemma1

1) $\vdash (\text{IFF } P \ Q)$

<assumption for proof of rule>

2) $1 \rightarrow \vdash (\text{IF } (\text{IFTRIED } \pi \ P) \ (\text{IFTRIED } \pi \ Q))$

<DRL-IFTR1 applied to (IF part of 1)>

3) $1 \rightarrow \vdash (\text{IF } (\text{IFTRIED } \pi \ Q) \ (\text{IFTRIED } \pi \ P))$

<DRL-IFTR1 applied to (ONLY-IF part of 1)>

4) $1 \rightarrow \vdash (\text{AND } (\text{IF } (\text{IFTRIED } \pi \ P) \ (\text{IFTRIED } \pi \ Q)) \ (\text{IF } (\text{IFTRIED } \pi \ Q) \ (\text{IFTRIED } \pi \ P)))$

<AND-INTRO applied to 2,3>

5) $1 \rightarrow \vdash (\text{IFF } (\text{IFTRIED } \pi \ P) \ (\text{IFTRIED } \pi \ Q))$

<IFF substituted for its definition of in 4>

QED (Lemma1)

We prove DR-INV2 for the case where $S1$ and $S2$ differ by only one substitution of Q for P . Once this is established, we can apply this rule successively to provide for multiple substitutions of Q .

We prove DR-INV2 for the single substitution case by induction on the number of IFTRIED operators that the substitution of Q is under the scope of (we do not have consider P-IFTRIED because it is defined in terms of IFTRIED)

Proof of Base case

This is the case where the substitution of Q for P in $S1$ does not fall under any IFTRIED operators (although Q and P may be IFTRIED statements); this case can be proved using a slight variant of rule DRL-INV2; the proof of DRL-INV2 goes through when Q and P contain IFTRIED statements as long as the P being replaced in $R1$ is not under the scope of IFTRIED

QED (Base case)

Proof continued on next page

Continuation of proof of DRL-IFTR2

Proof of Inductive step

Assume that DRL-IFTR2 holds for the substitution of Q for P where P is nested under n IFTRIED operators. We prove DRL-IFTR2 holds for the substitution of P for Q where the P is nested under $n+1$ IFTRIED operators.

Let $MN_n(P)$ be a sentence containing P nested under n IFTRIED operators

Let $S1$ have form $SF(\ulcorner IFTRIED \text{ pi } MN_n(P) \urcorner)$, where $SF(S)$ is a sentence containing sentence S in which S does fall not under any IFTRIED operators. Thus, $S2$ has form $SF(\ulcorner IFTRIED \text{ pi } MN_n(Q) \urcorner)$.

- 1) $\vdash (IFF P Q)$
<assumption for proof of rule>
- 2) $1 \rightarrow \vdash (IFF MN_n(P) MN_n(Q))$
<inductive hypothesis rule applied to 1>
- 3) $1 \rightarrow \vdash (IFF (IFTRIED \text{ pi } MN_n(P)) (IFTRIED \text{ pi } MN_n(Q)))$
<Lemma1 applied to 2>
- 4) $1 \rightarrow \vdash (IFF SF(\ulcorner IFTRIED \text{ pi } MN_n(P) \urcorner) SF(\ulcorner IFTRIED \text{ pi } MN_n(Q) \urcorner))$
<the variant of DRL-INV2 (mentioned in the base case) with substitutions $\ulcorner (IFTRIED \text{ pi } MN_n(P)) \urcorner / S1, \ulcorner (IFTRIED \text{ pi } MN_n(Q)) \urcorner / S2$ applied to 3>

QED (Inductive step)

QED (DRL-IFTR2)

TH-IFTR-IN1)

$$\{ \ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS) \}$$

$$\vdash (\text{IF } ILS \text{ (IFTRIED } pi \text{ ILS)))}$$

where ILS is a sentence in the interval logic fragment. $CI(ILS)$ is the set of intervals associated with any HOLDS, OCCURS, or OCC predicate contained in ILS

Proof of TH-IFTR-IN1

- 1) (MEETS i (TIME-OF pi))
 $\langle \text{assumption} \rangle$
- 2) $\{ \ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS) \}$
 $\vdash (\text{IF (MEETS } i \text{ (TIME-OF } pi))$
 $(\text{AND (ENDS} \leq i1 \text{ } i) \dots (\text{ENDS} \leq in \text{ } i))$
 $\text{where } CI(ILS) = \{i1, \dots, in\}$
 $\langle TH-IR8 \rangle$
- 3) $\{ \ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS) \} \cup 1$
 $\vdash (\text{AND (ENDS} \leq i1 \text{ } i) \dots (\text{ENDS} \leq in \text{ } i))$
 $\langle MP \text{ applied to 1,2} \rangle$
- 4) $\{ \ulcorner \text{ENDS} \leq ix \text{ } i \urcorner \mid ix \in CI(ILS) \}$
 $\vdash (\text{IF } ILS \text{ (INEV } i \text{ ILS))}$
 $\langle IF \text{ part of } TH-INV-IL2 \rangle$
- 5) $\vdash (\text{IF (AND (ENDS} \leq i1 \text{ } i) \dots (\text{ENDS} \leq in \text{ } i))$
 $(\text{IF } ILS \text{ (INEV } i \text{ ILS)))}$
 $\langle \text{equivalent to 4 (using definition of } S \vdash P) \rangle$
- 6) $\{ \ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS) \} \cup 1$
 $\vdash (\text{IF } ILS \text{ (INEV } i \text{ ILS))}$
 $\langle MP \text{ applied to 3,5} \rangle$
- 7) $\vdash (\text{IF (PRIOR } i \text{ (TIME-OF } pi))$
 $(\text{IF (INEV } i \text{ ILS) (IFTRIED } pi \text{ ILS)))}$
 $\langle AX-IFTR5 \rangle$
- 8) 1 $\vdash (\text{PRIOR } i \text{ (TIME-OF } pi))$
 $\langle MP \text{ applied to 1, TH-IR1} \rangle$
- 9) 1 $\vdash (\text{IF (INEV } i \text{ ILS) (IFTRIED } pi \text{ ILS)))}$
 $\langle MP \text{ applied to 8,7} \rangle$
- 10) $\{ \ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS) \} \cup 1$
 $\vdash (\text{IF } ILS \text{ (IFTRIED } pi \text{ ILS))}$
 $\langle MP-TRNS \text{ applied to 6,9} \rangle$
- 11) $\{ \ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS) \}$
 $\vdash (\text{IF (MEETS } i \text{ (TIME-OF } pi)) (\text{IF } ILS \text{ (IFTRIED } pi \text{ ILS))}$
 $\langle \text{discharging assumption 1 in 10} \rangle$
- 12) $\{ \ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS) \}$
 $\vdash (\text{IF (MEETS } ?v \text{ (TIME-OF } pi)) (\text{IF } ILS \text{ (IFTRIED } pi \text{ ILS))}$
 $\langle \text{substituting a variable } ?v \text{ for } i \text{ in 11 where we pick a } ?v \text{ that does not appear in } ILS \text{ or } pi \rangle$

Proof continued on next page

Continuation of Proof of TH-IFTR-IN1

- 13) $\{\ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS)\}$
 $\vdash (\text{IF } (\exists ?v \text{ (MEETS ?v (TIME-OF } pi)))$
 $(\text{IF } ILS \text{ (IFTRIED } pi \text{ ILS))})$

<EXT-INTRO applied to 12>

- 14) $\vdash (\exists ?v \text{ (MEETS ?v (TIME-OF } pi)))$

<MP applied to AX-IR4.AND-ELM>

- 15) $\{\ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS)\}$
 $\vdash (\text{IF } ILS \text{ (IFTRIED } pi \text{ ILS))})$

<MP applied to 14,13>

QED (TH-IFTR-IN1)

TH-IFTR-IN2)

- $\{\ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS)\}$
 $\vdash (\text{IFF } ILS \text{ (IFTRIED } pi \text{ ILS))})$

where ILS is a sentence in the interval logic fragment and $CI(ILS)$ is the set of intervals associated with any HOLDS, OCCURS, or OCC predicate contained in ILS

Proof of TH-IFTR-IN2

- 1) $\{\ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS)\}$
 $\vdash (\text{IF } ILS \text{ (IFTRIED } pi \text{ ILS))})$

<TH-IFTR-IN1>

- 2) $\{\ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(\ulcorner \text{NOT } ILS \urcorner)\}$
 $\vdash (\text{IF } (\text{NOT } ILS) \text{ (IFTRIED } pi \text{ (NOT } ILS)))$

<TH-IFTR-IN1 with substitution $[(\text{NOT } ILS)/ILS]$ >

- 3) $\{\ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS)\}$
 $\vdash (\text{IF } (\text{NOT } ILS) \text{ (IFTRIED } pi \text{ (NOT } ILS)))$

<equivalent to 2, since $CI(\ulcorner \text{NOT } ILS \urcorner) = CI(ILS)$ by definition of CI >

- 4) $\{\ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS)\}$
 $\vdash (\text{IF } (\text{NOT } (\text{IFTRIED } pi \text{ (NOT } ILS))) \text{ (NOT } (\text{NOT } ILS)))$

<MP applied to 3, TRNSP>

- 5) $\vdash (\text{IF } (\text{IFTRIED } pi \text{ ILS}) \text{ (NOT } (\text{IFTRIED } pi \text{ (NOT } ILS))))$

<AX-IFTR1>

- 6) $\{\ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS)\}$
 $\vdash (\text{IF } (\text{IFTRIED } pi \text{ ILS}) \text{ (NOT } (\text{NOT } ILS)))$

<MP-TRNS applied to 5,4>

- 7) $\{\ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS)\}$
 $\vdash (\text{IF } (\text{IFTRIED } pi \text{ ILS}) \text{ ILS})$

<DRL-IFTR3 applied to 6 using equivalence NNP-P>

- 8) $\{\ulcorner \text{PRIOR } ix \text{ (TIME-OF } pi) \urcorner \mid ix \in CI(ILS)\}$
 $\vdash (\text{IFF } ILS \text{ (IFTRIED } pi \text{ ILS))})$

<AND-INTRO applied to 1,7 and then substituting in IFF for its definition>

QED (TH-IFTR-IN2)

TH-IFTR-IN4)

$\vdash (\text{IF } (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) \text{ i})$
 $(\text{IF } (\text{POS } \text{i} (\text{EXECUTABLE } \text{pi})) (\text{INEV } \text{i} (\text{EXECUTABLE } \text{pi}))))$

Proof of TH-IFTR-IN4

- 1) $(\text{ENDS} \leq (\text{TIME-OF } \text{pi}) \text{ i})$
 $\langle \text{assumption} \rangle$
- 2) $\vdash (\text{IF } (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) \text{ i})$
 $(\text{IF } (\text{POS } \text{i} (\text{OCC } \text{pi})) (\text{INEV } \text{i} (\text{OCC } \text{pi}))))$
 $\langle \text{AX-INV8} \rangle$
- 3) 1 $\vdash (\text{IF } (\text{POS } \text{i} (\text{OCC } \text{pi})) (\text{INEV } \text{i} (\text{OCC } \text{pi})))$
 $\langle \text{MP applied to 2,9} \rangle$
- 4) $\vdash (\text{IF } (\text{OCC } \text{pi}) (\text{POS } \text{i} (\text{OCC } \text{pi})))$
 $\langle \text{TH-INV-SF2 with substitution } [(\text{OCC } \text{pi}) / P_i] \rangle$
- 5) 1 $\vdash (\text{IF } (\text{OCC } \text{pi}) (\text{INEV } \text{i} (\text{OCC } \text{pi})))$
 $\langle \text{MP-TRNSP applied to 4,9} \rangle$
- 6) $\vdash (\text{IF } (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) \text{ i})$
 $(\text{IF } (\text{OCC } \text{pi}) (\text{INEV } \text{i} (\text{OCC } \text{pi}))))$
 $\langle \text{discharging the assumption in 5} \rangle$
- 7) $\vdash (\text{IF } (\text{IFTRIED } \text{pi} (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) \text{ i}))$
 $(\text{IFTRIED } \text{pi} (\text{IF } (\text{OCC } \text{pi}) (\text{INEV } \text{i} (\text{OCC } \text{pi}))))))$
 $\langle \text{DRL-IFTR1 applied to 6} \rangle$
- 8) $\vdash (\text{IFF } (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) \text{ i})$
 $(\text{IFTRIED } \text{pi} (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) \text{ i})))$
 $\langle \text{TH-IFTR-INS} \rangle$
- 9) $\vdash (\text{IF } (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) \text{ i})$
 $(\text{IFTRIED } \text{pi} (\text{IF } (\text{OCC } \text{pi}) (\text{INEV } \text{i} (\text{OCC } \text{pi}))))))$
 $\langle \text{DRL-IFTR8 applied to 7 using equivalence 8} \rangle$
- 10) 1 $\vdash (\text{IFTRIED } \text{pi} (\text{IF } (\text{OCC } \text{pi}) (\text{INEV } \text{i} (\text{OCC } \text{pi}))))$
 $\langle \text{MP applied to 1,9} \rangle$
- 11) 1 $\vdash (\text{IF } (\text{IFTRIED } \text{pi} (\text{OCC } \text{pi}))$
 $(\text{IFTRIED } \text{pi} (\text{INEV } \text{i} (\text{OCC } \text{pi}))))$
 $\langle \text{MP applied to 10, AX-IFTR2} \rangle$

Proof continued on next page

Continuation of proof of TH-IFTR-IN4

12) $1 \vdash (\text{IF } (\text{EXECUTABLE } \text{pi}) (\text{IFTRIED } \text{pi} (\text{INEV } i (\text{OCC } \text{pi}))))$
<substituting EXECUTABLE for its definition in 11>

13) $\vdash (\text{IF } (\text{IFTRIED } \text{pi} (\text{INEV } i (\text{OCC } \text{pi})))$
 $(\text{INEV } i (\text{IFTRIED } \text{pi} (\text{OCC } \text{pi}))))$
<axiom AX-IFTR6 with substitution [(OCC pi)/P]>

14) $1 \vdash (\text{IF } (\text{EXECUTABLE } \text{pi}) (\text{INEV } i (\text{IFTRIED } \text{pi} (\text{OCC } \text{pi}))))$
<MP-TRNS applied to 12,13>

15) $1 \vdash (\text{IF } (\text{EXECUTABLE } \text{pi}) (\text{INEV } i (\text{EXECUTABLE } \text{pi})))$
<substituting EXECUTABLE for its definition in 14>

16) $\vdash (\text{IF } (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) i)$
 $(\text{IF } (\text{EXECUTABLE } \text{pi}) (\text{INEV } i (\text{EXECUTABLE } \text{pi}))))$
<discharging assumption in 15>

17) $\vdash (\text{IF } (\text{INEV } i (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) i))$
 $(\text{INEV } i (\text{IF } (\text{EXECUTABLE } \text{pi})$
 $(\text{INEV } i (\text{EXECUTABLE } \text{pi}))))$
<DRL-IFTR1 applied to 16>

18) $\vdash (\text{IF } (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) i)$
 $(\text{INEV } i (\text{IF } (\text{EXECUTABLE } \text{pi})$
 $(\text{INEV } i (\text{EXECUTABLE } \text{pi}))))$
<DRL-IFTR3 applied to 17 using equivalence TH-INV-IL4>

19) $1 \vdash (\text{INEV } i (\text{IF } (\text{EXECUTABLE } \text{pi})$
 $(\text{INEV } i (\text{EXECUTABLE } \text{pi}))))$
<MP applied to 1,18>

20) $1 \vdash (\text{IF } (\text{POS } i (\text{EXECUTABLE } \text{pi}))$
 $(\text{POS } i (\text{INEV } i (\text{EXECUTABLE } \text{pi}))))$
<MP applied to 19, TH-INV-SF4>

21) $1 \vdash (\text{IF } (\text{POS } i (\text{EXECUTABLE } \text{pi}))$
 $(\text{INEV } i (\text{EXECUTABLE } \text{pi})))$
<DRL-IFTR3 applied to 20 using equivalence TH-INV-SF7>

22) $\vdash (\text{IF } (\text{ENDS} \leq (\text{TIME-OF } \text{pi}) i)$
 $(\text{IF } (\text{POS } i (\text{EXECUTABLE } \text{pi}))$
 $(\text{INEV } i (\text{EXECUTABLE } \text{pi}))))$
<discharging assumption in 21>

QED (TH-IFTR-IN4)

Appendix G

Proof of Theorems in Chapter 5

See introduction to Appendix F for conventions used in presenting proofs.

TH-APG-1)

⊢ (IFF (AND (PRIOR i (TIME-OF pi1)) (PRIOR i (TIME-OF pi2)))
(PRIOR i (TIME-OF (COMP pi1 pi2))))

Proof of TH-APG-1

- 1) ⊢ (IF (AND (IN (TIME-OF pi1) ?i2)
 (IN (TIME-OF pi2) ?i2))
 (IN (TIME-OF (COMP pi1 pi2)) ?i2))
 <IF part of AX-IL6>
- 2) (AND (IN (TIME-OF pi1) ?i2) (IN (TIME-OF pi2) ?i2))
 <assumption>
- 3) 2 ⊢ (IN (TIME-OF (COMP pi1 pi2)) ?i2)
 <MP applied to 1,2>
- 4) (MEETS i ?i2)
 <assumption>
- 5) 2.4 ⊢ (PRIOR i (TIME-OF (COMP pi1 pi2)))
 <MP applied to (AND-INTRO 3,4), TH-IR4>
- 6) ⊢ (IF (AND (MEETS i ?i2)
 (IN (TIME-OF pi1) ?i2)
 (IN (TIME-OF pi2) ?i2))
 (PRIOR i (TIME-OF (COMP pi1 pi2))))
 <discharging the assumptions in 6>
- 7) ⊢ (IF (∃ ?i2 (AND (MEETS i ?i2)
 (IN (TIME-OF pi1) ?i2)
 (IN (TIME-OF pi2) ?i2)))
 (PRIOR i (TIME-OF (COMP pi1 pi2))))
 <EXT-INTRO applied to 6>
- 8) ⊢ (IF (AND (PRIOR i (TIME-OF pi1))
 (PRIOR i (TIME-OF pi2)))
 (∃ ?i2 (AND (MEETS i ?i2)
 (IN (TIME-OF pi1) ?i2)
 (IN (TIME-OF pi2) ?i2))))
 <TH-IR5>

Proof continued on next page

Continuation of TH-APG-1

- 9) \vdash (IF (AND (PRIOR i (TIME-OF pi1))
(PRIOR i (TIME-OF pi2)))
(PRIOR i (TIME-OF (COMP pi1 pi2))))
<MP-TRANS applied to 8,7>
 - 10) (PRIOR i (TIME-OF (COMP pi1 pi2)))
<assumption>
 - 11) \vdash (IN (TIME-OF (COMP pi1 pi2))
(TIME-OF (COMP pi1 pi2)))
<TH-IR7>
 - 12) \vdash (AND (IN (TIME-OF pi1) (TIME-OF (COMP pi1 pi2)))
(IN (TIME-OF pi2) (TIME-OF (COMP pi1 pi2))))
<MP applied to 10. (ONLY-IF part of AX-IL6 with substitutions [(TIME-OF (COMP pi1 pi2))/i]>
 - 13) \vdash (IN (TIME-OF pi1) (TIME-OF (COMP pi1 pi2)))
<MP applied to 12,AND-ELIM>
 - 14) $10 \vdash$ (PRIOR i (TIME-OF pi1))
<MP applied to (AND-INTRO applied to 10,13),(TH-IR8 with substitutions [i/i1. (TIME-OF (COMP pi1 pi2))/i2, (TIME-OF pi1)/i3]>
 - 15) \vdash (IN (TIME-OF pi2) (TIME-OF (COMP pi1 pi2)))
<MP applied to 12,AND-ELIM>
 - 16) $10 \vdash$ (PRIOR i (TIME-OF pi2))
<MP applied to (AND-INTRO applied to 10,15),(TH-IR8 with substitutions [i/i1. (TIME-OF (COMP pi1 pi2))/i2, (TIME-OF pi2)/i3]>
 - 17) \vdash (IF (PRIOR i (TIME-OF (COMP pi1 pi2)))
(AND (PRIOR i (TIME-OF pi1))
(PRIOR i (TIME-OF pi2))))
<AND-INTRO applied to 14,16 then assumption discharged>
 - 18) \vdash (IFF (PRIOR i (TIME-OF (COMP pi1 pi2)))
(AND(PRIOR i (TIME-OF pi1))
(PRIOR i (TIME-OF pi2))))
<IFF substituted for its definition after AND-INTRO applied to 17,9>
- QED (TH-APG-1)

DRL-APG-1)

From: $IRS \vdash (IF\ P\ Q)$

To: $IRS \vdash (IF\ NP-MC(P)\ NP-MC(Q))$

where IRS is an interval relation statement and NP-MC is a modal chain consisting only of IFTRIED and INEV operators

Lemma1)

From: $IRS \vdash (IF\ P\ Q)$

To: $IRS \vdash (IF\ (INEV\ i\ P)\ (INEV\ i\ Q))$

Proof of Lemma1

1) $IRS \vdash (IF\ P\ Q)$

<assumption for proof of rule>

2) $1 \rightarrow \vdash (IF\ IRS\ (IF\ P\ Q))$

<equivalent to 1 by definition>

3) $1 \rightarrow \vdash (IF\ (INEV\ i\ IRS)\ (INEV\ i\ (IF\ P\ Q)))$

<DRL-IFTR1 applied to 2>

4) $1 \rightarrow \vdash (IF\ IRS\ (INEV\ i\ (IF\ P\ Q)))$

<DRL-IFTR3 applied to 3 using equivalence TH-INV-IL4>

5) $1 \vdash IRS \vdash (IF\ (INEV\ i\ P)\ (INEV\ i\ Q))$

<assumption IRS discharged after MP-TRNS applied to 4, AX-INV2>

QED (Lemma1)

Lemma2)

From: $IRS \vdash (IF\ P\ Q)$

To: $IRS \vdash (IF\ (IFTRIED\ pi\ P)\ (IFTRIED\ pi\ Q))$

Proof of Lemma2

1) $IRS \vdash (IF\ P\ Q)$

<assumption for proof of rule>

2) $1 \vdash (IF\ IRS\ (IF\ P\ Q))$

<equivalent to 1 by definition>

3) $1 \rightarrow \vdash (IF\ (IFTRIED\ pi\ IRS)\ (IFTRIED\ pi\ (IF\ P\ Q)))$

<DRL-IFTR1 applied to 2>

4) $1 \rightarrow \vdash (IF\ IRS\ (IFTRIED\ pi\ (IF\ P\ Q)))$

<DRL-IFTR3 applied to 3 using equivalence TH-INV-IL4>

5) $1 \rightarrow IRS \vdash (IF\ (IFTRIED\ pi\ P)\ (IFTRIED\ pi\ Q))$

<discharging assumption IRS after MP-TRNS applied to 4, AX-IFTR2>

QED (Lemma2)

Proof continued on next page

Continuation of proof of DRL-APG-1

Proof of DRL-APG-1 by induction on the length of the modal chain

Proof of Base case: length = 1

If NP-MC has length 1, then NP-MC(P) has form $\neg(\text{INEV } i \text{ } P)$ or $\neg(\text{IFTRIED } \pi i \text{ } P)$

The proof that DRL-APG-1 is true when NP-MC(P) has form $\neg(\text{INEV } i \text{ } P)$ follows from Lemma1

Similarly, the proof that DRL-APG-1 is true when NP-MC(P) has form $\neg(\text{IFTRIED } \pi i \text{ } P)$ follows from Lemma2

QED (Base case)

Proof of Inductive step

Assume that DRL-APG-1 holds for any NP-MC_n that has length n. We prove that DRL-APG-1 holds for any modal chain with length n+1. If NP-MC has length n+1, then NP-MC(P) has form $\neg(\text{INEV } i \text{ } \text{NP-MC}_n(P))$ or $\neg(\text{IFTRIED } \pi i \text{ } \text{NP-MC}_n(P))$ where NP-MC_n is a modal chain with length n.

Proof of DRL-APG-1 for $\neg(\text{INEV } i \text{ } \text{NP-MC}_n(P))$

- 1) $\text{IRS} \vdash (\text{IF } P \text{ } Q)$
<assumption for proof of rule>
- 2) $1 \rightarrow \text{IRS} \vdash (\text{IF } \text{NP-MC}_n(P) \text{ } \text{NP-MC}_n(Q))$
<inductive hypothesis rule applied to 1>
- 3) $1 \rightarrow \vdash (\text{IF } \text{IRS} (\text{IF } \text{NP-MC}_n(P) \text{ } \text{NP-MC}_n(Q)))$
<equivalent to 2 by definition>
- 4) $1 \rightarrow \vdash (\text{IF } (\text{INEV } i \text{ } \text{IRS})$
 $(\text{INEV } i (\text{IF } \text{NP-MC}_n(P) \text{ } \text{NP-MC}_n(Q))))$
<DRL-IFTR1 applied to 3>
- 5) $1 \rightarrow \vdash (\text{IF } \text{IRS}$
 $(\text{INEV } i (\text{IF } \text{NP-MC}_n(P) \text{ } \text{NP-MC}_n(Q))))$
<DRL-IFTR3 applied to 4 using equivalence TH-INV-IL4>
- 6) $1 \rightarrow \vdash (\text{IF } \text{IRS} (\text{IF } (\text{INEV } i \text{ } \text{NP-MC}_n(P))$
 $(\text{INEV } i \text{ } \text{NP-MC}_n(Q))))$
<MP-TRNS applied to 5, AX-INV2>
- 7) $1 \rightarrow \text{IRS} \vdash (\text{IF } (\text{INEV } i \text{ } \text{NP-MC}_n(P))$
 $(\text{INEV } i \text{ } \text{NP-MC}_n(Q)))$
<equivalent to 6 by definition>

QED

Proof continued on next page

Continuation of proof of DRL-APG-1 (inductive step)

Proof of DRL-APG-1 for $\overline{[(\text{IFTRIED } \pi \text{ NP-MC}_n(P))]}$

- 1) $\text{IRS} \vdash (\text{IF } P \text{ } Q)$
<assumption for proof of rule>
- 2) $1 \rightarrow \text{IRS} \vdash (\text{IF NP-MC}_n(P) \text{ NP-MC}_n(Q))$
<inductive hypothesis rule applied to 1>
- 3) $1 \rightarrow \vdash (\text{IF IRS } (\text{IF NP-MC}_n(P) \text{ NP-MC}_n(Q)))$
<equivalent to 2 by definition>
- 4) $1 \rightarrow \vdash (\text{IF } (\text{IFTRIED } \pi \text{ IRS}) \text{ } (\text{IFTRIED } \pi \text{ } (\text{IF NP-MC}_n(P) \text{ NP-MC}_n(Q))))$
<DRL-IFTR1 applied to 3>
- 5) $1 \rightarrow \vdash (\text{IF IRS } (\text{IFTRIED } \pi \text{ } (\text{IF NP-MC}_n(P) \text{ NP-MC}_n(Q))))$
<DRL-IFTR3 applied to 4 using equivalence TH-INV-IL4>
- 6) $1 \rightarrow \vdash (\text{IF IRS } (\text{IF } (\text{IFTRIED } \pi \text{ NP-MC}_n(P)) \text{ } (\text{IFTRIED } \pi \text{ NP-MC}_n(Q))))$
<MP-TRNS applied to 5, AX-IFTR2>
- 7) $1 \rightarrow \text{IRS} \vdash (\text{IF } (\text{IFTRIED } \pi \text{ NP-MC}_n(P)) \text{ } (\text{IFTRIED } \pi \text{ NP-MC}_n(Q)))$
<equivalent to 6 by definition>

QED

QED (Inductive step)

QED (DRL-APG-1)

TH-APG-2)

$\vdash (\text{IF } (\text{AND } (\text{PRIOR } i \text{ } (\text{TIME-OF } \pi)) \text{ } (\text{INEV } i \text{ } P)) \text{ } (\text{INEV } i \text{ } (\text{IFTRIED } \pi \text{ } P)))$

Proof of TH-APG-2

- 1) $(\text{PRIOR } i \text{ } (\text{TIME-OF } \pi))$
<assumption>
- 2) $(\text{INEV } i \text{ } P)$
<assumption>
- 3) $\vdash (\text{IF } (\text{INEV } i \text{ } (\text{PRIOR } i \text{ } (\text{TIME-OF } \pi))) \text{ } (\text{INEV } i \text{ } (\text{IF } (\text{INEV } i \text{ } P) \text{ } (\text{IFTRIED } \pi \text{ } P))))$
<DRL-IFTR1 applied to AX-IFTR5>
- 4) $\vdash (\text{IF } (\text{PRIOR } i \text{ } (\text{TIME-OF } \pi)) \text{ } (\text{INEV } i \text{ } (\text{IF } (\text{INEV } i \text{ } P) \text{ } (\text{IFTRIED } \pi \text{ } P))))$
<DRL-IFTR3 applied to 3 using equivalence TH-INV-IL4>
- 5) $1 \vdash (\text{INEV } i \text{ } (\text{IF } (\text{INEV } i \text{ } P) \text{ } (\text{IFTRIED } \pi \text{ } P)))$
<MP applied to 1, 4>
- 6) $1 \vdash (\text{IF } (\text{INEV } i \text{ } (\text{INEV } i \text{ } P)) \text{ } (\text{INEV } i \text{ } (\text{IFTRIED } \pi \text{ } P)))$
<MP applied to 5, (AX-INV2 with substitutions $[(\text{INEV } i \text{ } P)/P, (\text{IFTRIED } \pi \text{ } P)/Q]$ >
- 7) $2 \vdash (\text{INEV } i \text{ } (\text{INEV } i \text{ } P))$
<MP applied to 2, AX-IFTR5>
- 8) $1, 2 \vdash (\text{INEV } i \text{ } (\text{IFTRIED } \pi \text{ } P))$
<MP applied to 7, 6>
- 9) $\vdash (\text{IF } (\text{AND } (\text{PRIOR } i \text{ } (\text{TIME-OF } \pi)) \text{ } (\text{INEV } i \text{ } P)) \text{ } (\text{INEV } i \text{ } (\text{IFTRIED } \pi \text{ } P)))$
<discharging assumptions in 8>

QED (TH-APG-2)

TH-APG-3)

\vdash (IF (IFTRIED pi1 (IFTRIED pi2 (AND (OCC pi1) (OCC pi2))))
(EXECUTABLE (COMP pi1 pi2)))

Proof of TH-APG-3

- 1) (IFTRIED pi1 (IFTRIED pi2 (AND (OCC pi1) (OCC pi2))))
 <assumption>
- 2) \vdash (IF (IFTRIED pi1
 (IFTRIED pi2 (AND (OCC pi1) (OCC pi2))))
 (IFF (IFTRIED (COMP pi1 pi2)
 (OCC (COMP pi1 pi2))
 (IFTRIED pi1
 (IFTRIED pi2 (OCC (COMP pi1 pi2)))))))
 <AX-IFTR7 with substitution [(OCC (COMP p1 pi2))/P]>
- 3) 1 \vdash (IFF (IFTRIED (COMP pi1 pi2) (OCC (COMP pi1 pi2)))
 (IFTRIED pi1
 (IFTRIED pi2 (OCC (COMP pi1 pi2)))))
 <MP applied to 1.2>
- 4) 1 \vdash (IFF (EXECUTABLE (COMP pi1 pi2))
 (IFTRIED pi1
 (IFTRIED pi2 (OCC (COMP pi1 pi2)))))
 <substituting EXECUTABLE for its definition in 3>
- 5) 1 \vdash (IF (IFTRIED pi1 (IFTRIED pi2 (OCC (COMP pi1 pi2))))
 (EXECUTABLE (COMP pi1 pi2)))
 <ONLY-IF part of 4>
- 6) 1 \vdash (IF (IFTRIED pi1
 (IFTRIED pi2 (AND (OCC pi1) (OCC pi2))))
 (EXECUTABLE (COMP pi1 pi2)))
 <DRL-IFTR8 applied to 5 using equivalence AX-IL5>
- 7) 1 \vdash (EXECUTABLE (COMP pi1 pi2))
 <MP applied to 1.6>
- 8) \vdash (IF (IFTRIED pi1
 (IFTRIED pi2 (AND (OCC pi1) (OCC pi2))))
 (EXECUTABLE (COMP pi1 pi2)))
 <discharging the assumption in 7>

QED (TH-APG-3)

TH-CH5-1)

If $S \models (\text{PRIOR } i (\text{TIME-OF } pi))$ and
 $S \models (\text{INEV } i (\text{IFTRIED } pi (\text{OCC } pi)))$
 then $S \models (\text{POS } i (\text{OCC } pi))$

Lemma1)

$\vdash (\text{IF } (\text{AND } (\text{PRIOR } i (\text{TIME-OF } pi))$
 $(\text{INEV } i (\text{IFTRIED } pi (\text{OCC } pi))))$
 $(\text{POS } i (\text{OCC } pi)))$

Proof of Lemma1

- 1) $(\text{PRIOR } i (\text{TIME-OF } pi))$
<assumption>
- 2) $(\text{INEV } i (\text{IFTRIED } pi (\text{OCC } pi)))$
<assumption>
- 3) $1 \vdash (\text{IF } (\text{INEV } i (\text{NOT } (\text{OCC } pi)))$
 $(\text{IFTRIED } pi (\text{NOT } (\text{OCC } pi))))$
<MP applied to 1, (AX-IFTR5 with substitution [(NOT (OCC pi))/P]>
- 4) $1 \vdash (\text{IF } (\text{NOT } (\text{IFTRIED } pi (\text{NOT } (\text{OCC } pi))))$
 $(\text{NOT } (\text{INEV } i (\text{NOT } (\text{OCC } pi))))$
<MP applied to 3, TRANSP>
- 5) $1 \vdash (\text{IF } (\text{IFTRIED } pi (\text{OCC } pi))$
 $(\text{NOT } (\text{INEV } i (\text{NOT } (\text{OCC } pi))))$
<MP-TRANS applied to AX-IFTR1,4>
- 6) $1 \vdash (\text{IF } (\text{IFTRIED } pi (\text{OCC } pi)) (\text{POS } i (\text{OCC } pi)))$
<substituting POS for its definition in 5>
- 7) $1 \vdash (\text{IF } (\text{INEV } i (\text{IFTRIED } pi (\text{OCC } pi)))$
 $(\text{INEV } i (\text{POS } i (\text{OCC } pi))))$
<DRL-APG-1 applied to 6>
- 8) $1,2 \vdash (\text{INEV } i (\text{POS } i (\text{OCC } pi)))$
<MP applied to 2,7>
- 9) $1,2 \vdash (\text{POS } i (\text{OCC } pi))$
<MP applied to 8, AX-INV1>
- 10) $\vdash (\text{IF } (\text{AND } (\text{PRIOR } i (\text{TIME-OF } pi))$
 $(\text{INEV } i (\text{IFTRIED } pi (\text{OCC } pi))))$
 $(\text{POS } i (\text{OCC } pi)))$
<discharging assumptions in 9>

QED (Lemma1)

Proof continued on next page

Lemma2)

If $\vdash (\text{IF } (\text{AND } P \ Q) \ R)$ and $S \models P$ and $S \models Q$
then $S \models R$

Proof of Lemma2

- 1) $\vdash (\text{IF } (\text{AND } P \ Q) \ R)$
<assumption for proof of meta-theorem>
- 2) $S \models P$
<assumption>
- 3) $S \models Q$
<assumption>
- 4) $2 \rightarrow$ For every model (m), if for every sentence (A)
if $A \in S$ then A is true in m, then P is true in m
<using definition of \models in 2>
- 5) $3 \rightarrow$ For every model (m), if for every sentence (A)
if $A \in S$ then A is true in m, then Q is true in m
<using definition of \models in 3>
- 6) $2,3 \rightarrow$ For every model (m), if for every sentence (A)
if $A \in S$ then A is true in m, then P is true in m
and Q is true in m
<taking 4 and 5 together>
- 7) $2,3 \rightarrow$ For every model (m), if for every sentence (A)
if $A \in S$ then A is true in m, then $\lceil (\text{AND } P \ Q) \rceil$
is true in m
<using the interpretation of AND in 6>
- 8) $1 \rightarrow S \models (\text{IF } (\text{AND } P \ Q) \ R)$
<using 1 and the fact ' $S \models X$ ' is true if ' $S \vdash X$ ' is true (our proof theory is sound)>
- 9) $1 \rightarrow$ For every model (m), if for every sentence (A)
if $A \in S$ then A is true in m, then $\lceil (\text{IF } (\text{AND } P \ Q) \ R) \rceil$
is true in m
<using definition of \models in 8>
- 10) $1 \rightarrow$ For every model (m), if for every sentence (A)
if $A \in S$ then A is true in m, then if $\lceil (\text{AND } P \ Q) \rceil$
is true in m then R is true in m
<using the interpretation of IF in 9>
- 11) $1-3 \rightarrow$ For every model (m), if for every sentence (A)
if $A \in S$ then A is true in m, then R is true in m
<taking 7 and 10 together>
- 12) $1-3 \rightarrow S \models R$
<substituting \models for its definition in 11>

QED (Lemma2)

Proof continued on next page

Continuation of proof of TH-CH5-1

Proof of TH-CH5-1

- 1) $S \models (\text{PRIOR } i (\text{TIME-OF } pi))$
 $\langle \text{assumption} \rangle$
 - 2) $S \models (\text{INEV } i (\text{IFTRIED } pi (\text{OCC } pi)))$
 $\langle \text{assumption} \rangle$
 - 3) $\vdash (\text{IF } (\text{AND } (\text{PRIOR } i (\text{TIME-OF } pi))$
 $(\text{INEV } i (\text{IFTRIED } pi (\text{OCC } pi))))$
 $(\text{POS } i (\text{OCC } pi)))$
 $\langle \text{Lemma1} \rangle$
 - 4) $1,2 \rightarrow S \models (\text{POS } i (\text{OCC } pi))$
 $\langle \text{Lemma2 applied to 3,1,2 using substitutions } [(\text{PRIOR } i (\text{TIME-OF } pi)) / P, (\text{INEV } i$
 $(\text{IFTRIED } pi (\text{OCC } pi))) / Q, (\text{POS } i (\text{OCC } pi)) / R_i \rangle$
- QED (TH-CH5-1)

SEQ-TH1)

- $$\vdash (\text{IF } (\text{PRIOR } (\text{TIME-OF } pi1)) (\text{TIME-OF } pi2)$$
- $$(\text{IFF } (\text{EXECUTABLE } (\text{COMP } pi1 pi2))$$
- $$(\text{AND } (\text{EXECUTABLE } pi1)$$
- $$(\text{IFTRIED } pi1 (\text{EXECUTABLE } pi2))))))$$

Proof of SEQ-TH1

- 1) $(\text{PRIOR } (\text{TIME-OF } pi1) (\text{TIME-OF } pi2))$
 $\langle \text{assumption} \rangle$
- 2) $(\text{EXECUTABLE } pi1)$
 $\langle \text{assumption} \rangle$
- 3) $(\text{IFTRIED } pi1 (\text{EXECUTABLE } pi2))$
 $\langle \text{assumption} \rangle$
- 4) $3 \vdash (\text{IFTRIED } pi1 (\text{IFTRIED } pi2 (\text{OCC } pi2)))$
 $\langle \text{replacing EXECUTABLE by its definition in 3} \rangle$
- 5) $1 \vdash (\text{IF } (\text{OCC } pi1) (\text{IFTRIED } pi2 (\text{OCC } pi1)))$
 $\langle \text{MP applied to 1, (TH-IFTR-IN1 with substitutions } [pi2/pi, (\text{OCC } pi1)/ILS] \rangle$
- 6) $1 \vdash (\text{IF } (\text{IFTRIED } pi1 (\text{OCC } pi1))$
 $(\text{IFTRIED } pi1 (\text{IFTRIED } pi2 (\text{OCC } pi1))))$
 $\langle \text{DRL-APG-1 applied to 5} \rangle$
- 7) $1,2 \vdash (\text{IFTRIED } pi1 (\text{IFTRIED } pi2 (\text{OCC } pi1)))$
 $\langle \text{MP applied to 2,6 after EXECUTABLE is replaced by its definition in 2} \rangle$
- 8) $1-3 \vdash (\text{IFTRIED } pi1 (\text{IFTRIED } pi2 (\text{AND } (\text{OCC } pi1) (\text{OCC } pi2))))$
 $\langle \text{DRL-IFTR5 applied to 7,4} \rangle$

Proof continued on next page

Continuation of proof of SEQ-TH1

- 9) 1-3 \vdash (IF (IFTRIED pi1 (IFTRIED pi2 (AND (OCC pi1) (OCC pi2))))
 (IFTRIED (COMP pi1 pi2) (AND (OCC pi1) (OCC pi2))))
<ONLY-IF part of (MP applied to 8, (AX-IFTR7 with substitution [(AND (OCC pi1) (OCC pi2))/P_i])>
- 10) 1-3 \vdash (IFTRIED (COMP pi1 pi2) (AND (OCC pi1) (OCC pi2)))
<MP APPLIED to 8,9>
- 11) 1-3 \vdash (IFTRIED (COMP pi1 pi2) (OCC (COMP pi1 pi2)))
<DRL-IFTR9 applied TO 10 using equivalence AX-IL5>
- 12) 1 \vdash (IF (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2)))
 (EXECUTABLE (COMP pi1 pi2)))
<discharging assumption 2 and 3 in 11 after substituting EXECUTABLE for its definition>
- 13) 1 \vdash (IF (EXECUTABLE (COMP pi1 pi2))
 (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2))))
<MP applied to 1, AX-IFTR8>
- 14) 1 \vdash (IFF (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2)))
 (EXECUTABLE (COMP pi1 pi2)))
<AND-INTRO applied to 12,13 and then IFF substituted for its definition>
- 15) \vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 (IFF (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2))
 (EXECUTABLE (COMP pi1 pi2)))))
<discharging assumption in 14>

QED (SEQ-TH1)

SEQ-TH2)

⊢ (IF (PRIOR (TIME-OF pi1)) (TIME-OF pi2))
 (IFF (INEV i (EXECUTABLE (COMP pi1 pi2)))
 (INEV i (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2))))))

Proof of SEQ-TH2

- 1) (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 <assumption>
- 2) 1 ⊢ (IFF (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2))))
 (EXECUTABLE (COMP pi1 pi2)))
 <MP applied to 1, SEQ-TH1>
- 3) 1 ⊢ (IF (INEV i (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2))))
 (INEV i (EXECUTABLE (COMP pi1 pi2))))
 <DRL-APG-1 applied to (IF part of 2)>
- 4) 1 ⊢ (IF (INEV i (EXECUTABLE (COMP pi1 pi2)))
 (INEV i (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2)))))
 <DRL-APG-1 applied to (ONLY-IF part of 2)>
- 5) 1 ⊢ (IFF (INEV i (EXECUTABLE (COMP pi1 pi2)))
 (INEV i (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2)))))
 <AND-INTRO applied to 3,4 and then IFF substituted for its definition>
- 6) ⊢ (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 (IFF (INEV i (EXECUTABLE (COMP pi1 pi2)))
 (INEV i (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2))))))
 <discharging the assumption in 5>

QED (SEQ-TH2)

TH-HT2)

$$\vdash (\text{IF } (\text{AND } (\text{PRIOR } I_p (\text{TIME-OF } p_{i1})) (\text{PRIOR } I_p (\text{TIME-OF } p_{i2}))) \\ (\text{NOT } (\text{DISJOINT } I_1 I_2)) \\ (\text{NOT } (= p_{n1} p_{n2}))) \\ (\forall ?p_{nx} ?p_{ny} ?brnr_x ?brnr_y ?ix ?iy (\text{INEV } I_p \text{ HT})) \\ (\text{INEV } I_p (\text{NOT } (\text{EXECUTABLE} \\ (\text{COMP } (\text{htg } p_{n1} \text{ brnr})@I_1 (\text{htg } p_{n2} \text{ brnr})@I_2))))))$$

where $\text{HT} =_{\text{def}}$

$$(\text{IF } (\text{AND } (\text{OCC } (\text{htg } ?p_{nx} ?brnr_x)@?ix) \\ (\text{OCC } (\text{htg } ?p_{ny} ?brnr_y)@?iy)) \\ (\text{OR } (\text{DISJOINT } ?ix ?iy) \\ (\text{AND } (\text{NOT } (= ?p_{nx} ?p_{ny})) (\text{NOT } (= ?brnr_x ?brnr_y))) \\ (\text{AND } (= ?p_{nx} ?p_{ny}) (= ?brnr_x ?brnr_y) (= ?ix ?iy))))$$

Lemma1)

$$\vdash (\text{IFF } (\text{NOT } (= t_1 t_2)) (\text{INEV } i (\text{NOT } (= t_1 t_2))))$$

Proof of Lemma1

- 1) $\vdash (\text{IF } (= t_1 t_2) (\text{INEV } i (= t_1 t_2)))$
<IF part of TH-INV-SF20>
- 2) $\vdash (\text{IF } (\text{NOT } (\text{INEV } i (= t_1 t_2))) (\text{NOT } (= t_1 t_2)))$
<MP applied to 1, TRANSP>
- 3) $\vdash (\text{IF } (\text{NOT } (\text{NOT } (\text{POS } i (\text{NOT } (= t_1 t_2)))))$
 $(\text{NOT } (= t_1 t_2)))$
<DRL-IFTR8 applied to 2 using equivalence TH-INV-SF1>
- 4) $\vdash (\text{IF } (\text{POS } i (\text{NOT } (= t_1 t_2))) (\text{NOT } (= t_1 t_2)))$
<DRL-IFTR8 applied to 3 using equivalence NNP-P>
- 5) $\vdash (\text{IF } (\text{INEV } i (\text{POS } i (\text{NOT } (= t_1 t_2))))$
 $(\text{INEV } i (\text{NOT } (= t_1 t_2))))$
<DRL-IFTR1 applied to 4>
- 6) $\vdash (\text{IF } (\text{POS } i (\text{NOT } (= t_1 t_2))) (\text{INEV } i (\text{NOT } (= t_1 t_2))))$
<DRL-IFTR8 applied to 5 using equivalence TH-INV-SF6>
- 7) $\vdash (\text{IF } (\text{NOT } (= t_1 t_2)) (\text{INEV } i (\text{NOT } (= t_1 t_2))))$
<MP-TRANS applied to (TH-INV-SF2 with substitution $[(\text{NOT } (= t_1 t_2))/P]$, 6)>
- 8) $\vdash (\text{IF } (\text{INEV } i (\text{NOT } (= t_1 t_2))) (\text{NOT } (= t_1 t_2)))$
<AX-INV1 with substitution $[(\text{NOT } (= t_1 t_2))/P]$ >
- 9) $\vdash (\text{IFF } (\text{NOT } (= t_1 t_2)) (\text{INEV } i (\text{NOT } (= t_1 t_2))))$
<AND-INTRO applied to 7,8 and IFF substituted for its definition>

QED (Lemma1)

Proof continued on next page

Continuation of proof of TH-HT2

Proof of TH-HT2

- 1) $(\forall ?pnx ?pny ?brnrx ?brnry ?ix ?iy \text{ HT})$
 $\langle \text{assumption} \rangle$
- 2) $(\text{NOT} (\text{DISJOINT } I1 \ I2))$
 $\langle \text{assumption} \rangle$
- 3) $(\text{NOT} (= pn1 \ pn2)))$
 $\langle \text{assumption} \rangle$
- 4) $\vdash (= brnr \ brnr)$
 $\langle \text{AX-FO6} \rangle$
- 5) $\vdash (\text{NOT} (\text{NOT} (= brnr \ brnr)))$
 $\langle \text{MP applied to 4, (ONLY-IF part of NNP-P)} \rangle$
- 6) $\vdash (\text{OR} (\text{NOT} (\text{NOT} (= pn1 \ pn2)))$
 $(\text{NOT} (\text{NOT} (= brnr \ brnr))))$
 $\langle \text{MP applied to 5, AX-FO2} \rangle$
- 7) $\vdash (\text{NOT} (\text{AND} (\text{NOT} (= pn1 \ pn2)) (\text{NOT} (= brnr \ brnr))))$
 $\langle \text{SUBST applied to 6 using equivalence DEM1} \rangle$
- 8) $3 \vdash (\text{OR} (\text{NOT} (= pn1 \ pn2)) (\text{NOT} (= brnr \ brnr)) (\text{NOT} (= I1 \ I2)))$
 $\langle \text{MP applied to (MP applied to 3, AX-FO2), AX-FO2} \rangle$
- 9) $3 \vdash (\text{NOT} (\text{AND} (= pn1 \ pn2) (= brnr \ brnr) (= I1 \ I2)))$
 $\langle \text{SUBST applied twice to 8 using equivalence DEM1} \rangle$
- 10) $2,3 \vdash (\text{AND} (\text{NOT} (\text{DISJOINT } I1 \ I2))$
 $(\text{NOT} (\text{AND} (\text{NOT} (= pn1 \ pn2)) (\text{NOT} (= brnr \ brnr))))$
 $(\text{NOT} (\text{AND} (= pn1 \ pn2) (= brnr \ brnr) (= I1 \ I2))))$
 $\langle \text{AND-INTRO applied to (AND-INTRO applied to 2, 7), 9} \rangle$
- 11) $2,3 \vdash (\text{NOT} (\text{OR} (\text{DISJOINT } I1 \ I2)$
 $(\text{AND} (\text{NOT} (= pn1 \ pn2)) (\text{NOT} (= brnr \ brnr)))$
 $(\text{AND} (= pn1 \ pn2) (= brnr \ brnr) (= I1 \ I2))))$
 $\langle \text{SUBST applied twice to 10 using equivalence DEM2} \rangle$
- 12) $1-3 \vdash (\text{NOT} (\text{AND} (\text{OCC} (\text{htg } pn1 \ brnr) @I1) (\text{OCC} (\text{htg } pn2 \ brnr) @I2)))$
 $\langle \text{MP applied to 11, (MP applied to (MP applied to 1, AX-FO5 for the variable substitutions$
 $\{pn1/?pnx, pn2/?pny, brnr/?brnrx, brnr/?brnry, I1/?ix, I2/?iy\}, \text{TRNSP})} \rangle$
- 13) $\vdash (\text{IF} (\text{AND} (\text{INEV } Ip \ (\forall ?pnx ?pny ?brnrx ?brnry ?ix ?iy \text{ HT}))$
 $(\text{INEV } Ip \ (\text{NOT} (\text{DISJOINT } I1 \ I2)))$
 $(\text{INEV } Ip \ (\text{NOT} (= pn1 \ pn2))))$
 $(\text{INEV } Ip \ (\text{NOT} (\text{AND} (\text{OCC} (\text{htg } pn1 \ brnr) @I1)$
 $(\text{OCC} (\text{htg } pn2 \ brnr) @I2))))$
 $\langle \text{DRL-IFTR3 applied two times to (DRL-IFTR1 applied to 12 after discharging assumptions)$
 $\text{using equivalence TH-INV-SF9} \rangle$

Proof continued on next page

TH-CR2)

$$\vdash (\text{IF } (\text{PRIOR } I_p I) \\ (\text{IF } (\text{AND } (\text{INEV } I_p \text{ CR-S1}) \\ (\forall ?i ?pi (\text{INEV } I_p (\text{IF } (\text{NOT } (\text{HOLDS } \text{pnf } ?i)) \\ (\text{IFTRIED } ?pi (\text{NOT } (\text{HOLDS } \text{pnf } ?i)))))) \\ (\text{INEV } I_p (\text{IF } (\text{EXECUTABLE } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I)) \\ (\text{HOLDS } \text{pnf } I))))$$

where $\text{CR-S1} =_{\text{def}}$
 $(\text{IF } (\text{AND } (\text{OCC } (\text{carry } sc1)@I) (\text{OCC } (\text{carry } sc2)@I)) (\text{HOLDS } \text{pnf } I))$

Proof of TH-CR2

- 1) (PRIOR $I_p I$)
<assumption>
- 2) (INEV $I_p \text{ CR-S1}$)
<assumption>
- 3) $(\forall ?i ?pi$
 $(\text{INEV } I_p (\text{IF } (\text{NOT } (\text{HOLDS } \text{pnf } ?i))$
 $(\text{IFTRIED } ?pi (\text{NOT } (\text{HOLDS } \text{pnf } ?i))))))$
<assumption>
- 4) (IF(NOT (HOLDS $\text{pnf } I$))
 $(\text{IFTRIED}(\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I)$
 $(\text{NOT } (\text{HOLDS } \text{pnf } I))))$
<assumption>
- 5) 4 \vdash (IF (NOT (IFTRIED (COMP (carry $sc1$)@I (carry $sc2$)@I)
 $(\text{NOT } (\text{HOLDS } \text{pnf } I))))$
 $(\text{NOT } (\text{NOT } (\text{HOLDS } \text{pnf } I))))$
<MP applied to 4, TRSNP>
- 6) 4 \vdash (IF (NOT (IFTRIED (COMP (carry $sc1$)@I (carry $sc2$)@I)
 $(\text{NOT } (\text{HOLDS } \text{pnf } I))))$
 $(\text{HOLDS } \text{pnf } I))$
<SUBST applied to 5 using equivalence NNP-P>
- 7) 4 \vdash (IF (IFTRIED (COMP (carry $sc1$)@I (carry $sc2$)@I) (HOLDS $\text{pnf } I$))
 $(\text{HOLDS } \text{pnf } I))$
<MP-TRNS applied to AX-IFTR1,6>
- 8) 1 \vdash (PRIOR $I_p (\text{TIME-OF } (\text{carry } sc1)@I))$
<MP applied to 1, (MP applied to $\lceil (= I (\text{TIME-OF } (\text{carry } sc1)@I) \rceil, \text{AX-FO7}) \rceil$ >
- 9) 1 \vdash (PRIOR $I_p (\text{TIME-OF } (\text{carry } sc2)@I))$
<MP applied to 1, (MP applied to $\lceil (= I (\text{TIME-OF } (\text{carry } sc2)@I) \rceil, \text{AX-FO7}) \rceil$ >

Proof continued on next page

Continuation of proof of TH-CR2

- 10) $1 \vdash (\text{PRIOR } I_p (\text{TIME-OF } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I))$
<MP applied (AND-INTRO applied to 8,9). (IF part of TH-APG-1)>
- 11) $1,2 \vdash (\text{IFTRIED } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I) \text{ CR-S1})$
<MP applied to 2. (MP applied to 10.AX-IFTR5)>
- 12) $1,2 \vdash (\text{IF } (\text{IFTRIED } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I) (\text{AND } (\text{OCC } (\text{carry } sc1)@I) (\text{OCC } (\text{carry } sc2)@I))) (\text{IFTRIED } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I) (\text{HOLDS } pnf\ I)))$
<MP applied to 11.AX-IFTR2>
- 13) $1,2,4 \vdash (\text{IF } (\text{EXECUTABLE } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I) (\text{HOLDS } pnf\ I))$
<MP-TRANS applied to 12,7 and EXECUTABLE substituted for its definition>
- 14) $1 \vdash (\text{IF } (\text{INEV } I_p (\text{AND } (\text{INEV } I_p \text{ CR-S1}) (\text{IF } (\text{NOT } (\text{HOLDS } pnf\ I)) (\text{IFTRIED } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I) (\text{NOT } (\text{HOLDS } pnf\ I)))))) (\text{INEV } I_p (\text{IF } (\text{EXECUTABLE } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I) (\text{HOLDS } pnf\ I))$
<DRL-APG-1 applied to 13 after discharging assumptions 2 and 4>
- 15) $1 \vdash (\text{IF } (\text{AND } (\text{INEV } I_p \text{ CR-S1}) (\text{INEV } I_p (\text{IF } (\text{NOT } (\text{HOLDS } pnf\ I)) (\text{IFTRIED } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I) (\text{NOT } (\text{HOLDS } pnf\ I)))))) (\text{INEV } I_p (\text{IF } (\text{EXECUTABLE } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I) (\text{HOLDS } pnf\ I))$
<DRL-IFTR8 applied to twice to 14, first using equivalence TH-INV-SF9 and then using equivalence TH-INV-SF5>
- 16) $3 \vdash (\text{INEV } I_p (\text{IF } (\text{NOT } (\text{HOLDS } pnf\ I)) (\text{IFTRIED } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I) (\text{NOT } (\text{HOLDS } pnf\ I))))$
<MP applied to 3.AX-FO5 for the variable substitutions $\{(COMP (\text{carry } sc1)@I (\text{carry } sc2)@I)/?pi\}$ I/?i>
- 17) $\vdash (\text{IF } (\text{PRIOR } I_p\ I) (\text{IF } (\text{AND } (\text{INEV } I_p \text{ CR-S1}) (\forall ?i\ ?pi (\text{INEV } I_p (\text{IF } (\text{NOT } (\text{HOLDS } pnf\ ?i)) (\text{IFTRIED } ?pi (\text{NOT } (\text{HOLDS } pnf\ ?i)))))) (\text{INEV } I_p (\text{IF } (\text{EXECUTABLE } (\text{COMP } (\text{carry } sc1)@I (\text{carry } sc2)@I) (\text{HOLDS } pnf\ I))$
<Discharging assumptions 2 and 3 then 1 after (MP applied to (AND-INTRO 2,17).16)>
- QED (TH-CR2)

Appendix H

Proof that the Algorithm is Sound

Proof for NON-OVRLP and proof showing that algorithm is sound with respect to the semantics

See introduction to Appendix F for conventions used in presenting proofs.

NON-OVRLP)

\vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
(INEV i NI-CND(pi1,pi2)))

Note: see p. 264 for definition of NI-CND(pi1,pi2)

Lemma1)

\vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
(IF (AND (EXECUTABLE pi1)
(IFTRIED pi1 (EXECUTABLE pi2)))
(IF (OCC pi2) (IFTRIED pi1 (OCC pi2)))))

Proof of Lemma1

- 1) (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 <assumption>
- 2) (EXECUTABLE pi1)
 <assumption>
- 3) (IFTRIED pi1 (EXECUTABLE pi2))
 <assumption>
- 4) (OCC pi2)
 <assumption>
- 5) (OCC pi1)
 <assumption>
- 6) (EXECUTABLE pi2)
 <assumption>
- 7) \vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 (IF (OCC pi1) (IFTRIED pi2 (OCC pi1))))
 <TH-IFTR-IN1 with substitutions [pi2/pi, (OCC pi1)/ILS]>
- 8) 1,5 \vdash (IFTRIED pi2 (OCC pi1))
 <MP applied to 5, (MP applied to 1,7)>
- 9) 1,5,6 \vdash (IFTRIED pi2 (AND (OCC pi2) (OCC pi1)))
 <DRL-IFTR5 applied to 6,8 after replacing EXECUTABLE by its definition in 6>
- 10) \vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 (IF (AND (OCC pi1) (EXECUTABLE pi2))
 (IFTRIED pi2 (AND (OCC pi2) (OCC pi1)))))
 <discharging assumptions 5 and 6, then assumption 1 in 9>

Proof of NON-OVRLP (Lemma1) continued on next page

Continuation of proof of NON-OVRLP (Lemma1)

- 11) \vdash (IF (IFTRIED pi1 (PRIOR (TIME-OF pi1) (TIME-OF pi2)))
 (IFTRIED pi1 (IF (AND (OCC pi1) (EXECUTABLE pi2))
 (IFTRIED pi2 (AND (OCC pi2) (OCC pi1))))))

<DRL-IFTR1 applied to 10>

- 12) \vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 (IFTRIED pi1 (IF (AND (OCC pi1) (EXECUTABLE pi2))
 (IFTRIED pi2 (AND (OCC pi2) (OCC pi1))))))

<DRL-IFTR3 applied to 11 using equivalence TH-IFTR-INS>

- 13) 1 \vdash (IF (IFTRIED pi1 (AND (OCC pi1) (EXECUTABLE pi2)))
 (IFTRIED pi1 (IFTRIED pi2 (AND (OCC pi2) (OCC pi1)))))

<MP applied to (MP applied to 1,12), AX-IFTR2>

- 14) 1-3 \vdash (IFTRIED pi1 (IFTRIED pi2 (AND (OCC pi2) (OCC pi1))))
 <MP applied to (DRL-IFTR5 applied to 2,3 after EXECUTABLE replaced by its definition in 2), 13>

- 15) 4 \vdash (IF (OCC pi2) (IFTRIED pi2 (OCC pi2)))
 <ONLY-IF part of (MP applied to 4, (AX-IFTR3 with substitutions [pi2/pi, (OCC pi2)/P]))>

- 16) 4 \vdash (EXECUTABLE pi2)
 <MP applied to 4, 15 and EXECUTABLE substituted for its definition>

- 17) 1-4 \vdash (IFTRIED pi2 (IFTRIED pi1 (AND (OCC pi2) (OCC pi1))))
 <MP applied to 14, (MP applied to 16, (AX-IFTR9 with substitutions [pi2/pi1, pi1/pi2]))>

- 18) 4 \vdash (IF (IFTRIED pi2 (IFTRIED pi1 (AND (OCC pi2) (OCC pi1))))
 (IFTRIED pi1 (AND (OCC pi2) (OCC pi1))))
 <IF part of (MP applied to 4, (AX-IFTR9 with substitutions [pi2/pi, (IFTRIED pi1 (AND (OCC pi2) (OCC pi1)))/P]))>

- 19) 1-4 \vdash (IFTRIED pi1 (AND (OCC pi2) (OCC pi1)))
 <MP applied to 17, 18>

- 20) 1-4 \vdash (AND (IFTRIED pi1 (OCC pi2))
 (IFTRIED pi1 (OCC pi1)))
 <DRL-IFTR3 applied to 19 using equivalence TH-IFTR-SC1>

- 21) 1-3 \vdash (IF (OCC pi2) (IFTRIED pi1 (OCC pi2)))
 <Discharging assumption 4 after AND-ELIM applied to 20>

- 22) \vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 (IF (AND (EXECUTABLE pi1)
 (IFTRIED pi1 (EXECUTABLE pi2)))
 (IF (OCC pi2) (IFTRIED pi1 (OCC pi2))))
 <discharging assumptions 2 and 3, then assumption 1 in 21>

QED (Lemma1)

Proof of NON-OVRLP continued on next page

Continuation of proof of NON-OVRLP

Proof of NON-OVRLP

- 1) (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 $\langle \text{assumption} \rangle$
- 2) (EXECUTABLE pi1)
 $\langle \text{assumption} \rangle$
- 3) (EXECUTABLE pi2)
 $\langle \text{assumption} \rangle$
- 4) (IFTRIED pi2 (EXECUTABLE pi1))
 $\langle \text{assumption} \rangle$
- 5) (IFTRIED pi1 (EXECUTABLE pi2))
 $\langle \text{assumption} \rangle$
- 6) \vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 (IF (OCC pi1) (IFTRIED pi2 (OCC pi1))))
 $\langle \text{TH-IFTR-INV1 with substitutions } [pi2/pi, (OCC pi1)/ILS] \rangle$
- 7) 1 \vdash (IF (OCC pi1) (IFTRIED pi2 (OCC pi1)))
 $\langle \text{MP applied to 1.6} \rangle$
- 8) 1.4 \vdash (IF (OCC pi1) (IFTRIED pi2 (OCC pi1)))
 $\langle \text{ANTCD-INTRO applied to 7} \rangle$
- 9) 1 \vdash (IF (IFTRIED pi2 (EXECUTABLE pi1))
 (IF (OCC pi1) (IFTRIED pi2 (OCC pi1))))
 $\langle \text{discharging assumption 4} \rangle$
- 10) 1.2.5 \vdash (IF (OCC pi2) (IFTRIED pi1 (OCC pi2)))
 $\langle \text{MP applied to (AND-INTRO applied to 2.5), (MP applied to 1, Lemma1)} \rangle$
- 11) 1.2 \vdash (IF (IFTRIED pi1 (EXECUTABLE pi2))
 (IF (OCC pi2) (IFTRIED pi1 (OCC pi2))))
 $\langle \text{discharging assumption 5 in 10} \rangle$
- 12) 1-3 \vdash (AND (IF (IFTRIED pi2 (EXECUTABLE pi1))
 (IF (OCC pi1) (IFTRIED pi2 (OCC pi1))))
 (IF (IFTRIED pi1 (EXECUTABLE pi2))
 (IF (OCC pi2) (IFTRIED pi1 (OCC pi2)))))
 $\langle \text{ANTCD-INTRO (introducing 9) applied to (AND-INTRO applied to 9,11)} \rangle$
- 13) 1 \vdash NI-CND(pi1,pi2)
 $\langle \text{substituting NI-CND for its definition after discharging assumptions 2 and 9 in 12} \rangle$
- 14) \vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 NI-CND(pi1,pi2))
 $\langle \text{discharging assumption 1 in 13} \rangle$
- 15) \vdash (IF (INEV i (PRIOR (TIME-OF pi1) (TIME-OF pi2)))
 (INEV i NI-CND(pi1,pi2)))
 $\langle \text{DRL-INV1 applied to 14} \rangle$
- 16) \vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 (INEV i NI-CND(pi1,pi2)))
 $\langle \text{DRL-INV3 applied to 15 using equivalence TH-INV-IL4} \rangle$

QED (NON-OVRLP)

Proof that the Algorithm is Sound with Respect to the Logic

We make use of the following definitions:

$$\begin{aligned} \text{PI-DISJOINT}(\text{pix}, \text{piy}) &=_{\text{def}} \\ &(\text{OR} (\text{PRIOR} (\text{TIME-OF } \text{pix}) (\text{TIME-OF } \text{piy})) \\ &(\text{PRIOR} (\text{TIME-OF } \text{piy}) (\text{TIME-OF } \text{pix}))) \end{aligned}$$

$$\text{CMP}(\text{pi}) =_{\text{def}} \text{pi}$$

$$\begin{aligned} \text{CMP}(\text{pi}_1, \dots, \text{pi}_i) &=_{\text{def}} \\ &\text{CMP}(\text{pi}_1, \dots, \text{pi}_{i-2}, (\text{COMP } \text{pi}_{i-1} \text{pi}_i)) \end{aligned}$$

$$\begin{aligned} \text{NI-CND}(\text{pix}, \text{piy}) &=_{\text{def}} \\ &(\text{IF} (\text{AND} (\text{EXECUTABLE } \text{pix}) (\text{EXECUTABLE } \text{piy})) \\ &(\text{AND} (\text{IF} (\text{IFTRIED } \text{piy} (\text{EXECUTABLE } \text{pix})) \\ &(\text{IF} (\text{OCC } \text{pix}) (\text{IFTRIED } \text{piy} (\text{OCC } \text{pix})))) \\ &(\text{IF} (\text{IFTRIED } \text{pix} (\text{EXECUTABLE } \text{piy})) \\ &(\text{IF} (\text{OCC } \text{piy}) (\text{IFTRIED } \text{pix} (\text{OCC } \text{piy})))))) \end{aligned}$$

$$\begin{aligned} \text{EXC-NI}(\text{pix}, \text{piy}) &=_{\text{def}} \\ &(\text{IF} (\text{EXECUTABLE } \text{piy}) \\ &(\text{IF} (\text{OCC } \text{pix}) (\text{IFTRIED } \text{piy} (\text{OCC } \text{pix})))) \end{aligned}$$

$$\begin{aligned} \text{GEN-ST}(i, \text{pi}, \text{CG}) &=_{\text{def}} \\ &(\text{AND} (\text{PRIOR } i (\text{TIME-OF } \text{pi})) \\ &(\text{INEV } i (\text{EXECUTABLE } \text{pi})) \\ &(\text{INEV } i (\text{IF} (\text{OCC } \text{pi}) \text{CG}))) \end{aligned}$$

$$\begin{aligned} \text{GEN-ST}(i, \text{pi}, \text{CG}, \text{pi}_1, \dots, \text{pi}_i) &=_{\text{def}} \\ &(\text{AND } \text{GEN-ST}(i, \text{pi}, \text{CG}, \text{pi}_1, \dots, \text{pi}_{i-1}) \\ &(\text{INEV } i_p (\text{EXC-NI}(\text{pi}, \text{pi}_i)))) \end{aligned}$$

$$\begin{aligned} \text{NEW-CG}(\text{CG}, \text{pi}, \text{C}) &=_{\text{def}} \\ &(\text{AND} (\text{CG} - \text{C}) \text{EC}(\text{pi})) \end{aligned}$$

$$\begin{aligned} \text{NEW-CG}(\text{CG}, \text{pi}, \text{C}, \text{pi}_1, \dots, \text{pi}_i) &=_{\text{def}} \\ &(\text{AND } \text{NEW-CG}(\text{CG}, \text{pi}, \text{C}, \text{pi}_1, \dots, \text{pi}_{i-1}) \text{ ;if } \text{OVRLP}(\text{pi}, \text{pi}_i) \\ &\text{NI}(\text{pi}, \text{pi}_i) \\ &\text{NEW-CG}(\text{CG}, \text{pi}, \text{C}, \text{pi}_1, \dots, \text{pi}_{i-1}) \quad \text{ ;otherwise} \end{aligned}$$

The description of the planning problem (S)

An input to the algorithm is given by
 $\langle G, PE, PI-SET, OVRLP, EC, EFF, NI \rangle$ where

G	is a member of ILS (the set of interval logic sentences), that specifies the goal
PE	is a subset of ILS, that specifies the planning environment
PI-SET	is a set of terms denoting the simple plan instance that can be entered into the plan
OVRLP	is a relation defined on $PI-SET \times PI-SET$ that holds for any pair of simple plan instances that overlap in time
EC	is a function from $PI-SET$ to ILS, that specifies the executability conditions for each simple plan instance
EFF	is a function from $PI-SET$ to ILS, that specifies the effects for each simple plan instance
NI	is a partial function from $PI-SET \times PI-SET$ to ILS defined for all pairs $\langle pi1, pi2 \rangle$ when $OVRLP(pi1, pi2)$ holds, that specifies the non-interference conditions between pairs of overlapping plan instances. We assume that for any overlapping plan instances (pi1 and pi2), $NI(pi1, pi2) = NI(pi2, pi1)$.

In the proofs, we assume that S is the set of sentences corresponding to the planning problem $\langle G, PE, PI-SET, OVRLP, EC, EFF, NI \rangle$ given as input. S is defined as follows:

$$\begin{aligned}
 S =_{\text{def}} & \{ (PRIOR \text{ } I_p \text{ } (TIME-OF \text{ } pi)) \mid pi \in PI-SET \} \cup \\
 & \{ (INEV \text{ } I_p \text{ } il) \mid il \in PE \} \cup \\
 & \{ (INEV \text{ } I_p \text{ } (IF \text{ } EC(pi) \text{ } (EXECUTABLE \text{ } pi))) \mid pi \in PI-SET \} \cup \\
 & \{ (INEV \text{ } I_p \text{ } (IF \text{ } (OCC \text{ } pi) \text{ } EFF(pi))) \mid pi \in PI-SET \} \cup \\
 & \{ (INEV \text{ } I_p \text{ } (IF \text{ } NI(pi1, pi2) \text{ } NI-CND(pi1, pi2))) \\
 & \quad \mid OVRLP(pi1, pi2) \} \cup \\
 & \{ PI-DISJOINT(pi1, pi2) \mid OVRLP(pi1, pi2) \text{ is false} \}
 \end{aligned}$$

MAIN-PROOF)

If the sequence of operators $op1: \dots : opn$ is applicable in the initial state and the result of applying the sequence leads to any empty causal gap and $INPLAN = \{pi1, \dots, pii\}$
then $S \models (AND (INEV Ip (EXECUTABLE CMP(pi1, \dots, pii)))$
 $(INEV Ip (IF (OCC CMP(pi1, \dots, pii)) G)))$

where $i \geq 1$

We use j to refer to the integer having the property that opj is the last INTRO operator in the sequence $op1: \dots : opn$ (which introduces plan instance pii). Since we are assuming that there is at least one INTRO operator in the sequence, such a j exists.

We let CG_{j-1} refer to the causal gap that is produced by applying the sequence $op1: \dots : opj-1$ (which might be empty) to the initial state. Associated with CG_{j-1} is $pi1, \dots, pi_{j-1}$, which are the plan instances that have been entered when CG_{j-1} is reached.

We let CG_j refer to the causal gap produced by applying INTRO using pii to remove C_j from CG_{j-1} . Since $pi1, \dots, pi_{j-1}$ have been entered when CG_j is reached, $CG_j = NEW-CG(CG_{j-1}, pi_j, CG_j, pi_1, \dots, pi_{j-1})$.

Proof of MAIN-PROOF

- 1) The sequence of operators $op1: \dots : opn$ is applicable in the initial state and the result of applying the sequence leads to any empty causal gap and $INPLAN = \{pi1, \dots, pii\}$
<assumption>
- 2) $1 \rightarrow$ INTRO using pii can be applied to remove C_j from causal gap CG_{j-1}
<using 1 and the definition of j >
- 3) $1 \rightarrow$ An applicable sequence of zero or more REMOVES applied to causal gap CG_j leads to an empty causal gap
<using 1 and the fact that opj is the last INTRO operator in $op1: \dots : opn$, and thus $opj+1; \dots : opn$ are all REMOVES>
- 4) $1 \rightarrow S \models GEN-ST(Ip, pii, CG_{j-1}, pi1, \dots, pi_{j-1})$
<TERM-COND applied to 2,3>
- 5) $1 \rightarrow S \models (AND (INEV Ip (EXECUTABLE CMP(pi1, \dots, pii)))$
 $(INEV Ip (IF (OCC CMP(pi1, \dots, pii)) G)))$
<REDUCTIONS applied to 4 with substitutions $[pii/pi, j-1/k, i-1/m]$ using the definition of CG_{j-1} >

QED (MAIN-PROOF)

TERM-COND)

If INTRO using pi_i can be applied to remove C_j
 from causal gap CG_{j-1}
 and an applicable sequence of zero or more REMOVEs applied to
 causal gap CG_j leads to an empty causal gap
 then $S \models \text{GEN-ST}(Ip.pi_i.CG_{j-1}.pi_1, \dots, pi_{i-1})$
 where $CG_j = \text{NEW-CG}(CG_{j-1}.pi_i.C_j.pi_1, \dots, pi_{i-1})$

Proof of TERM-COND

- 1) INTRO using pi_i can be applied to remove C_j from causal gap CG_{j-1}
<assumption>
- 2) An applicable sequence of zero or more REMOVEs applied to causal gap CG_j (which equals $\text{NEW-CG}(CG_{j-1}.pi_i.C_j.pi_1, \dots, pi_{i-1})$) leads to an empty causal gap
<assumption>
- 3) $1,2 \rightarrow S \vdash \text{GEN-ST}(Ip.pi_i.CG_{j-1}.pi_1, \dots, pi_{i-1})$
<TERM-COND applied to 1,2>*
- 4) $1,2 \rightarrow S \models \text{GEN-ST}(Ip.pi_i.CG_{j-1}.pi_1, \dots, pi_{i-1})$
<using 3 and the fact that ' $S \models X$ ' is true if ' $S \vdash X$ ' is true (our proof theory is sound)>

QED (TERM-COND)

TERM-COND*)

If INTRO using pi_i can be applied to remove C_j
 from causal gap CG_{j-1}
 and an applicable sequence of zero or more REMOVEs applied to
 causal gap CG_j leads to an empty causal gap
 then $S \vdash \text{GEN-ST}(Ip.pi_i.CG_{j-1}.pi_1, \dots, pi_{i-1})$
 where $CG_j = \text{NEW-CG}(CG_{j-1}.pi_i.C_j.pi_1, \dots, pi_{i-1})$

Proof of TERM-COND* by induction on the number of
 plan instances in $\{pi_1, \dots, pi_{i-1}\}$

Proof of base case: the set is empty ($i=1$)

- 1) INTRO using pi_1 can be applied to remove C_j from causal gap CG_{j-1}
<assumption>
- 2) An applicable sequence of zero or more REMOVEs applied to causal gap CG_j
 (which equals $\text{NEW-CG}(CG_{j-1}.pi_1.C_j)$) leads to an empty causal gap
<assumption>

Proof continued on next page

Continuation of TERM-COND* (base case)

- 3) $(\text{PRIOR } I_p (\text{TIME-OF } \pi_1))$
 $\langle \text{assumption} \rangle$
 - 4) $(\text{INEV } I_p (\text{IF } \text{EC}(\pi_1) (\text{EXECUTABLE } \pi_1)))$
 $\langle \text{assumption} \rangle$
 - 5) $S \vdash (\text{PRIOR } I_p (\text{TIME-OF } \pi_1))$
 $\langle \text{ANTCD-INTRO applied to 3 (applicable since } \{(\text{PRIOR } I_p (\text{TIME-OF } \pi_1))\} \text{ is a subset of } S \rangle$
 - 6) $2 \rightarrow \vdash (\text{INEV } I_p \text{EC}(\pi_1))$
 $\langle \text{TAIL-RMVS applied to 2 (EC}(\pi_1) \text{ is a conjunct in } \text{CG}_j) \rangle$
 - 7) $2 \rightarrow 4 \vdash (\text{INEV } I_p (\text{EXECUTABLE } \pi_1))$
 $\langle \text{DRL-IFTR}_4 \text{ applied to 6,4} \rangle$
 - 8) $2 \rightarrow S \vdash (\text{INEV } I_p (\text{EXECUTABLE } \pi_1))$
 $\langle \text{ANTCD-INTRO applied to 7 (applicable since } \{(\text{INEV } I_p (\text{IF } \text{EC}(\pi_1) (\text{EXECUTABLE } \pi_1)))\} \text{ is a subset of } S \rangle$
 - 9) $2 \rightarrow \vdash (\text{INEV } I_p (\text{CG}_{j-1} - C_j))$
 $\langle \text{TAIL-RMVS applied to 2 ((CG}_{j-1} - C_j) \text{ is a conjunct in } \text{CG}_j) \rangle$
 - 10) $1 \rightarrow \{\text{EFF}(\pi_1)\} \cup \text{PE} \vdash C_j$
 $\langle \text{follows from 1 since INTRO using } \pi_1 \text{ to remove } C_j \text{ is applicable} \rangle$
 - 11) $1.2 \rightarrow S \vdash (\text{INEV } I_p (\text{IF } (\text{OCC } \pi_1) \text{CG}_{j-1}))$
 $\langle \text{ANTCD-INTRO applied to (MP applied to 9, (lemma1 applied to 10 with substitutions } [\pi_1/\pi, \text{CG}_{j-1}/\text{CG}, C_j/C]) \text{ (note: ANTCD-INTRO is applicable since S-EF (mentioned in lemma1) is a subset of } S \rangle$
 - 12) $1.2 \rightarrow S \vdash (\text{AND } (\text{PRIOR } I_p (\text{TIME-OF } \pi_1))$
 $(\text{INEV } I_p (\text{EXECUTABLE } \pi_1))$
 $(\text{INEV } I_p (\text{IF } (\text{OCC } \pi_1) \text{CG}_{j-1}))$
 $\langle \text{AND-INTRO applied to 5,8,11} \rangle$
 - 13) $1.2 \rightarrow S \vdash \text{GEN-ST}(I_p, \pi_1, \text{CG}_{j-1})$
 $\langle \text{substituting GEN-ST for its definition in 12} \rangle$
- QED (base case)

Proof continued on next page

Continuation of TERM-COND*

Proof of inductive step

Assume that TERM-COND* holds if the size of $\{pi_1, \dots, pi_{i-1}\}$ is less than k where k is a positive integer greater than 1. We prove that TERM-COND* holds when the size of $\{pi_1, \dots, pi_{i-1}\}$ is k ($i=k$).

Lemma1)

From: $\{EFF(pi)\} \cup PE \vdash_{IL} C$
 To: $S-EF \vdash (IF (INEV Ip (CG - C))$
 $(INEV Ip (IF (OCC pi) CG)))$

where $S-EF =_{def} \{(INEV i p) \mid p \in PE\}$
 $\cup \{(INEV i (IF (OCC pi) EFF(pi)))\}$

Proof of Lemma1

- 1) $PE \cup \{EFF(pi)\} \vdash_{IL} C$
<assumption for proof of rule>
- 2) $1 \rightarrow PE \vdash (IF EFF(pi) C)$
<discharging assumption $\{EFF(pi)\}$ in 1 and using fact that if ' $A \vdash_{IL} B$ ' is true true then ' $A \vdash B$ ' is true (if B is derivable from A in the interval logic fragment then B is derivable from A using all axioms and inference rules)>
- 3) $(IF (OCC pi) EFF(pi))$
<assumption>
- 4) $1 \rightarrow PE \cup 3 \vdash (IF (OCC pi) C)$
<MP-TRANS applied to 3,2>
- 5) $(OCC pi)$
<assumption>
- 6) $(CG - C)$
<assumption>
- 7) $1 \rightarrow PE \cup 3, 5 \vdash C$
<MP applied to 5,4>
- 8) $\vdash (IF (AND C (CG - C)) CG)$
<property of $(P - Q)$ >
- 9) $1 \rightarrow PE \cup 3, 5, 6 \vdash CG$
<MP applied to (AND-INTRO applied to 7,6),8>
- 10) $1 \rightarrow PE \cup 3 \vdash (IF (CG - C) (IF (OCC pi) CG))$
<assumption 5 then 6 discharged in 9>
- 11) $1 \rightarrow S-EF \vdash (INEV Ip (IF (CG - C) (IF (OCC pi) CG)))$
<DRL-IFTR8 applied $|PE|$ times using equivalence TH-INV-SF9 to (DRL-IFTR1 applied to 10 after assumptions discharged)>
- 12) $1 \rightarrow S-EF \vdash (IF (INEV Ip (CG - C)) (INEV (IF (OCC pi) CG)))$
<MP applied to 11, A.I-INV2>

QED (lemma1)

Proof continued on next page

Continuation of TERM-COND* (inductive step (Lemma1))

The main proof for the inductive step

For convenience, we let CG_j^* refer to
 $NEW-CG(CG_{j-1}, \pi_k, C_j, \pi_1, \dots, \pi_{k-2})$

- 1) INTRO using π_k can be applied to remove C_j from causal gap CG_{j-1}
<assumption>
- 2) An applicable sequence of zero or more REMOVEs applied to causal gap CG_j
 (which equals $NEW-CG(CG_{j-1}, \pi_k, C_j, \pi_1, \dots, \pi_{k-1})$) leads to an empty causal gap
<assumption>
- 3) $2 \rightarrow$ An applicable sequence of REMOVEs applied to causal gap CG_j^* leads to
 an empty causal gap
<since every conjunct in CG_j^ is also in CG_j , we can form a sequence of
 REMOVEs for CG_j^* by deleting from the sequence of REMOVEs meeting 2 the
 ones that remove conjuncts that do not appear in CG_j^* >*
- 4) $1, 2 \rightarrow S \vdash GEN-ST(Ip, \pi_k, CG_{j-1}, \pi_1, \dots, \pi_{k-2})$
<the inductive hypothesis applied to 1, 2>
- 5) π_k and π_{k-1} overlap
<assumption>
- 6) $2, 5 \rightarrow S \vdash (INEV Ip NI(\pi_k, \pi_{k-1}))$
*<TAIL-REMS applied to 2 (if π_k and π_{k-1} overlap then $NI(\pi_k, \pi_{k-1})$ is a
 conjunct in CG_j)>*
- 7) $\{(PRIOR Ip (TIME-OF \pi_{k-1})), (INEV Ip (IF NI(\pi_k, \pi_{k-1}) NI-
 CND(\pi_k, \pi_{k-1})))\}$
 $\vdash (IF (INEV Ip NI(\pi_k, \pi_{k-1}))$
 $(IF (INEV Ip (EXECUTABLE \pi_k))$
 $(INEV Ip EXC-NI(\pi_k, \pi_{k-1}))))$
<TERM-COND-OVRLP with substitutions $\pi_{k-1}/\pi_k, \pi_k/\pi_{k-1}$ >
- 8) $5 \rightarrow S \vdash (IF (INEV Ip NI(\pi_k, \pi_{k-1}))$
 $(IF (INEV Ip (EXECUTABLE \pi_k))$
 $(INEV Ip EXC-NI(\pi_k, \pi_{k-1}))))$
*<ANTCD-INTRO applied to 7 and using 5 (applicable since $\{(PRIOR Ip (TIME-OF$
 $\pi_{k-1})), (INEV Ip (IF NI(\pi_k, \pi_{k-1}) NI-CND(\pi_k, \pi_{k-1})))\}$ is a subset of S, the
 latter because it is assumed that π_k and π_{k-1} overlap>*

Proof continued on next page

Continuation of TERM-COND* (inductive step)

- 9) $2.5 \rightarrow S \vdash (\text{IF } (\text{INEV } I_p (\text{EXECUTABLE } \pi_k))$
 $(\text{INEV } I_p \text{ EXC-NI}(\pi_k, \pi_{k-1}))$
<MP applied to 6,8>
- 10) $1.2 \rightarrow S \vdash (\text{INEV } I_p (\text{EXECUTABLE } \pi_k))$
<AND-ELIM applied to 4 after GEN-ST replaced with its definition>
- 11) $1.2.5 \rightarrow S \vdash (\text{INEV } I_p \text{ EXC-NI}(\pi_k, \pi_{k-1}))$
<MP applied to 10,9>
- 12) π_k and π_{k-1} do not overlap
<assumption>
- 13) $\{(\text{PRIOR } I_p (\text{TIME-OF } \pi_{k-1})), \text{PI-DISJOINT}(\pi_{k-1}, \pi_k)\}$
 $\vdash (\text{IF } (\text{INEV } I_p (\text{EXECUTABLE } \pi_k))$
 $(\text{INEV } I_p \text{ EXC-NI}(\pi_k, \pi_{k-1})))$
<TERM-COND-DISJ with substitutions $\pi_k/\pi_{k-1}, \pi_{k-1}/\pi_k$ >
- 14) $12 \rightarrow S \vdash (\text{IF } (\text{INEV } I_p (\text{EXECUTABLE } \pi_k))$
 $(\text{INEV } I_p \text{ EXC-NI}(\pi_k, \pi_{k-1})))$
<ANTCD-INTRO applied to 13 and using 12 (applicable since $\{(\text{PRIOR } I_p (\text{TIME-OF } \pi_{k-1})), \text{PI-DISJOINT}(\pi_{k-1}, \pi_k)\}$ is a subset of S, the latter because it is assumed that π_k and π_{k-1} do not overlap)>
- 15) $1,2,12 \rightarrow S \vdash (\text{INEV } I_p \text{ EXC-NI}(\pi_k, \pi_{k-1}))$
<MP applied to 10,14>
- 16) $1,2 \rightarrow S \vdash (\text{INEV } I_p \text{ EXC-NI}(\pi_k, \pi_{k-1}))$
<using 11, 15, and fact that either 5 or 12 must hold>
- 17) $1,2 \rightarrow S \vdash \text{GEN-ST}(I_p, \pi_k, \text{CG}_{j-1}, \pi_1, \dots, \pi_{k-1})$
<AND-INTRO applied to 4,16 and then GEN-ST substituted for its definition>

QED (inductive step)

QED (TERM-COND*)

TAIL-RMVS)

If an applicable sequence of k ($k \geq 0$) REMOVES applied to
causal gap CG leads to an empty causal gap
then $S \vdash (\text{INEV } I_p C)$ for each conjunct C in the causal gap CG

Proof of TAIL-RMVS by induction on k

Proof of base case: $k=0$

If $k=0$ then there are no conjuncts in CG and thus ' $S \vdash (\text{INEV } I_p C)$ for each
conjunct C in the causal gap CG' trivially holds.

QED (Base case)

Proof continued on next page

Continuation of TAIL-RMVS

Proof of Inductive step

Assume that TAIL-RMVS holds for any $k < n$ where n is a positive integer. We prove that TAIL-RMVS holds for $k = n$.

1) An applicable sequence of n REMOVEs applied to causal gap CG leads to an empty causal gap

<assumption>

2) $1 \rightarrow$ An applicable sequence of $n-1$ REMOVEs applied to $(CG - C_1)$ leads to an empty causal gap where C_1 is the conjunct removed by the first REMOVE in the sequence

<using 1 and the definition of an applicable sequence of REMOVEs>

3) $1 \rightarrow S \vdash (INEV \text{ Ip } C)$ for each conjunct C in $(CG - C_1)$

<The inductive hypothesis applied to 2>

4) $1 \rightarrow PE \vdash_{IL} C_1$

<follows from 1 because the first REMOVE, which removes C_1 , is applicable>

5) $1 \rightarrow PE \vdash C_1$

<follows from 4 and the fact that ' $S \vdash X$ ' is true if ' $S \vdash_{IL} X$ ' is true since \vdash_{IL} refers to the derivability relation in a fragment (the interval logic fragment) of our logic>

6) $1 \rightarrow \{(INEV \text{ i } p) \mid p \in PE\} \vdash (INEV \text{ Ip } C_1)$

<Lemma-ST1-ST2 applied to 5>

7) $1 \rightarrow S \vdash (INEV \text{ Ip } C_1)$

<ANTCD-INTRO applied to 6 (applicable since $\{(INEV \text{ i } p) \mid p \in PE\}$ is a subset of S)>

8) $1 \rightarrow S \vdash (INEV \text{ Ip } C)$ for each conjunct C in CG

<follows from 3, 7, and the fact that if C is in CG then $C = C_1$ or C is in $(CG - C_1)$ >

QED (inductive step)

QED (TAIL-RMVS)

TERM-COND-DISJ)

{(PRIOR I_p (TIME-OF pix)), PI-DISJOINT(pix,pi)}
 \vdash (IF (INEV I_p (EXECUTABLE pi))
 (INEV I_p EXC-NI(pi,pix)))

Lemma1)

(IF (IF A (IF (AND B C) D))
 (IF A (IF C (IF B D))))

Proof of Lemma1

- 1) (IF A (IF (AND B C) D))
<assumption>
- 2) A
<assumption>
- 3) B
<assumption>
- 4) C
<assumption>
- 5) 1,2 \vdash (IF (AND B C) D)
<MP applied to 2,1>
- 6) 1-4 \vdash D
<MP applied to (AND-INTRO applied to 3,4),5>
- 7) 1-4 \vdash (IF A (IF C (IF B D)))
<Discharging assumption 3 then 4 then 2 in 6>
- 8) \vdash (IF (IF A (IF (AND B C) D))
 (IF A (IF C (IF B D))))
<Discharging assumption 1 in 7>

QED (lemma1)

Proof continued on next page

Continuation of TERM-COND-DISJ

Proof of TERM-COND-DISJ

- 1) (PRIOR lp (TIME-OF pix))
<assumption>
- 2) PI-DISJOINT(pix, pi)
<assumption>
- 3) (INEV lp (EXECUTABLE pi))
<assumption>
- 4) (PRIOR (TIME-OF pix) (TIME-OF pi))
<assumption>
- 5) \vdash (IF (PRIOR (TIME-OF pix) (TIME-OF pi))
 (IF (AND (EXECUTABLE pix)
 (IFTRIED pix (EXECUTABLE pi))
 (IF (OCC pi) (IFTRIED pix (OCC pi))))))
<Lemma1 in NON-OVRLP with substitutions $[pix/pi1, pi/pi2]$ and substituting EXC-NI for its definition>
- 6) \vdash (IF (PRIOR (TIME-OF pix) (TIME-OF pi))
 (IF (IFTRIED pix (EXECUTABLE pi))
 (IF (EXECUTABLE pix)
 (IF (OCC pi) (IFTRIED pix (OCC pi))))))
<MP applied to 5. Lemma1>
- 7) \vdash (IF (INEV lp (PRIOR (TIME-OF pix) (TIME-OF pi)))
 (INEV lp (IF (IFTRIED pix (EXECUTABLE pi))
 EXC-NI(pi, pix))))
<DRL-IFTR1 applied to 6 and EXC-NI substituted for its definition>
- 8) \vdash (INEV lp (IF (IFTRIED pix (EXECUTABLE pi))
 EXC-NI(pi, pix)))
<MP applied to 4. (DRL-IFTR8 applied to 7 using equivalence TH-INV-IL4)>
- 9) \vdash (IF (INEV lp (IFTRIED pix (EXECUTABLE pi)))
 (INEV lp EXC-NI(pi, pix)))
<MP applied to 8. AX-INV2>
- 10) \vdash (IF (INEV lp (PRIOR lp (TIME-OF pix)))
 (INEV lp (IF (INEV lp (EXECUTABLE pi))
 (IFTRIED pix (EXECUTABLE pi))))))
<DRL-IFTR1 applied to AX-IFTR5 with substitutions $[lp/i, pix/pi, (EXECUTABLE pi)/P_i]$ >

Proof continued on next page

Continuation of TERM-COND-DISJ

- 11) $1 \vdash (\text{INEV } I_p (\text{IF } (\text{INEV } I_p (\text{EXECUTABLE } p_i)) (\text{IFTRIED } p_{ix} (\text{EXECUTABLE } p_i))))$
<MP applied to 1, (DRL-IFTR3 applied to 10 using equivalence TH-INV-IL4)>
 - 12) $1 \vdash (\text{IF } (\text{INEV } I_p (\text{EXECUTABLE } p_i)) (\text{INEV } I_p (\text{IFTRIED } p_{ix} (\text{EXECUTABLE } p_i))))$
<DRL-IFTR2 applied to (MP applied to 11, AX-INV2) using equivalence TH-INV-SF5>
 - 13) 1.3.4 $\vdash (\text{INEV } I_p \text{EXC-NI}(p_i, p_{ix}))$
<MP applied to (MP applied to 3.12), 9>
 - 14) 1.3 $\vdash (\text{IF } (\text{PRIOR } (\text{TIME-OF } p_{ix}) (\text{TIME-OF } p_i)) (\text{INEV } I_p \text{EXC-NI}(p_i, p_{ix})))$
<Discharging assumption 4 in 13>
 - 15) $(\text{PRIOR } (\text{TIME-OF } p_i) (\text{TIME-OF } p_{ix}))$
<assumption>
 - 16) $\vdash (\text{IF } (\text{PRIOR } (\text{TIME-OF } p_i) (\text{TIME-OF } p_{ix})) (\text{IF } (\text{OCC } p_i) (\text{IFTRIED } p_{ix} (\text{OCC } p_i))))$
<TH-IFTR-IN1 with substitutions $p_{ix}/p_i, (\text{OCC } p_i)/P_i>$
 - 17) 15 $\vdash (\text{IF } (\text{OCC } p_i) (\text{IFTRIED } p_{ix} (\text{OCC } p_i)))$
<MP applied to 15.16>
 - 18) 15 $\vdash (\text{IF } (\text{EXECUTABLE } p_{ix}) (\text{IF } (\text{OCC } p_i) (\text{IFTRIED } p_{ix} (\text{OCC } p_i))))$
<ANTCD-INTRO applied to 17>
 - 19) $\vdash (\text{IF } (\text{INEV } I_p (\text{PRIOR } (\text{TIME-OF } p_i) (\text{TIME-OF } p_{ix}))) (\text{INEV } I_p \text{NI-EXC}(p_i, p_{ix})))$
<DRL-IFTR1 applied to 18 after assumption discharged and NI-EXC substituted for its definition>
 - 20) $\vdash (\text{IF } (\text{PRIOR } (\text{TIME-OF } p_i) (\text{TIME-OF } p_{ix})) \text{NI-EXC}(p_i, p_{ix}))$
<DRL-IFTR3 applied to 19 using equivalence TH-INV-IL4>
 - 21) 2 $\vdash (\text{OR } (\text{PRIOR } (\text{TIME-OF } p_{ix}) (\text{TIME-OF } p_i)) (\text{PRIOR } (\text{TIME-OF } p_i) (\text{TIME-OF } p_{ix})))$
<Substituting PI-DISJOINT in 2 with its definition>
 - 22) 1-3 $\vdash (\text{INEV } I_p \text{NI-EXC}(p_i, p_{ix}))$
<MP applied to (MP applied to (AND-INTRO applied to 21, 14, 20), RCS), AX-FO1>
 - 23) $\{(\text{PRIOR } I_p (\text{TIME-OF } p_{ix})), \text{PI-DISJOINT}(p_{ix}, p_i)\}$
 $\vdash (\text{IF } (\text{INEV } I_p (\text{EXECUTABLE } p_i)) (\text{INEV } I_p \text{EXC-NI}(p_i, p_{ix})))$
<Discharging assumption 3 in 22>
- QED (TERM-COND-DISJ)

TERM-COND-OVRLP)

$$\{(PRIOR\ I_p\ (TIME-OF\ pix)),\ (INEV\ I_p\ (IF\ NI(pi,pix)\ NI-CND(pi,pix)))\}$$

$$\vdash (IF\ (INEV\ I_p\ NI(pi,pi:))$$

$$(IF\ (INEV\ I_p\ (EXECUTABLE\ pi))$$

$$(INEV\ I_p\ EXC-NI(pi,pix))))$$

Proof of TERM-COND-OVRLP

- 1) (PRIOR I_p (TIME-OF pix))
 <assumption>
- 2) (INEV I_p (IF $NI(pi,pix)\ NI-CND(pi,pix)$))
 <assumption>
- 3) (INEV $I_p\ NI(pi,pix)$)
 <assumption>
- 4) (INEV I_p (EXECUTABLE pi))
 <assumption>
- 5) (EXECUTABLE pix)
 <assumption>
- 6) $NI-CND(pi,pix)$
 <assumption>
- 7) (EXECUTABLE pi)
 <assumption>
- 8) (IFTRIED pix (EXECUTABLE pi))
 <assumption>
- 9) 5-7 \vdash (IF (IFTRIED pix (EXECUTABLE pi))
 (IF (OCC pi) (IFTRIED pix (OCC pi))))
 <AND-ELIM applied to (MP applied to (AND-INTRO applied to 7,5),6 after NI-CND
 substituted with its definition)>
- 10) 5-8 \vdash (IF (OCC pi) (IFTRIED pix (OCC pi)))
 <MP applied to 8,9>
- 11) 6-8 \vdash (IF (EXECUTABLE pix)
 (IF (OCC pi) (IFTRIED pix (OCC pi))))
 <Discharging assumption 5 in 10>

Proof continued on next page

Continuation of TERM-COND-OVRLP

- 12) \vdash (IF (AND NI-CND(pi,pix)
 (EXECUTABLE pi)
 (IFTRIED pix (EXECUTABLE pi)))
 EXC-NI(pi,pix))
<Discharging assumptions in 11 and substituting EXC-NI for its definition>
- 13) \vdash (IF (AND (INEV Ip NI-CND(pi,pix))
 (INEV Ip (EXECUTABLE pi))
 (INEV Ip (IFTRIED pix (EXECUTABLE pi)))
 (INEV Ip EXC-NI(pi,pix))
 <DRL-IFTR8 applied twice to (DRL-IFTR1 applied to 12) using equivalence TH-INV-SF9>
- 14) 2.3 \vdash (INEV Ip NI-CND(pi,pix))
<DRL-IFTR4 applied to 8.2>
- 15) \vdash (IF (INEV Ip (PRIOR Ip (TIME-OF pix)))
 (INEV Ip (IF (INEV Ip (EXECUTABLE pi))
 (IFTRIED pix (EXECUTABLE pi)))))
 <DRL-IFTR1 applied to AX-IFTR5 with substitutions [Ip/i, pix/pi, (EXECUTABLE pi)/P]>
- 16) 1 \vdash (INEV Ip (IF (INEV Ip (EXECUTABLE pi))
 (IFTRIED pix (EXECUTABLE pi))))
 <MP applied to 1, (DRL-IFTR8 applied to 15 using equivalence TH-INV-IL4)>
- 17) 1 \vdash (IF (INEV Ip (EXECUTABLE pi))
 (INEV Ip (IFTRIED pix (EXECUTABLE pi))))
 <DRL-IFTR8 applied to (MP applied to 16, AX-INV2) using equivalence TH-INV-SF5>
- 18) 1.4 \vdash (INEV Ip (IFTRIED pix (EXECUTABLE pi)))
 <MP applied to 4.17>
- 19) 1.4 \vdash (INEV Ip EXC-NI(pi,pix))
 <MP applied to (AND-INTRO applied to 14.4.18).18>
- 20) {(PRIOR Ip (TIME-OF pix)), (INEV Ip (IF NI(pi,pix) NI-CND(pi,pix)))}
 \vdash (IF (INEV Ip NI(pi,pix))
 (IF (INEV Ip (EXECUTABLE pi))
 (INEV Ip EXC-NI(pi,pix))))
 <Discharging assumption 4 then 8 in 19>

QED (TERM-COND-OVRLP)

REDUCTIONS)

If $S \models \text{GEN-ST}(Ip, pi, CG_k, pi_1, \dots, pi_m)$
 then $S \models (\text{AND} (\text{INEV } Ip (\text{EXECUTABLE } \text{CMP}(pi_1, \dots, pi_m, pi)))$
 $(\text{INEV } Ip (\text{IF} (\text{OCC} (\text{CMP}(pi_1, \dots, pi_m, pi) G))))$

where CG_k is the causal gap produced by applying the applicable sequence of operators $op_1; \dots; op_k$ to the initial causal gap CG_0 which is set to G ; this sequence includes the introduction of the plan instances pi_1, \dots, pi_m (in that order).

Proof of REDUCTIONS by induction on k

Proof of base case: $k=0$ (since $m \leq k$, m must equal 0)

1) $S \models \text{GEN-ST}(Ip, pi, CG_0)$

<assumption>

2) $1 \rightarrow S \vdash (\text{AND} (\text{INEV } Ip (\text{EXECUTABLE } \text{CMP}(pi)))$
 $(\text{INEV } Ip (\text{IF} (\text{OCC} (\text{CMP}(pi) G))))$

<AND-ELIM applied to 1 after replacing GEN-ST by its definition, substituting CMP for its definition, and using the fact that CG_0 is set to G >

QED (base case)

Proof of Inductive step

Assume that REDUCTIONS holds for $j < n$ where n is a positive integer. We prove that REDUCTIONS holds for $j=n$.

1) $S \models \text{GEN-ST}(Ip, pi, CG_n, pi_1, \dots, pi_m)$

<assumption>

2) the operator applied to CG_{n-1} to produce causal gap CG_n (i.e. op_n) is REMOVE_C

<assumption>

3) $2 \rightarrow PE \vdash_{IL} C$

<using 2 (since REMOVE_C is applicable)>

4) $2 \rightarrow CG_n$ is equivalent to $(CG_{n-1} - C)$

<using 2 and the fact that the causal gap produced by applying REMOVE_C to causal gap CG is $(CG - C)$ >

5) $1, 2 \rightarrow S \models \text{GEN-ST}(Ip, pi, (CG_{n-1} - C), pi_1, \dots, pi_m)$

<replacing CG_n in 1 with its equivalent form given in 4>

6) $1, 2 \rightarrow S \models \text{GEN-ST}(Ip, pi, CG_{n-1}, pi_1, \dots, pi_m)$

<REDUCTION-REMOVE applied to 3.5 using substitutions $\{CG_{n-1}/CG, m/i\}$ >

Proof continued on next page

Continuation of REDUCTIONS (inductive step)

- 7) $1,2 \rightarrow S \models (\text{AND} (\text{INEV } I_p (\text{EXECUTABLE } \text{CMP}(pi_1, \dots, pi_m, pi)))$
 $(\text{INEV } I_p (\text{IF} (\text{OCC} (\text{CMP}(pi_1, \dots, pi_m, pi) G))))$
<inductive hypothesis applied to 6>
- 8) the operator applied to CG_{n-1} to produce causal gap CG_n (i.e. op_n) is $\text{INTRO}_{pi'_m C}$
<assumption>
- 9) $8 \rightarrow \{\text{EFF}(pi_m)\} \cup PE \vdash_{IL} C$
<using 8 (since $\text{INTRO}_{pi'_m C}$ is applicable)>
- 10) $8 \rightarrow CG_n$ equals $\text{NEW-}CG(CG_{n-1}, pi_m, C, pi_1, \dots, pi_{m-1})$
<using 8 and the fact that the causal gap produced by applying $\text{INTRO}_{pi'_m C}$ to causal gap CG (where pi_1, \dots, pi_{m-1} have already been entered) is $\text{NEW-}CG(CG, pi_m, C, pi_1, \dots, pi_{m-1})$ >
- 11) $1,8 \rightarrow S \models \text{GEN-ST}(I_p, pi, \text{NEW-}CG(CG_{n-1}, pi_m, C, pi_1, \dots, pi_{m-1}), pi_1, \dots, pi_{m-1}, pi_m)$
<replacing CG_n in 1 with its equivalent form given in 8>
- 12) $1,2 \rightarrow S \models \text{GEN-ST}(I_p, (\text{COMP } pi_m pi), CG_{n-1}, pi_1, \dots, pi_{m-1})$
<REDUCTION-INTRO applied to 9,11 using substitutions $/CG_{n-1}/CG, pi_m/pix, m-1/i$ >
- 13) $1,8 \rightarrow S \models (\text{AND} (\text{INEV } I_p (\text{EXECUTABLE } \text{CMP}(pi_1, \dots, pi_m, pi)))$
 $(\text{INEV } I_p (\text{IF} (\text{OCC} (\text{CMP}(pi_1, \dots, pi_m, pi) G))))$
<inductive hypothesis applied to 12 and then $\text{CMP}(pi_1, \dots, pi_{m-1}, (\text{COMP } pi_m pi))$ replaced (twice) by its equivalent form $\text{CMP}(pi_1, \dots, pi_{m-1}, pi_m, pi)$ >
- 14) $1 \rightarrow S \models (\text{AND} (\text{INEV } I_p (\text{EXECUTABLE } \text{CMP}(pi_1, \dots, pi_m, pi)))$
 $(\text{INEV } I_p (\text{IF} (\text{OCC} (\text{CMP}(pi_1, \dots, pi_m, pi) G))))$
<using 7, 14 and the fact that either 2 or 7 must hold (since the operator applied to CG_{n-1} must either be a remove or the introduction of a plan instance, which is necessarily pi_m since it is the last plan instance entered)>

QED (inductive step)

QED (REDUCTIONS)

REDUCTION-REMOVE)

If $PE \vdash_{II} C$
 and $S \models GEN-ST(Ip, pi, (CG - C), pi_1, \dots, pi_i)$
 then $S \models GEN-ST(Ip, pi, CG, pi_1, \dots, pi_i)$
 where $\{(INEV \ i \ p) \mid p \in PE\} \subseteq S$

Lemma1)

$(IF (IF P Q) (IF (AND A P B) (AND A Q B)))$

Proof of Lemma1

1) $\vdash (IF P Q)$
<assumption>

2) P
<assumption>

3) A
<assumption>

4) B
<assumption>

5) $1, 2 \vdash Q$
<MP applied to 2, 1>

6) $1-4 \vdash (AND A Q B)$
<AND-INTRO applied to 3, 5, 4>

7) $1 \vdash (IF (AND A P B) (AND A Q B))$
<discharging assumptions 2, 3, and 4 in 6>

8) $\vdash (IF (IF P Q) (IF (AND A P B) (AND A Q B)))$
<discharging assumption in 7>

QED (Lemma1)

Proof continued on next page

Proof of REDUCTION-REMOVE

1) $PE \vdash_{IL} C$

<assumption for proof of meta-theorem>

2) $S \models GEN-ST(Ip, pi, (CG - C), pi_1, \dots, pi_i)$

<assumption for proof of meta-theorem>

3) $1 \rightarrow PE \vdash C$ <using 1 and fact that if ' $A \vdash_{IL} B$ ' is true then ' $A \vdash B$ ' is true (if B is derivable from A in the interval logic fragment then B is derivable from A using all axioms and inference rules)>4) $1 \rightarrow \{(INEV i p) \mid p \in PE\} \vdash (IF (INEV Ip (IF (OCC pi) (CG - C)))$
(INEV Ip (IF (OCC pi) CG)))<Lemma-ST1-ST2 applied to 3 using substitutions $[Ip/i, (OCC pi)/A_i]$ >5) $1 \rightarrow S \vdash (IF (INEV Ip (IF (OCC pi) (CG - C)))$
(INEV Ip (IF (OCC pi) CG)))<ANTCD-INTRO applied to 4 (applicable because $\{(INEV i p) \mid p \in PE\}$ is a subset of S)>6) $1 \rightarrow S \vdash (IF (AND (PRIOR Ip (TIME-OF pi))$
(INEV Ip (EXECUTABLE pi))
(INEV Ip (IF (OCC pi) (CG - C)))
(INEV Ip EXC-NI(pi, pi₁))(INEV Ip EXC-NI(pi, pi_i))
(AND (PRIOR Ip (TIME-OF pi))
(INEV Ip (EXECUTABLE pi))
(INEV Ip (IF (OCC pi) CG))
(INEV Ip EXC-NI(pi, pi₁))(INEV Ip EXC-NI(pi, pi_i))<MP applied to 5, (Lemma1 with substitutions $[(INEV Ip (IF (OCC pi) (CG - C))]/P, (INEV Ip (IF (OCC pi) CG))/Q, (AND (PRIOR Ip (TIME-OF pi)) (INEV Ip (EXECUTABLE pi)))/A, (AND (INEV Ip EXC-NI(pi, pi_1)) \dots (INEV Ip EXC-NI(pi, pi_i)))/B_i]$ >7) $1 \rightarrow S \vdash (IF GEN-ST(Ip, pi, (CG - C), pi_1, \dots, pi_i)$
GEN-ST(Ip, pi, CG, pi₁, \dots, pi_i))

<substituting GEN-ST for its definition in 6>

8) $1, 2 \rightarrow S \models GEN-ST(Ip, pi, CG, pi_1, \dots, pi_i)$

<REDUCTION-AUX-1 applied to 7, 2>

QED (REDUCTION-REMOVE)

Lemma-ST1-ST2)

From: $PE \vdash C$

To: $\{(INEV\ i\ p) \mid p \in PE\}$
 $\vdash (IF\ (INEV\ i\ (IF\ A\ (CG - C)))$
 $(INEV\ i\ (IF\ A\ CG)))$

Proof of Lemma-ST1-ST2

- 1) $PE \vdash C$
<assumption for proof of rule>
- 2) A
<assumption>
- 3) $(IF\ A\ (CG - C))$
<assumption>
- 4) $2,3 \vdash (CG - C)$
<MP applied to 2,3>
- 5) $1 \rightarrow PE \cup 2,3 \vdash (AND\ (CG - C)\ C)$
<AND-INTRO applied to 4,1>
- 6) $\vdash (IF\ (AND\ (CG - C)\ C))$
<Property of (CG - C)>
- 7) $1 \rightarrow PE \cup 2,3 \vdash CG$
<MP applied to 5,6>
- 8) $1 \rightarrow PE \cup 3 \vdash (IF\ A\ CG)$
<discharging assumption 2 in 5>
- 9) $1 \rightarrow PE \vdash (IF\ (IF\ A\ (CG - C))\ (IF\ A\ CG))$
<discharging assumption 3 in 8>
- 10) $1 \rightarrow \{(INEV\ i\ p) \mid p \in PE\}$
 $\vdash (INEV\ i\ (IF\ (IF\ A\ (CG - C))\ (IF\ A\ CG)))$
<assumptions put on left side after DRL-IFTR3 applied $|PE|-1$ times using equivalence TH-INV-SF9 to (DRL-IFTR1 applied to 9 after assumptions (members of the set PE) discharged)>
- 11) $1 \rightarrow \{(INEV\ i\ p) \mid p \in PE\}$
 $\vdash (IF\ (INEV\ i\ (IF\ A\ (CG - C)))$
 $(INEV\ i\ (IF\ A\ CG)))$
<DRL-IFTR1 applied to 10>

QED (Lemma-ST1-ST2)

Lemma-ST0-INT1)

$\vdash (\text{IF } (\exists ?y P(ty)) (\exists ?x P(?x)))$

where $P(t)$ is a function from term t to a sentence, $?y$ is the only variable appearing in term ty , ty has the same type as $?x$, there are no free occurrences of $?y$ in $P(?x)$, and there are no free occurrences of $?x$ in $P(ty)$

Proof of Lemma-ST0-INT1

1) $\vdash (\text{IF } (\forall ?x (\text{NOT } P(?x))) (\text{NOT } P(ty)))$

<AX-FO5 which is applicable because ty has the same type as $?x$, and $?y$, which is the only variable appearing in ty , is not bound in $P(ty)$ >

2) $\vdash (\text{IF } (\forall ?x (\text{NOT } P(?x))) (\forall ?y (\text{NOT } P(ty))))$

<UNIV-INTRO applied to 1 which is applicable since there are no free occurrences of $?y$ in $P(?x)$ >

3) $\vdash (\text{IF } (\text{NOT } (\forall ?y (\text{NOT } P(ty)))) (\text{NOT } (\forall ?x (\text{NOT } P(?x)))))$

<MP applied to 2, TRANSP>

4) $\vdash (\text{IF } (\exists ?y P(ty)) (\exists ?x P(?x)))$

<substituting \exists for its definition in S >

QED (Lemma-ST0-INT1)

Lemma-INT1-INT2)

From: $PE \cup \{EFF(pi1)\} \vdash_{IL} C1$ To: $S-EF \vdash (IF (INEV i (IF (OCC pi2) (CG - C1)))$
 $(INEV i (IF (OCC (COMP pi2 pi1)) CG)))$ where $S-EF =_{def} \{ \neg (INEV i p) \mid p \in PE \}$
 $\cup \{ \neg (INEV i (IF (OCC pi1) EFF(pi1))) \}$

Proof of Lemma-INT1-INT2

1) $PE \cup \{EFF(pi1)\} \vdash_{IL} C1$ *<assumption for proof of rule>*2) $1 \rightarrow PE \vdash (IF EFF(pi1) C1)$ *<discharging assumption $\{EFF(pi1)\}$ in 1 and using fact that if ' $A \vdash_{IL} B$ ' is true true then ' $A \vdash B$ ' is true (if B is derivable from A in the interval logic fragment then B is derivable from A using all axioms and inference rules)>*3) $(IF (OCC pi1) EFF(pi1))$ *<assumption>*4) $1 \rightarrow PE \cup 3 \vdash (IF (OCC pi1) C1)$ *<MP-TRNS applied to 3,2>*5) $(IF (OCC pi2) (CG - C1))$ *<assumption>*6) $1 \rightarrow PE \cup 3,5 \vdash (IF (AND (OCC pi1) (OCC pi2))$
 $(AND C1 (CG - C1)))$ *<AND-MI applied to 4,5>*7) $1 \rightarrow PE \cup 3,5 \vdash (IF (OCC (COMP pi1 pi2))$
 $(AND C1 (CG - C1)))$ *<SUBST applied to 6 using equivalence AX-IL5>*8) $\vdash (IF (AND C1 (CG - C1)) CG)$ *<property of (P - Q)>*9) $1 \rightarrow PE \cup 3,5 \vdash (IF (OCC (COMP pi1 pi2)) CG)$ *<MP-TRNS applied to 7,8>*10) $1 \rightarrow PE \cup 3 \vdash (IF (IF (OCC pi2) (CG - C1))$
 $(IF (OCC (COMP pi1 pi2)) CG))$ *<discharging assumption 5 in 9>*11) $1 \rightarrow S-EF \vdash (INEV i (IF (IF (OCC pi2) (CG - C1))$
 $(IF (OCC (COMP pi1 pi2)) CG)))$ *<DRL-IFTR3 applied $|PE|$ times using equivalence TH-INV-SF9 to (DRL-IFTR1 applied to 10 after assumptions discharged)>*12) $1 \rightarrow S-EF \vdash (IF (INEV i (IF (OCC pi2) (CG - C1)))$
 $(INEV i (IF (OCC (COMP pi1 pi2)) CG)))$ *<MP applied to 11, AX-INV2>*

QED (Lemma-INT1-INT2)

Lemma-INT2-INT3)

\vdash (IF (PRIOR i (TIME-OF (COMP pi2 pi1)))
 (IF (AND (INEV i (EXECUTABLE pi2))
 (INEV i (IF (OCC pi2) (EXECUTABLE pi1)))
 (INEV i (IF (OCC pi2) (IFTRIED pi1 (OCC pi2))))))
 (INEV i (EXECUTABLE (COMP pi2 pi1))))

Lemma1)

\vdash (IF (PRIOR i (TIME-OF pi))
 (IF (AND (INEV i (EXECUTABLE pi)) (INEV i (IF (OCC pi) P)))
 (INEV i (IFTRIED pi P))))

Proof of Lemma1

- 1) (PRIOR i (TIME-OF pi))
<assumption>
- 2) (INEV i (EXECUTABLE pi))
<assumption>
- 3) (INEV i (IF (OCC pi) P))
<assumption>
- 4) \vdash (IF (PRIOR i (TIME-OF pi))
 (IF (INEV i Q) (IFTRIED pi Q)))
<AX-IFTR6>
- 5) \vdash (IF (INEV i (PRIOR i (TIME-OF pi)))
 (INEV i (IF (INEV i Q) (IFTRIED pi Q))))
<DRL-IFTR1 applied to 4>
- 6) \vdash (IF (PRIOR i (TIME-OF pi))
 (INEV i (IF (INEV i Q) (IFTRIED pi Q))))
<DRL-IFTR3 applied to 5 using equivalence TH-INV-IL4>
- 7) 1 \vdash (INEV i (IF (INEV i Q) (IFTRIED pi Q)))
<MP applied to 1,6>
- 8) 1 \vdash (IF (INEV i (INEV i Q)) (INEV i (IFTRIED pi Q)))
<MP applied to 7,AX-INV2>
- 9) 1 \vdash (IF (INEV i Q) (INEV i (IFTRIED pi Q)))
<DRL-IFTR3 applied to 8 using equivalence TH-INV-SF5>
- 10) 1,3 \vdash (INEV i (IFTRIED pi (IF (OCC pi) P)))
<MP applied to 9,(9 with substitution [(IF (OCC pi) P)/Q]>
- 11) 2 \vdash (INEV i (IFTRIED pi (OCC pi)))
<replacing EXECUTABLE by its definition in 2>
- 12) 1-3 \vdash (INEV i (IFTRIED pi P))
<DRL-IFTR4 applied to 11,10>
- 13) \vdash (IF (PRIOR i (TIME-OF pi))
 (IF (AND (INEV i (EXECUTABLE pi))
 (INEV i (IF (OCC pi) P)))
 (INEV i (IFTRIED pi P))))
<discharging assumptions 2,3 then 1 in 12>

QED (Lemma1)

Proof of Lemma-INT2-INT3

- 1) (PRIOR i (TIME-OF (COMP pi2 pi1)))
 <assumption>
- 2) (INEV i (EXECUTABLE pi2))
 <assumption>
- 3) (INEV i (IF (OCC pi2) (EXECUTABLE pi1)))
 <assumption>
- 4) (INEV i (IF (OCC pi2) (IFTRIED pi1 (OCC pi2))))
 <assumption>

- 5) $1 \vdash$ (AND (PRIOR i (TIME-OF pi1))
 (PRIOR i (TIME-OF pi2)))
 <MP applied to 1, (ONLY-IF part of TH-APG-1)>
- 6) $1 \vdash$ (PRIOR i (TIME-OF pi2))
 <AND-ELIM applied to 5>

- 7) $1-3 \vdash$ (INEV i (IFTRIED pi2 (EXECUTABLE pi1)))
 <MP applied to (AND-INTRO applied to 2,3), (MP applied to 6, Lemma1 with substitutions
 [pi2/pi, (EXECUTABLE pi1)/P_i]>
- 8) $1-3 \vdash$ (INEV i (IFTRIED pi2 (IFTRIED pi1 (OCC pi1))))
 <EXECUTABLE replaced by its definition in 7>

- 9) $1,2,4 \vdash$ (INEV i (IFTRIED pi2 (IFTRIED pi1 (OCC pi2))))
 <MP applied to (AND-INTRO applied to 2,4), (MP applied to 6, Lemma1 with substitutions
 [pi2/pi, (IFTRIED pi1 (OCC pi2))/P_i]>
- 10) $1-4 \vdash$ (INEV i (IFTRIED pi2
 (IFTRIED pi1 (AND (OCC pi2)
 (OCC pi1)))))
 <DRL-IFTR5 applied to 9,8>
- 11) \vdash (IF (INEV i (IFTRIED pi2 (IFTRIED pi1 (AND (OCC pi2) (OCC pi1)))))
 (INEV i (EXECUTABLE (COMP pi2 pi1))))
 <DRL-INV1 applied to TH-APG-9 with substitutions [pi2/pi1, pi1/pi2]>
- 12) $1-4 \vdash$ (INEV i (EXECUTABLE (COMP pi2 pi1)))
 <MP applied to 10,11>

- \vdash (IF (PRIOR i (TIME-OF (COMP pi2 pi1)))
 (IF (AND (INEV i (EXECUTABLE pi2))
 (INEV i (IF (OCC pi2) (EXECUTABLE pi1)))
 (INEV i (IF (OCC pi2) (IFTRIED pi1 (OCC pi2)))))
 (INEV i (EXECUTABLE (COMP pi2 pi1))))
 <Discharging assumption 1 after discharging 2-4 in 12>

QED (Lemma-INT2-INT3)

Lemma-INT3-INT4

$\{ \sim(\text{INEV } i (\text{IF } \text{EC}(\text{pi1}) (\text{EXECUTABLE } \text{pi1})))^{\sim} \} \vdash (\text{IF } (\text{INEV } i (\text{IF } (\text{OCC } \text{pi2}) \text{EC}(\text{pi1})))$
 $(\text{INEV } i (\text{IF } (\text{OCC } \text{pi2}) (\text{EXECUTABLE } \text{pi1}))))$

Proof of Lemma-INT3-INT4

- 1) $(\text{IF } \text{EC}(\text{pi1}) (\text{EXECUTABLE } \text{pi1}))$
<assumption>
- 2) $(\text{IF } (\text{OCC } \text{pi2}) \text{EC}(\text{pi1}))$
<assumption>
- 3) 1,2 $\vdash (\text{IF } (\text{OCC } \text{pi2}) (\text{EXECUTABLE } \text{pi1}))$
<MP-TRNS applied to 2,1>
- 4) $\vdash (\text{IF } (\text{IF } \text{EC}(\text{pi1}) (\text{EXECUTABLE } \text{pi1}))$
 $(\text{IF } (\text{IF } (\text{OCC } \text{pi2}) \text{EC}(\text{pi1}))$
 $(\text{IF } (\text{OCC } \text{pi2}) (\text{EXECUTABLE } \text{pi1}))))$
<discharging assumption 2 then 1 in 3>
- 5) $\vdash (\text{IF } (\text{INEV } i (\text{IF } \text{EC}(\text{pi1}) (\text{EXECUTABLE } \text{pi1})))$
 $(\text{INEV } i (\text{IF } (\text{IF } (\text{OCC } \text{pi2}) \text{EC}(\text{pi1}))$
 $(\text{IF } (\text{OCC } \text{pi2}) (\text{EXECUTABLE } \text{pi1}))))$
<DRL-IFTR1 applied to 4>
- 6) $\{ \sim(\text{INEV } i (\text{IF } \text{EC}(\text{pi1}) (\text{EXECUTABLE } \text{pi1})))^{\sim} \}$
 $\vdash (\text{INEV } i (\text{IF } (\text{IF } (\text{OCC } \text{pi2}) \text{EC}(\text{pi1}))$
 $(\text{IF } (\text{OCC } \text{pi2}) (\text{EXECUTABLE } \text{pi1}))))$
<writing 5 in an equivalent form>
- 7) $\{ \sim(\text{INEV } i (\text{IF } \text{EC}(\text{pi1}) (\text{EXECUTABLE } \text{pi1})))^{\sim} \}$
 $\vdash (\text{IF } (\text{INEV } i (\text{IF } (\text{OCC } \text{pi2}) \text{EC}(\text{pi1})))$
 $(\text{INEV } i (\text{IF } (\text{OCC } \text{pi2}) (\text{EXECUTABLE } \text{pi1}))))$
<MP applied to 6, AX-INV2>

QED (Lemma-INT3-INT4)

Lemma-INT4-INT5)

$$\vdash (\text{IF } (\text{INEV } i \text{ (IF } P \text{ (AND } Q \text{ R}))) \\ (\text{AND } (\text{INEV } i \text{ (IF } P \text{ Q})) (\text{INEV } i \text{ (IF } P \text{ R}))))$$

Proof of Lemma-INT4-INT5

- 1) P
 <assumption>
- 2) $(\text{IF } P \text{ (AND } Q \text{ R)})$
 <assumption>
- 3) 1,2 $\vdash (\text{AND } Q \text{ R})$
 <MP applied to 1,2>
- 4) 1,2 $\vdash Q$
 <AND-ELIM applied to 3>
- 5) 2 $\vdash (\text{IF } P \text{ Q})$
 <discharging assumption 1 in 4>
- 6) 1,2 $\vdash R$
 <AND-ELIM applied to 3>
- 7) 2 $\vdash (\text{IF } P \text{ R})$
 <discharging assumption 1 in 6>
- 8) 2 $\vdash (\text{AND } (\text{IF } P \text{ Q}) (\text{IF } P \text{ R}))$
 <AND-INTRO applied to 5,7>
- 9) $\vdash (\text{IF } (\text{IF } P \text{ (AND } Q \text{ R)}) \\ (\text{AND } (\text{IF } P \text{ Q}) (\text{IF } P \text{ R})))$
 <discharging assumption 2 in 8>
- 10) $\vdash (\text{IF } (\text{INEV } i \text{ (IF } P \text{ (AND } Q \text{ R}))) \\ (\text{INEV } i \text{ (AND } (\text{IF } P \text{ Q}) (\text{IF } P \text{ R}))))$
 <DRL-IFTR1 applied to 9>
- 11) $\vdash (\text{IF } (\text{INEV } i \text{ (IF } P \text{ (AND } Q \text{ R}))) \\ (\text{AND } (\text{INEV } i \text{ (IF } P \text{ Q})) (\text{INEV } i \text{ (IF } P \text{ R}))))$
 <DRL-IFTR3 applied to 10 with equivalence TH-INV-SF9>

QED (Lemma-INT4-INT5)

Lemma-aux-1)

$$\vdash (\text{IF } (\text{AND } (\text{INEV } i \text{ (IF } Q \text{ (IF } P \text{ R}))) (\text{INEV } i \text{ (IF } P \text{ Q}))) \\ (\text{INEV } i \text{ (IF } P \text{ R})))$$

Proof of Lemma-aux-1

1) (IF Q (IF P R))
<assumption>

2) (IF P Q)
<assumption>

3) P
<assumption>

4) 2.3 \vdash Q
<MP applied to 3.2>

5) 1-3 \vdash (IF P R)
<MP applied to 4.1>

6) 1-3 \vdash R
<MP applied to 3.5>

7) 1.2 \vdash (IF P R)
<discharging assumption 3 in 6>

8) \vdash (IF (AND (IF Q (IF P R)) (IF P Q))
 (IF P R))
<discharging>

9) \vdash (IF (INEV i (AND (IF Q (IF P R)) (IF P Q)))
 (INEV i (IF P R)))
<DRL-IFTR1 applied to 8>

10) \vdash (IF (AND (INEV i (IF Q (IF P R))) (INEV i (IF P Q)))
 (INEV i (IF P R)))
<DRL-IFTR8 applied to 9 with equivalence TH-INV-SF9>

QED (Lemma-aux-1)

Lemma-aux-2)

$\vdash (\text{IF } (\text{IF } P \ Q) \ (\text{IF } (\exists ?x \ P) \ (\exists ?x \ Q)))$

Proof of Lemma-aux-2

1) $(\text{IF } P \ Q)$

<assumption>

2) $1 \vdash (\text{IF } (\text{NOT } Q) \ (\text{NOT } P))$

<MP applied to 1, TRANSP>

3) $\vdash (\text{IF } (\forall ?x \ (\text{NOT } Q)) \ (\text{NOT } Q))$

<AX-FO5 with substitutions $[(\text{NOT } Q)/P1, (\text{NOT } Q)/P2]$ >

4) $1 \vdash (\text{IF } (\forall ?x \ (\text{NOT } Q)) \ (\text{NOT } P))$

<MP-TRANS applied to 3,2>

5) $1 \vdash (\text{IF } (\forall ?x \ (\text{NOT } Q)) \ (\forall ?x \ (\text{NOT } P)))$

<UNV-INTRO applied to 4 with substitutions $[(\forall ?x \ (\text{NOT } Q))/P, (\text{NOT } P)]$; this rule is applicable because $?x$ does not occur free in $(\forall ?x \ (\text{NOT } Q))$ >

6) $1 \vdash (\text{IF } (\text{NOT } (\forall ?x \ (\text{NOT } P))) \ (\text{NOT } (\forall ?x \ (\text{NOT } Q))))$

<MP applied to 5, TRANSP>

7) $1 \vdash (\text{IF } (\exists ?x \ P) \ (\exists ?x \ Q))$

< \exists substituted for its definition in 6>

8) $\vdash (\text{IF } (\text{IF } P \ Q) \ (\text{IF } (\exists ?x \ P) \ (\exists ?x \ Q)))$

<discharging assumption in 7>

QED (Lemma-aux-2)

Lemma-INTS2-INTS5-OVRLP)

S-PR_US-NI

\vdash (IF (AND (INEV Ip (IF (OCC pi3) (EXECUTABLE pi2)))
 (INEV Ip (IF (OCC pi3) NI(pi1,pi2)))
 (INEV Ip EXC-NI(pi3,pi1)))
 (INEV Ip EXC-NI(\neg (COMP pi3 pi2) \neg ,pi1)))

where S-PR =_{def} { \neg (PRIOR Ip (TIME-OF pi1)) \neg }

S-NI =_{def} { \neg (INEV Ip (IF NI(pi1,pi2) NI-CND(pi1,pi2))) \neg }

Proof of Lemma-INTS2-INTS5-OVRLP

- 1) (PRIOR Ip (TIME-OF pi1))
 <assumption>
- 2) (INEV Ip (IF NI(pi1,pi2) NI-CND(pi1,pi2)))
 <assumption>
- 3) (INEV Ip (IF (OCC pi3) (EXECUTABLE pi2)))
 <assumption>
- 4) (INEV Ip (IF (OCC pi3) NI(pi1,pi2)))
 <assumption>
- 5) (INEV Ip EXC-NI(pi3,pi1))
 <assumption>
- 6) (IF (OCC pi3) NI(pi1,pi2))
 <assumption>
- 7) (IF NI(pi1,pi2) NI-CND(pi1,pi2))
 <assumption>
- 8) 6,7 \vdash (IF (OCC pi3) NI-CND(pi1,pi2))
 <MP-TRNS applied to 6,7>
 <discharging assumptions in 8>
- 9) \vdash (IF (AND (INEV Ip (IF (OCC pi3) NI(pi1,pi2)))
 (INEV Ip (IF NI(pi1,pi2) NI-CND(pi1,pi2)))
 (INEV Ip (IF (OCC pi3) NI-CND(pi1,pi2))))
 <DRL-IFTR3 applied using equivalence TH-INV-SF9 to (DRL-IFTR1 applied to 8 after assumptions discharged)>
- 10) 2,4 \vdash (INEV Ip (IF (OCC pi3) NI-CND(pi1,pi2)))
 <MP applied to (AND-INTRO applied to 4,2),9>
- 11) 1-5 \vdash (INEV Ip EXC-NI(\neg (COMP pi3 pi2) \neg ,pi1))
 <MP applied to (AND-INTRO applied to 1,9,5),(MP applied to 10,Lemma-INTS2-INTS5-main)>
- 12) S-PR_US-NI \vdash (IF (AND (INEV Ip (IF (OCC pi3) (EXECUTABLE pi2)))
 (INEV Ip (IF (OCC pi3) NI(pi1,pi2)))
 (INEV Ip EXC-NI(pi3,pi1)))
 (INEV Ip EXC-NI(\neg (COMP pi3 pi2) \neg ,pi1)))
 <S-PR substituted for 1 and S-NI substituted for 2 after assumptions 9-5 discharged in 11>

QED (Lemma-INTS2-INTS5-OVRLP)

Lemma-INTS2-INTS5-NON-OVRLP)

S-PR \vdash (IF PI-DISJOINT(pi1,pi2)
 (IF (AND (INEV Ip (IF (OCC pi3) (EXECUTABLE pi2)))
 (INEV Ip EXC-NI(pi3,pi1)))
 (INEV Ip EXC-NI(^(COMP pi3 pi2)^,pi1)))
 where S-PR =_{def} {^(PRIOR Ip (TIME-OF pi1))^}

Proof of Lemma-INTS2-INTS5-NON-OVRLP

- 1) (PRIOR Ip (TIME-OF pi1))
 <assumption>
- 2) (OR (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 (PRIOR (TIME-OF pi2) (TIME-OF pi1))
 <assumption>
- 3) (INEV Ip (IF (OCC pi3) (EXECUTABLE pi2)))
 <assumption>
- 4) (INEV Ip EXC-NI(pi3,pi1))
 <assumption>
- 5) \vdash (IF (PRIOR (TIME-OF pi1) (TIME-OF pi2))
 (INEV Ip NI-CND(pi1,pi2)))
 <NON-OVRLP with substitution /Ip/i>
- 6) \vdash (IF (PRIOR (TIME-OF pi2) (TIME-OF pi1))
 (INEV Ip NI-CND(pi2,pi1)))
 <NON-OVRLP with substitutions /Ip/i, pi2/pi1, pi1/pi2>
- 7) \vdash (IF (INEV Ip NI-CND(pi2,pi1))
 (INEV Ip NI-CND(pi1,pi2))
 <DRL-IFTR1 applied to Lemmas>
- 8) \vdash (IF (PRIOR (TIME-OF pi2) (TIME-OF pi1))
 (INEV Ip NI-CND(pi1,pi2)))
 <MP-TRNS applied to 6, 7>
- 9) 2 \vdash (INEV Ip NI-CND(pi1,pi2))
 <MP applied to (RCS applied to 2,5,8), AX-FOI>
- 10) (OCC pi3)
 <assumption>
- 11) NI-CND(pi1,pi2)
 <assumption>
- 12) 10,11 \vdash NI-CND(pi1,pi2)
 <ANTCD-INTRO applied to 11 using antecedant 10>
- 13) \vdash (IF NI-CND(pi1,pi2) (IF (OCC pi3) NI-CND(pi1,pi2)))
 <discharging assumption 10 then 11 in 12>
- 14) \vdash (IF (INEV Ip NI-CND(pi1,pi2))
 (INEV Ip (IF (OCC pi3) NI-CND(pi1,pi2))))
 <DRL-IFTR1 applied to 13>

Proof continued on next page

Continuation of Lemma-INTS2-INTS5-NON-OVRLP

15) $2 \vdash (\text{INEV } I_p (\text{IF } (\text{OCC } \pi_3) \text{NI-CND}(\pi_1, \pi_2)))$
<MP applied to 9,14>

16) $1-4 \vdash (\text{INEV } I_p \text{EXC-NI}(\neg(\text{COMP } \pi_3 \pi_2)^\sim, \pi_1))$
<MP applied to (AND-INTRO applied to 1,9,4).(MP applied to 15.Lemma-INTS2-INTS5-main)>

17) $S\text{-PR} \vdash (\text{IF } \text{PI-DISJOINT}(\pi_1, \pi_2)$
 $(\text{PRIOR } (\text{TIME-OF } \pi_2) (\text{TIME-OF } \pi_1)))$
 $(\text{IF } (\text{AND } (\text{INEV } I_p (\text{IF } (\text{OCC } \pi_3)$
 $(\text{EXECUTABLE } \pi_2)))$
 $(\text{INEV } I_p \text{EXC-NI}(\pi_3, \pi_1)))$
 $(\text{INEV } I_p \text{EXC-NI}(\neg(\text{COMP } \pi_3 \pi_2)^\sim, \pi_1)))$
<S-PR substituted for 1 and PI-DISJOINT substituted for its definition after assumptions 9 and 4 then 2 discharged in 16>

QED (Lemma-INTS2-INTS5-NON-OVRLP)

Lemmax)

$\vdash (\text{IF } \text{NI-CND}(\pi_2, \pi_1) \text{NI-CND}(\pi_1, \pi_2))$

Proof of Lemmax

1) $\text{NI-CND}(\pi_2, \pi_1)$
<assumption>

2) $1 \vdash (\text{IF } (\text{AND } (\text{EXECUTABLE } \pi_2) (\text{EXECUTABLE } \pi_1))$
 $(\text{AND } (\text{IF } (\text{IFTRIED } \pi_1 (\text{EXECUTABLE } \pi_2))$
 $(\text{IF } (\text{OCC } \pi_2) (\text{IFTRIED } \pi_1 (\text{OCC } \pi_2))))$
 $(\text{IF } (\text{IFTRIED } \pi_2 (\text{EXECUTABLE } \pi_1))$
 $(\text{IF } (\text{OCC } \pi_1) (\text{IFTRIED } \pi_2 (\text{OCC } \pi_1))))))$
<replacing NI-CND by its definition in 1>

3) $1 \vdash (\text{IF } (\text{AND } (\text{EXECUTABLE } \pi_1) (\text{EXECUTABLE } \pi_2))$
 $(\text{AND } (\text{IF } (\text{IFTRIED } \pi_2 (\text{EXECUTABLE } \pi_1))$
 $(\text{IF } (\text{OCC } \pi_1) (\text{IFTRIED } \pi_2 (\text{OCC } \pi_1))))$
 $(\text{IF } (\text{IFTRIED } \pi_1 (\text{EXECUTABLE } \pi_2))$
 $(\text{IF } (\text{OCC } \pi_2) (\text{IFTRIED } \pi_1 (\text{OCC } \pi_2))))))$
<SUBST applied twice to 2 using equivalence (IFF (AND P Q) (AND Q P))>

4) $1 \vdash \text{NI-CND}(\pi_1, \pi_2)$
<NI-CND substituted for its definition in 3>

5) $\vdash (\text{IF } \text{NI-CND}(\pi_2, \pi_1) \text{NI-CND}(\pi_1, \pi_2))$
<discharging assumption in 4>

QED (Lemmax)

Lemma-INTS2-INTS5-main)

⊢ (IF (INEV Ip (IF (OCC pi3) NI-CND(pi1,pi2)))
 (IF (AND (PRIOR Ip (TIME-OF pi1))
 (INEV Ip (IF (OCC pi3) (EXECUTABLE pi2)))
 (INEV Ip EXC-NI(pi3,pi1)))
 (INEV Ip EXC-NI(¬(COMP pi3 pi2)¬,pi1))))

Proof of Lemma-INTS2-INTS5-main

1) (INEV Ip (IF (OCC pi3) NI-CND(pi1,pi2)))
 <assumption>

2) (PRIOR Ip (TIME-OF pi1))
 <assumption>

3) (INEV Ip (IF (OCC pi3) (EXECUTABLE pi2)))
 <assumption>

4) (INEV Ip EXC-NI(pi3,pi1))
 <assumption>

5) 2 ⊢ (IF (AND (INEV Ip (IF (OCC pi3) (EXECUTABLE pi2)))
 (INEV Ip EXC-NI(pi3,pi1))
 (INEV Ip (IF (EXECUTABLE pi1)
 (IF (OCC pi3)
 (IFTRIED pi1
 (EXECUTABLE pi2)))))))
 <MP applied to 2,(Lemma-1 with substitutions [Ip/i, pi1/pi, (EXECUTABLE pi1)/A, (OCC pi3)/P, (EXECUTABLE pi2)/Q,]>

6) 2-4 ⊢ (INEV Ip (IF (EXECUTABLE pi1)
 (IF (OCC pi3)
 (IFTRIED pi1 (EXECUTABLE pi2))))
 <MP applied to (AND-INTRO applied to 3,4),5>

7) 1-4 ⊢ (INEV Ip (IF (EXECUTABLE pi1)
 (IF (OCC pi3)
 (IF (OCC pi2)
 (IFTRIED pi1 (OCC pi2))))))
 <MP applied to (AND-INTRO applied to 1,6),(Lemma-2 with substitutions [Ip/i, (OCC pi3)/P,]>

8) 1-4 ⊢ (INEV Ip (IF (EXECUTABLE pi1)
 (IF (OCC (COMP pi3 pi2))
 (IFTRIED pi1 (OCC pi2))))
 <MP applied to 7,(Lemma-3 with substitution [Ip/i, (EXECUTABLE pi1)/A,(IFTRIED pi1 (OCC pi2))/P]>

9) (IF (EXECUTABLE pi1)
 (IF (OCC pi3) (IFTRIED pi1 (OCC pi3))))
 <assumption>

10) (EXECUTABLE pi1)
 <assumption>

Proof continued on next page

- 11) $9,10 \vdash (\text{IF } (\text{OCC } \text{pi3}) (\text{IFTRIED } \text{pi1 } (\text{OCC } \text{pi3})))$
<MP applied to 10,9>
- 12) $9,10 \vdash (\text{IF } (\text{OCC } \text{pi2})$
 $(\text{IF } (\text{OCC } \text{pi3}) (\text{IFTRIED } \text{pi1 } (\text{OCC } \text{pi3}))))$
<ANTCD-INTRO applied to 11>
- 13) $9 \vdash (\text{IF } (\text{EXECUTABLE } \text{pi1})$
 $(\text{IF } (\text{OCC } \text{pi2})$
 $(\text{IF } (\text{OCC } \text{pi3}) (\text{IFTRIED } \text{pi1 } (\text{OCC } \text{pi3}))))))$
<discharging assumption 10 in 12>
- 14) $\vdash (\text{IF } (\text{INEV } \text{Ip } (\text{IF } (\text{EXECUTABLE } \text{pi1})$
 $(\text{IF } (\text{OCC } \text{pi3})$
 $(\text{IFTRIED } \text{pi1 } (\text{OCC } \text{pi3}))))))$
 $(\text{INEV } \text{Ip } (\text{IF } (\text{EXECUTABLE } \text{pi1})$
 $(\text{IF } (\text{OCC } \text{pi2})$
 $(\text{IF } (\text{OCC } \text{pi3})$
 $(\text{IFTRIED } \text{pi1 } (\text{OCC } \text{pi3})))))))))$
<DRL-IFTR1 applied to 13 after discharging the assumption>
- 15) $4 \vdash (\text{INEV } \text{Ip } (\text{IF } (\text{EXECUTABLE } \text{pi1})$
 $(\text{IF } (\text{OCC } \text{pi2})$
 $(\text{IF } (\text{OCC } \text{pi3})$
 $(\text{IFTRIED } \text{pi1 } (\text{OCC } \text{pi3}))))))$
<MP applied to 4,14 after EXC-NI is replaced by its definition in 4>
- 16) $4 \vdash (\text{INEV } \text{Ip } (\text{IF } (\text{EXECUTABLE } \text{pi1})$
 $(\text{IF } (\text{OCC } (\text{COMP } \text{pi2 } \text{pi3}))$
 $(\text{IFTRIED } \text{pi1 } (\text{OCC } \text{pi3}))))))$
<MP applied to 15, (Lemma-3 with substitutions $[Ip/i, (\text{EXECUTABLE } \text{pi1})/A, \text{pi2}/\text{pi3}, \text{pi3}/\text{pi2}, (\text{OCC } \text{pi3})/P]$ >
- 17) $4 \vdash (\text{INEV } \text{Ip } (\text{IF } (\text{EXECUTABLE } \text{pi1})$
 $(\text{IF } (\text{OCC } (\text{COMP } \text{pi3 } \text{pi2}))$
 $(\text{IFTRIED } \text{pi1 } (\text{OCC } \text{pi3}))))))$
<MP applied to 16, (MP applied to AX-FO7, AX-IL3)>
- 18) $1-4 \vdash (\text{INEV } \text{Ip } (\text{IF } (\text{EXECUTABLE } \text{pi1})$
 $(\text{IF } (\text{OCC } (\text{COMP } \text{pi3 } \text{pi2}))$
 $(\text{IFTRIED } \text{pi1 } (\text{OCC } (\text{COMP } \text{pi3 } \text{pi2}))))))$
<MP applied to (AND-INTRO applied to 8,17), (Lemma-4 with substitutions $[Ip/i, (\text{EXECUTABLE } \text{pi1})/A, (\text{OCC } (\text{COMP } \text{pi3 } \text{pi2}))/P]$ >
- 19) $1-4 \vdash (\text{INEV } \text{Ip } \text{EXC-NI}(\sim(\text{COMP } \text{pi3 } \text{pi2})^\sim, \text{pi1}))$
<EXC-NI substituted for its definition in 18>
- 20) $\vdash (\text{IF } (\text{INEV } \text{Ip } (\text{IF } (\text{OCC } \text{pi3}) \text{NI-CND}(\text{pi1}, \text{pi2})))$
 $(\text{IF } (\text{AND } (\text{PRIOR } \text{Ip } (\text{TIME-OF } \text{pi1}))$
 $(\text{INEV } \text{Ip } (\text{IF } (\text{OCC } \text{pi3}) (\text{EXECUTABLE } \text{pi2})))$
 $(\text{INEV } \text{Ip } \text{EXC-NI}(\text{pi3}, \text{pi1})))$
 $(\text{INEV } \text{Ip } \text{EXC-NI}(\sim(\text{COMP } \text{pi3 } \text{pi2})^\sim, \text{pi1})))$
<discharging assumptions 2-4 then 1 in 19>
- QED (Lemma-INTS2-INTS5-main)

Lemma-1)

$$\vdash (\text{IF } (\text{PRIOR } i \text{ (TIME-OF } \pi)) \\ (\text{IF } (\text{AND } (\text{INEV } i \text{ (IF } P \ Q)) \\ (\text{INEV } i \text{ (IF } A \ (\text{IF } P \ (\text{IFTRIED } \pi \ P)))) \\ (\text{INEV } i \text{ (IF } A \ (\text{IF } P \ (\text{IFTRIED } \pi \ Q))))))$$

Proof of Lemma-1

- 1) (IFTRIED π (IF $P \ Q$))
<assumption>
- 2) A
<assumption>
- 3) P
<assumption>
- 4) (IF A (IF P (IFTRIED π P)))
<assumption>
- 5) 2-4 \vdash (IFTRIED π P)
<MP applied to 3, (MP applied to 2,4)>
- 6) 1-4 \vdash (IFTRIED π Q)
<DRL-IFTR4 applied to 5,1>
- 7) 1,4 \vdash (IF A (IF P (IFTRIED π Q)))
<discharging assumption 3 then assumption 2 in 6>
- 8) \vdash (IF (AND (IFTRIED π (IF $P \ Q$))
 (IF A (IF P (IFTRIED π P))))
 (IF A (IF P (IFTRIED Q))))
<discharging the assumptions in 7>
- 9) \vdash (IF (AND (INEV i (IFTRIED π (IF $P \ Q$))
 (INEV i (IF A (IF P (IFTRIED π P))))
 (INEV i (IF A (IF P (IFTRIED Q))))))
<DRL-IFTR3 applied to (DRL-IFTR1 applied to 8) using equivalence TH-INV-SF9>
- 10) (PRIOR i (TIME-OF π))
<assumption>
- 11) (INEV i (IF $P \ Q$))
<assumption>
- 12) \vdash (IF (PRIOR i (TIME-OF π))
 (IF (INEV i (IF $P \ Q$)) (IFTRIED π (IF $P \ Q$))))
<AX-IFTR5 with substitution [(IF $P \ Q$)/P]>

Proof continued on next page

Continuation of Lemma-1

- 13) \vdash (IF (INEV i (PRIOR i (TIME-OF pi)))
 (INEV i (IF (INEV i (IF P Q))
 (IFTRIED pi (IF P Q)))))
<DRL-IFTR1 applied to 12>
- 14) \vdash (IF (PRIOR i (TIME-OF pi))
 (INEV i (IF (INEV i (IF P Q))
 (IFTRIED pi (IF P Q)))))
<DRL-IFTR3 applied to 13 using equivalence TH-INV-IL4>
- 15) 10 \vdash (INEV i (IF (INEV i (IF P Q))
 (IFTRIED pi (IF P Q)))))
<MP applied to 10,14>
- 16) 10 \vdash (IF (INEV i (INEV i (IF P Q)))
 (INEV i (IFTRIED pi (IF P Q)))))
<MP applied to 15, AX-IFTR2>
- 17) 10 \vdash (IF (INEV i (IF P Q))
 (INEV i (IFTRIED pi (IF P Q)))))
<DRL-IFTR3 applied to 16 using equivalence TH-INV-SF5>
- 18) 10,11 \vdash (INEV i (IFTRIED pi (IF P Q)))
<MP applied to 11,17>
- 19) (INEV i (IF A (IF P (IFTRIED pi P))))
<assumption>
- 20) 10,11,19 \vdash (AND (INEV i (IFTRIED pi (IF P Q)))
 (INEV i
 (IF A (IF P (IFTRIED pi P)))))
<AND-INTRO applied to 18,19>
- 21) 10,11,19 \vdash (INEV i (IF A (IF P (IFTRIED pi Q)))))
<MP applied to 20,9>
- 22) \vdash (IF (PRIOR i (TIME-OF pi))
 (IF (AND (INEV i (IF P Q))
 (INEV i
 (IF A (IF P (IFTRIED pi P)))))
 (INEV i (IF A (IF P (IFTRIED pi Q)))))
<discharging assumption 11 and 19, then 10 in 21>
- QED (Lemma-1)

Lemma-2)

$$\vdash (\text{IF } (\text{AND } (\text{INEV } i \text{ (IF } P \text{ NI-CND}(\pi_1, \pi_2))) \\ (\text{INEV } i \text{ (IF (EXECUTABLE } \pi_1) \\ (\text{IF } P \text{ (IFTRIED } \pi_1 \\ (\text{EXECUTABLE } \pi_2)))))) \\ (\text{INEV } i \text{ (IF (EXECUTABLE } \pi_1) \\ (\text{IF } P \text{ (IF (OCC } \pi_2) \\ (\text{IFTRIED } \pi_1 \text{ (OCC } \pi_2))))))))))$$

Proof of Lemma-2

- 1) (EXECUTABLE π_1)
 <assumption>
- 2) P
 <assumption>
- 3) (IF P NI-CND(π_1, π_2))
 <assumption>
- 4) (IF (EXECUTABLE π_1)
 (IF P (IFTRIED π_1 (EXECUTABLE π_2))))
 <assumption>
- 5) 2,3 \vdash NI-CND(π_1, π_2)
 <MP applied to 2,3>
- 6) 1,2,4 \vdash (IFTRIED π_1 (EXECUTABLE π_2))
 <MP applied to 2,(MP applied to 1,4)>
- 7) 1-4 \vdash (IF (OCC π_2) (IFTRIED π_1 (OCC π_2)))
 <MP applied to (AND-INTRO applied to 5.1,6), Lemma-2-1>
- 8) 3,4 \vdash (IF (EXECUTABLE π_1)
 (IF P (IF (OCC π_2)
 (IFTRIED π_1 (OCC π_2))))))
 <discharging assumption 2, then assumption 1 in 7>
- 9) \vdash (IF (AND (INEV i (IF P NI-CND(π_1, π_2)))
 (INEV i
 (IF (EXECUTABLE π_1)
 (IF P (IFTRIED π_1 (EXECUTABLE π_2))))))
 (INEV i
 (IF (EXECUTABLE π_1)
 (IF P (IF (OCC π_2) (IFTRIED π_1 (OCC π_2))))))
 <DRL-IFTR3 applied twice using equivalence TH-INV-SF9 to (DRL-IFTR1 applied to 8 after discharging assumptions)>

QED (Lemma-2)

Lemma-2-1)

$$\vdash (\text{IF } (\text{AND NI-CND}(\text{pi1}, \text{pi2}) \\ (\text{EXECUTABLE pi1}) \\ (\text{IFTRIED pi1 (EXECUTABLE pi2)})) \\ (\text{IF (OCC pi2) (IFTRIED pi1 (OCC pi2))}))$$

Proof of Lemma-2-1

- 1) NI-CND(pi1, pi2)
 <assumption>
- 2) (EXECUTABLE pi1)
 <assumption>
- 3) (IFTRIED pi1 (EXECUTABLE pi2))
 <assumption>
- 4) (OCC pi2)
 <assumption>
- 5) $\vdash (\text{IF (OCC pi2)} \\ (\text{IFF (IFTRIED pi2 (OCC pi2)) (OCC pi2)})) \\ \text{<AX-IFTR3 with substitutions [pi2/pi, (OCC pi2)/P]>}$
- 6) $4 \vdash (\text{IFF (EXECUTABLE pi2) (OCC pi2)}) \\ \text{<MP applied to 4,5 and EXECUTABLE substituted for its definition>}$
- 7) $4 \vdash (\text{EXECUTABLE pi2}) \\ \text{<MP applied to 4, (ONLY-IF part of 6)>}$
- 8) $1,2,4 \vdash (\text{AND (IF (IFTRIED pi2 (EXECUTABLE pi1)}) \\ (\text{IF (OCC pi1)} \\ (\text{IFTRIED pi2 (OCC pi1)})) \\ (\text{IF (IFTRIED pi1 (EXECUTABLE pi2)}) \\ (\text{IF (OCC pi2)} \\ (\text{IFTRIED pi1 (OCC pi2)}))))) \\ \text{<MP applied to (AND-INTRO applied to 2,7),1>}$
- 9) $1-4 \vdash (\text{IF (OCC pi2) (IFTRIED pi1 (OCC pi2)})) \\ \text{<MP applied to 3, (AND-ELIM applied to 8)>}$
- 10) $1-4 \vdash (\text{IFTRIED pi1 (OCC pi2)}) \\ \text{<MP applied to 4,9>}$
- 11) $1-3 \vdash (\text{IF (OCC pi2) (IFTRIED pi1 (OCC pi2)})) \\ \text{<discharging assumption 4 in 11>}$
- 12) $\vdash (\text{IF (AND NI-CND}(\text{pi1}, \text{pi2}) \\ (\text{EXECUTABLE pi1}) \\ (\text{IFTRIED pi1 (EXECUTABLE pi2)})) \\ (\text{IF (OCC pi2) (IFTRIED pi1 (OCC pi2)})) \\ \text{<discharging the assumptions in 11>}$

Q.E.D. (Lemma-2-1)

Lemma-3)

$$\vdash (\text{IF } (\text{INEV } i \text{ (IF } A \text{ (IF } (\text{OCC } \pi_3) \text{ (IF } (\text{OCC } \pi_2) P)))) \\ (\text{INEV } i \text{ (IF } A \text{ (IF } (\text{OCC } (\text{COMP } \pi_3 \pi_2)) P))))$$

Proof of Lemma-3

- 1) A
 <assumption>
- 2) (OCC π_3)
 <assumption>
- 3) (OCC π_2)
 <assumption>
- 4) (IF A (IF (OCC π_3) (IF (OCC π_2) P)))
 <assumption>
- 5) 1-4 \vdash P
 <MP applied to 3, (MP applied to 2, (MP applied to 1,4))>
- 6) 1,4 \vdash (IF (AND (OCC π_3) (OCC π_2)) P)
 <discharging assumptions 2 and 3 in 5>
- 7) 1,4 \vdash (IF (OCC (COMP $\pi_3 \pi_2$)) P)
 <DRL-IFTR9 applied to 6 using equivalence AX-IL5>
- 8) 4 \vdash (IF A (IF (OCC (COMP $\pi_3 \pi_2$)) P))
 <discharging assumption 1 in 7>
- 9) \vdash (IF (INEV i (IF A (IF (OCC π_3)
 (IF (OCC π_2) P))))
 (INEV i (IF A (IF (OCC (COMP $\pi_3 \pi_2$))
 P))))
 <DRL-IFTR1 applied to 8 after discharging assumption 4>

QED (Lemma-3)

Lemma-4)

$$\vdash (\text{IF } (\text{AND } (\text{INEV } i \text{ (IF } A \text{ (IF } P \text{ (IFTRIED } \pi_1 \text{ (OCC } \pi_3)))))) \\ (\text{INEV } i \text{ (IF } A \text{ (IF } P \text{ (IFTRIED } \pi_1 \text{ (OCC } \pi_2)))))) \\ (\text{INEV } i \text{ (IF } A \text{ (IF } P \text{ (IFTRIED } \pi_1 \text{ (OCC (COMP } \pi_3 \pi_2)))))))$$

Proof of Lemma-4

- 1) A
 <assumption>
- 2) P
 <assumption>
- 3) (IF A (IF P (IFTRIED π_1 (OCC π_3))))
 <assumption>
- 4) (IF A (IF P (IFTRIED π_1 (OCC π_2))))
 <assumption>
- 5) 1-3 \vdash (IFTRIED π_1 (OCC π_3))
 <MP applied to 2, (MP applied to 1,3)>
- 6) 1,2,4 \vdash (IFTRIED π_1 (OCC π_2))
 <MP applied to 2, (MP applied to 1,4)>
- 7) 1-4 \vdash (IFTRIED π_1 (AND (OCC π_3) (OCC π_2)))
 <DRL-IFTR5 applied to 5,6>
- 8) 1-4 \vdash (IFTRIED π_1 (OCC (COMP $\pi_3 \pi_2$)))
 <DRL-IFTR3 applied to 7 using equivalence AX-IL5>
- 9) 3,4 \vdash (IF A (IF P (IFTRIED π_1 (OCC (COMP $\pi_3 \pi_2$))))
 <discharging assumption 2 then assumption 1 in 8>
- 9) \vdash (IF (AND(INEV i (IF A (IF P (IFTRIED π_1 (OCC π_3))))
 (INEV i (IF A (IF P (IFTRIED π_1 (OCC π_2))))))
 (INEV i (IF A (IF P (IFTRIED π_1 (OCC (COMP $\pi_3 \pi_2$)))))))
 <DRL-IFTR3 applied using equivalence TH-INV-SF9 to (DRL-IFTR1 applied to 9 after
 discharging assumptions) >

QED (Lemma-4)