Center for Foundations of Intelligent Systems

Technical Report
98-15

Normal Forms and Syntactic Completeness Proofs for Functional Independencies (final version)

D. Wijesekera, M. Ganesh, J. Srivastava and A. Nerode

November 1998

CORNELL UNIVERSITY

625 Rhodes Hall, Ithaca, NY 14853 (607) 255-8005

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 1 March 1999	3. REPORT TYPE AND DATES COVERED TECHNICAL

4. TITLE AND SUBTITLE
NORMAL FORMS AND SYNTACTIC COMPLETENESS PROOFS FOR FUNCTIONAL INDEPENDENCIES (final version)

5. FUNDING NUMBERS
DAAH04-96-1-0341

6. AUTHOR(S)
D. WIJESEKERA, M. GANESH, J. SRIVASTAVA, and A. NERODE

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Regents of the University of California
c/o Sponsored Projects Office
336 Sproul Hall
Berkeley, CA 94720-5940

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 35873.134-MA-MUR

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

We prove normal form theorems of a complete axiom system for the inference of functional dependencies and independencies in relational databases. We also show that all proofs in our system have a normal form where the application of independency rules is limited to three levels. Our normal form results in a faster proof-search engine in deriving consequences of functional independencies. As a result, we get a new construction of an Armstrong relation for a given set of functional dependencies. It is also shown that an Armstrong relation for a set of functional dependencies and independencies do not exist in general, and this generalizes the same result valid under the closed-world assumption.

14. SUBJECT TERMS
completeness proofs, data mining, functional dependencies, integrity constraints

15. NUMBER OF PAGES
42

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Technical Report
98-15

Normal Forms and Syntactic Completeness Proofs for Functional Independencies (final version)

D. WIJESEKERA, M. GANESH, J. SRIVASTAVA
AND A. NERODE

November 1998

Normal Forms and Syntactic Completeness Proofs for Functional Independencies

Duminda Wijesekera[1], M. Ganesh[2], Jaideep Srivastava[3] and Anil Nerode[4]
Honeywell Space Systems, Clearwater, FL[1]
Lawrence Livermore National Laboratory, Livermore, CA[2]
Dept. of Computer Science, Univ. of Minnesota Minneapolis, MN 55455[3]
Dept. of Mathematics, Cornell University, Ithaca, NY 14853[4]
e-mail: {wijesek|ganesh|srivasta}@cs.umn.edu, anil@math.cornell.edu

Abstract

We prove normal form theorems of a complete axiom system for the inference of functional dependencies and independencies in relational databases. We also show that all proofs in our system have a normal form where the application of independency rules is limited to three levels. Our normal form results in a faster proof-search engine in deriving consequences of functional independencies. As a result, we get a new construction of an Armstrong relation for a given set of functional dependencies. It is also shown that an Armstrong relation for a set of functional dependencies and independencies do not exist in general, and this generalizes the same result valid under the closed-world assumption.

Keywords: Completeness Proofs, Data Mining, Functional Dependencies, Integrity Constraints

This work is supported by Air Force contract number F30602-96-C-0130 to Honeywell Inc, via subcontract number B09030541/AF to the University of Minnesota, and DOD MURI grant DAAH04-96-10341 to Cornell University

1 Introduction

Databases have proven to be a very useful tool for the storage, retrieval, and manipulation of data in an organized, and systematic fashion. Commercial database systems have matured over the years and have been successfully utilized in various business and scientific applications, resulting in a multi-billion dollar industry [Gra95, Yan95]. Although newer and later developments have an object-oriented flavor to varying degrees, the basic framework of databases were developed on relational technology [Cod70]. At the heart of this successful paradigm are two simple but overwhelmingly strong abstractions of storing data in tables and a non-procedural language to query such tables.

Additional constraints that need to be imposed between data tables or between attribute values of the same table have to be imposed by specifying extra conditions. *Functional dependencies*, which are constraints between values of sets of attributes in a data table is the focus of this paper. A set of attributes Y is said to be functionally dependent on a set of attributes X (denoted $X \rightarrow Y$) if any two rows that have the same values for attributes in X, also have the same values for attributes in Y. Data dependencies of various kinds were defined and investigated as a means of specifying and enforcing known relationships between entities in a database. Relations in which given types of dependencies hold among entities result in particular normal forms [BBG78], thereby making cleaner and more modular data tables. The modularity is necessary to maintain proper semantics during insert, delete and update operations [Var88]. There are algorithms that automatically produce normalized designs of logical data models from specifications of dependencies that exist between attributes in a relation [Ull88].

In addition to enforcing semantic constraints, functional dependencies have many other uses such as in semantic query optimization [Bel96, Dec87], *data cleansing*, where the nature of schema can be used to identify invalid entries and correct some erroneous entries, in schema integration, in database restructuring [CAdS84, MR94, MR92a], and in knowledge categorization [PS92]. The publication [MR94] lists other applications of functional dependencies.

If functional dependencies are known at schema design time, they can be used in the design process itself. Conversely, over the years there has been a lot of collected data, without apriori knowledge about their dependencies, requiring the need to mine for functional dependencies from attribute values in databases. In process of mining for dependencies the search for dependencies holding in the given state of the database can be enhanced by accounting for logical consequences of already mined ones, thereby using the well known inference rules for functional dependencies, commonly referred to as *Armstrong's axioms* [Arm74], which we shall refer to as *Armstrong's Rules.*

Algorithms that mine for functional dependencies such as [Bel95a, Jan88] use Armstrong's rules in the stated way. In such algorithms once a functional dependency is known to fail, it is equally expedient to weed out other potential functional dependencies that would imply the invalid one. For this explicit purpose *Functional Independencies* were proposed in [Jan88]. Hence, eliminating mining for consequences of learned dependencies and independencies is facilitated by finding a set of rules to infer new dependencies and independencies from already discovered ones, and consequently a complete axiomatization of functional dependencies and independencies merit interest.

In this respect, Janas [Jan88] presented an axiomatization for both functional dependencies and independencies, which was argued to be incomplete by Bell [Bel95a, Bel95b]. We find some of the arguments presented in these two publications incomplete and inaccurate, and this paper remedies those defects.

Consequently, this paper provides a syntactic completeness proof of a complete axiomatization of functional dependencies and independencies. In the process we show that all proofs in our system have *normal forms.* The existence of normal forms can be exploited by a proof execution engine in two different ways. Firstly, the structure of proofs that we look for is restricted. Secondly, we need not search for any non-normal proofs, and that results in considerable savings in time.

1.1 Independencies and Excluded Dependencies

Simultaneously with the work of Janas [Jan88] there has been work done in *excluded function dependencies (XFD's)* [Tha88, GL90]. Both these papers use *excluded functional dependencies (XFD's)* to refer to functional dependencies that are not valid in any given instance of a database, but the notions of *completeness* used in them are remarkably different. In [GL90], a set A of XFD's are said to be complete if there is a database instance in which A constitute the set of all invalid dependencies. Using the *closed world assumption* they show how to construct an Armstrong relation from a complete set of XFD's. Conversely, the notion of completeness given in [Tha88] is the same as ours, and using the deduction theorem for closed formulae, it shows an equivalent system is complete for functional dependencies and independencies.

1.2 Related Work

Dependency theory has a long and rich history as been summerized in [FV86, Var88, Kan90] In addition to developing diverse notions of data dependencies, these works also addressed the issues of equivalence and relationships between them. In the field of dependency mining there are fewer works. Although this article does not deal directly with dependency mining, it is the main beneficiary of our work and hence we summerize some of the related works.

Mining for functional dependencies can be reduced to a computing a small cover (a set of deductively equivalent set of functional dependencies holding in a database state [Mai83]). The work reported in [MR87] provides an efficient algorithm to compute a small cover by considering possible counter examples for assumed functional dependencies (called disagree sets in [MR87]). and complexity bound of finding a small covers are given in [MR92b]. [BMT89] shows that for relations of modest sizes the algorithms presented in literature for dependency mining accomplish their task in reasonable time, thereby showing that tools such as [BMR85] that use such algorithms run with acceptable performance. [SF96] also address the problem of mining for functional dependencies from relations by constructing positive and negative covers. They maintain a set of possible dependencies and independencies in the potential positive and negative covers. They express the need to use, but do not use inference rules to expedite the process of constructing positive and negative covers. Work reported in [MR94] presents algorithms to extract functional dependencies from relations that uses optimizations other than the usage of inference rules.

1.3 Summary of Work

We show that Janas' axiomatization [Jan88] is incomplete with respect to functional dependencies and independencies and that a variant of Bell's axiomatization [Bel95b] in conjunction with the *Armstrong's Rules* is complete. These new axioms are called the FI Axioms, referring to the fact that they are axioms for functional independencies. Our approach follows the proof-theoretic tradition [Tak91] in mathematical logic.

In Section 2, we present the notations used and review appropriate concepts from logic. In Section 3 we describe various proof-theoretic properties of the FI axiom system to show soundness and completeness. In Section 4, we show that Janas' system is incomplete with respect to deriving functional dependencies and independencies. Departing from standard practice in dependency theory, in Section 5 we prove that every proof that uses FI axioms can be transformed into a proof in *normal form*. In Section 6, we derive the consequences of the normal form theorem to prove completeness. Some of the more detailed auxiliary results in this section are proved in detail in the appendix. In Section 7, we show that the FI axioms are complete for functional dependencies and independencies. In our approach, we develop consistency properties to create models and then show that a failed attempt to derive a functional independency produces a complete consistency property. In Section 8 we show the connection between Armstrong relations and our construction of counter models.

One of the advantages of our approach is that, in addition to giving direct proof-theoretic justifications of syntactic results, we also state and prove a normalization theorem. The important property of this normal form is that the application of independency axioms are limited to three levels and are in a specific order. This fact can be utilized when searching for derived independencies in that, one need only look for proofs that satisfy these conditions. Hence the running time of the proof-search procedures are reduced significantly.

2 Syntax, Semantics and Proof Rules

This section contains basic terminology used to formulate and prove the completeness theorem for functional independencies.

2.1 Syntax

Our syntax consists of the following components:

1. U is the set of all attributes.

2. Subsets of attributes (i.e., subsets of U) are denoted by upper case letters (possibly subscripted). Union of subsets X and Y is denoted by XY.

3. Attribute values are denoted by lower case letters (possibly subscripted) of corresponding attribute sets.

4. Two connectives $\rightarrow$ and $\not\rightarrow$ denote respectively dependencies and independencies, and the connective $\subseteq$ denotes subset relationship between sets of attributes.

5. Sentences of the form $(X \rightarrow Y)$ and $(X \not\rightarrow Y)$, and $(X \subseteq Y)$ where X and Y are sets of attributes as given in 2.

2.2 Semantics

A model to interpret our syntax consist of a data table that has all elements of U as attributes. For the purposes of this work, we assume that the database consists of a universal relation (i.e. all data tables in a database as one data table). Rows i and j are respectively denoted by t_i and t_j. The values of attributes corresponding to the attribute set A in row t_i is denoted by $t_i[A]$.

Definition 1 (Satisfaction) *Let $\mathcal{T}$ be a model and A, B be sets of attributes. Then:*

1. *We say a data table (model) $\mathcal{T}$ satisfies functional dependency $(A \rightarrow B)$ (Notation: $\mathcal{T} \models (A \rightarrow B)$) [EN94], if for all rows i and j of $\mathcal{T}$ if $t_i[A] = t_j[A]$, then $t_i[B] = t_j[B]$.*

2. *We say a data table (model) $\mathcal{T}$ satisfies functional independency, $(A \not\rightarrow B)$ (Notation: $\mathcal{T} \models (A \not\rightarrow B)$) if $\mathcal{T} \not\models (A \rightarrow B)$, i.e there are two rows i and j of $\mathcal{T}$ with $t_i[A] = t_j[A]$ and $t_i[B] \neq t_j[B]$.*

2.3 Rules of Inference

Rules of inference popularly know as Armstrong's axioms [EN94] are used to derive functional dependencies as listed below. Keeping with the spirit of this terminology we denote other rules of inferences by the description *Axioms*.

Armstrong's Rules (Axioms)

Reflexivity If $X \subseteq Y$ then $Y \to X$ *FD*1

Augmentation $$\frac{W \subseteq V \qquad X \to Y}{XV \to YW}$$ *FD*2

Transitivity $$\frac{X \to Y \qquad Y \to Z}{X \to Z}$$ *FD*3

Armstrong's Rules have been shown to be complete for functional dependencies [Ull88, Mai83].

In order to compute the set of valid functional dependencies in a given data table, the concept of *functional independency* was proposed. Analogous to Armstrong's Rules, Janas proposed an axiomatization [Jan88] as given below.

Janas' Rules

$$\frac{X \not\to Y}{X \not\to YZ}$$ *J*1

$$\frac{XZ \not\to YZ}{XZ \not\to Y}$$ *J*2

$$\frac{X \to Y \qquad X \not\to Z}{Y \not\to Z}$$ *J*3

The above axiomatization was claimed to be incomplete by Bell [Bel95a, Bel95b]. However no satisfactory proof was provided. In addition, Bell proposed the following axiomatization for functional independencies.

Bell's Rules

$$\frac{W \subseteq V \qquad V \not\to YW}{V \not\to Y}$$ *B*1

$$\frac{X \to Y \qquad X \not\to Z}{Y \not\to Z}$$ *B*2

$$\frac{Y \to Z \qquad X \not\to Z}{X \not\to Y}$$ *B*3

Following Bell's work we propose the following axiomatization, which in the presence of Armstrong's rules is equivalent (i.e. has the same set of theorems) as that of Bell's. The only difference between our rules and those of Bell's are that we have replaced *B*1 with *FI*1, where the set inclusion in the antecedent has been replaced by a dependency. The reason for this change, which will become clear in Section 5, is to have a dependency instead of set inclusion so as to lend proof method to a more syntactic analysis.

FI Rules (Rules for Functional Independency Inference)

$$\frac{V \to W \qquad V \not\to YW}{V \not\to Y} \quad FI1$$

$$\frac{X \to Y \qquad X \not\to Z}{Y \not\to Z} \quad FI2$$

$$\frac{Y \to Z \qquad X \not\to Z}{X \not\to Y} \quad FI3$$

By constructing appropriate consistency properties for proof rules (eg. [Fit83]), we prove that the last axiomatization is complete for functional independencies.

2.4 Equivalence of Bell's and FI Systems

The only difference between proof rules we use and those proposed by Bell [Bel95a, Bel95b] is that the antecedent $W \subseteq V$ in $B1$ has been replaced by $(V \to W)$ in the antecedent of $FI1$. In this section we show that $B1$ and $FI1$ are equivalent in the presence of Armstrong's rules. In order to so we prove $FI1$ using $B1$ and vice-versa.

Proving *FI1* using *B1*, *B3* and *FD2*

$$\cfrac{V \subseteq V \qquad \cfrac{\cfrac{V \to W}{VY \to WY}\ FD2 \qquad V \not\to YW}{V \not\to VY}\ B3}{V \not\to Y}\ B1$$

Proving *B1* using *FI1*

$$\cfrac{\cfrac{W \subseteq V}{V \to W}\ FD1 \qquad V \not\to YW}{V \not\to Y}\ FI1$$

3 Proof-Theoretic Properties

We first prove some structural theorems about proofs. In these proofs, we use notation from proof theory, such as threads in proofs, proof fragments and equivalence of proof fragments, etc. We provide the basic definitions here and refer the reader to a standard textbook in proof theory such as [Tak91] for further details. We do so because our proofs are proof-theoretic in nature, as opposed to model-theoretic proof provided by Bell. Hence we redefine some terminology to better suit our proofs. Throughout we use Σ as a set of dependencies and Σ' as a set of independencies.

3.1 Notation from Logic

Definition 2 (Rule) *A rule (of inference) is an expression of the form $\frac{S}{T}$ or of the form $\frac{S_1\ S_2}{T}$, where S, S_1, S_2 and T are sentences. In these rules, S, S_1 and S_2 are respectively called antecedents and T is called the consequent.*

Definition 3 (Proof) *A proof [Tak91] P is a tree of sentences satisfying the condition that every non-leaf node and its children constitutes an instance of an inference rule.*

Throughout this paper we use Σ and Σ' for sets of dependency and independency sentences respectively.

Following customary nomenclature, the sentences at the leaves of a proof P are called the *assumptions* of P and the sentence at the root in a proof P is called the *conclusion* of P. Also, following our convention, suppose Σ and Σ' are respectively sets of functional dependencies and independencies. Then, we say that a sentence ψ is a logical consequence of set of sentences $\Sigma \cup \Sigma'$ if there is a proof of ψ where the assumptions are taken from the set $\Sigma \cup \Sigma'$, and where the rules of inference are drawn from the FI system. Then we also say that ψ is a logical consequence of of $\Sigma \cup \Sigma'$. We use the notation $\Sigma \cup \Sigma' \vdash \psi$ to indicate so. We also write $Cn(\Sigma \cup \Sigma')$ for the set of logical consequences of $\Sigma \cup \Sigma'$. i.e. $Cn(\Sigma \cup \Sigma') = \{\psi : \Sigma \cup \Sigma' \vdash \psi\}$. Also, proof that uses only Armstrong's rules (i.e. *FD*1, *FD*2 and *FD*3) is called a FD-proof and one which involves the FI rules (i.e. *FI*1, *FI*2 and *FI*3) is called an independency proof. We use $\Sigma \vdash_{FD} \psi$ to indicate that there is a proof of ψ using assumptions from Σ with rules of inferences drawn from Armstrong's system. Similarly $\Sigma' \vdash_I \psi$ to indicate that there is a proof of ψ using assumptions from Σ' using independency rules. Similarly, $\Sigma \cup \Sigma' \vdash_{Janas} \psi$ indicates that there is a proof of ψ with assumptions drawn from $\Sigma \cup \Sigma'$ using rules of Janas' system.

Definition 4 (Threads in Proofs) *A sequence of sentences is called a thread [Tak91] if :*

- *It begins with an assumption and ends with the conclusion and*
- *All sentence in the sequence except the last is an antecedent of an inference rule and it is immediately followed by the consequent of the same inference rule.*

Definition 5 (Fragment of a Proof) *A part of a proof which itself is a proof is called a fragment of a proof (sometimes called a subproof [Tak91]).*

3.2 Domain Specific Results

Notice that our proof system is stated as a natural deduction system [Pra65] with two connectives, $\rightarrow$ and $\not\rightarrow$. In this section we prove many proof-theoretic results that would reveal the nature of deductions (i.e. proofs) in our system and eventually lead us to the proof of the completeness theorem. Some of the results proved have appeared in [Bel95b, Bel95a], but with a very different flavor of proofs. Some proofs given in [Bel95b] and [Bel95a] are inaccurate, unjustified, or lemmas used in them are unproved and non-trivial. Specifically they are as follows: In Lemma 1 of [Bel95b], it is claimed that a partially filled table can be completed without affecting Σ because a set of dependencies and independencies $\Sigma \cup \Sigma'$ is consistent. Consistency as defined in this paper says that there is some data table that satisfies $\Sigma \cup \Sigma'$, and not that a partially filled up table can be completed. Furthermore, Corollary 1 (presumably to Lemma 1) is stated without a proof, and we do not see it is a corollary to any lemma proved up to the statement. We prove this corollary by syntactic means. In Lemma 4, where the incompleteness of Janas system is shown, at one step it is claimed that J1 and J2 could not have been applied, and we do not see any trivial justification. In Theorem 2 (Completeness of Bell's system) it is not clear that the case analysis is exhaustive. To avoid such problems, we provide all necessary proofs in complete detail.

We begin by first showing that the addition of independencies does not affect the derivable dependencies, in the following lemma.

Definition 6 (Dependency Property) *We say that a proof system A has the Dependency property if $\Sigma \cup \Sigma' \vdash_A (X \rightarrow Y)$, then $\Sigma \vdash_{FD} (X \rightarrow Y)$.*

Lemma 1 (Dependence Property of FI) *If* $\Sigma \cup \Sigma' \vdash (X \to Y)$ *then* $\Sigma \vdash_{FD} (X \to Y)$

Proof: By induction on the height of the proof tree of $(X \to Y)$

Suppose that $\Sigma \cup \Sigma' \vdash (X \to Y)$. Then consider the proof t of $(X \to Y)$ from $\Sigma \cup \Sigma'$ with the minimal height. Notice that the only proof rules that have $\to$ as the main connective in the consequent could have been used in t as the last step. Hence, they have to be one of Armstrong's rules, *FD* 1, *FD* 2 or *FD*3.

Case 1. The last rule used to deduce $(X \to Y)$ is either *FD* 1 or *FD* 2.

Then t is of the form:

$$\cfrac{\cfrac{t_1}{P \to Q}}{X \to Y}$$

Thus, $\frac{t_1}{(P \to Q)}$ is a proof of $(P \to Q)$ from $\Sigma \cup \Sigma'$ with a length shorter than that of t. Hence, by the inductive assumption, there is a proof t_1' of $(P \to Q)$ from Σ. Hence, $\frac{t_1'}{(X \to Y)}$ is a proof of $(X \to Y)$ from Σ.

Case 2. The last rule used to deduce $(X \to Y)$ is *FD* 3.

Then t is of the form:

$$\cfrac{\cfrac{t_1}{P_1 \to Q_1} \quad \cfrac{t_2}{P_2 \to Q_2}}{X \to Y} \quad FD3$$

Consequently, by an argument similar to Case 1, there are proofs t_1' and t_2' respectively of $(P_1 \to Q_1)$ and $(P_2 \to Q_2)$ from Σ. Hence, the following is a proof of $(X \to Y)$ from Σ:

$$\frac{t_1' \qquad\qquad t_2'}{X \to Y}$$

■

At the heart of all our arguments is the simple but powerful fact that every proof in this system has a unique proof thread in which the major connective is $\not\to$.

Definition 7 (Independency Thread and Single Independency Thread Property)

- *A proof thread in which the connective at every step is* $\not\to$ *is said to be an independency thread.*
- *If a proof that has a unique independency thread is said to have the single independency thread property.*

Lemma 2 (Single Independency Thread Property of FI) *Every FI proof has at most one independency thread. If the conclusion is an independency then it has an independency thread, otherwise it does not have any.*

Proof: (By induction on the height of the proof tree)
Suppose t is a proof tree with the least height, of ψ from $\Sigma \cup \Sigma'$, where ψ is either a functional independency or a functional dependency.

Case 1. ψ is of the form $(X \not\rightarrow Y)$
We show by induction on the height of t that there is exactly one thread where the main connective is $\not\rightarrow$.

In this case, proof rules that could have been used in the last step are *FI* 1, *FI* 2 or *FI*3. Then t is of the following form:

$$\frac{V \rightarrow W \qquad\qquad \dfrac{t_1}{P \not\rightarrow Q}}{X \not\rightarrow Y}\quad FI2$$

Consequently, by the inductive argument, $\frac{t_1}{(P \not\rightarrow Q)}$ that has $(P \not\rightarrow Q)$ as the conclusion and a smaller height has a single thread of independencies. Hence t has a single thread of independencies; namely the thread that extends the thread in $\frac{t_1}{(P \not\rightarrow Q)}$ by adding $(X \not\rightarrow Y)$ to its bottom.

Case 2. ψ is of the form $(X \rightarrow Y)$
In this case, we show that $\not\rightarrow$ does not appear in the proof tree.

In this case, because the main connective of the conclusion is $\rightarrow$, only Armstrong's axioms (i.e. *FD*1, *FD*2 or *FD*3) could have been applied at the last step of the proof. If the last rule applied is either *FD*1 or *FD*2, then t is of the following form:

$$\frac{\dfrac{t_1}{P \rightarrow Q}}{X \rightarrow Y}$$

Then, by the inductive hypothesis $\not\rightarrow$ does not appear in $\frac{t_1}{(P \rightarrow Q)}$.
Suppose the last rule used is *FD*3, then t is of the following form:

$$\frac{\dfrac{t_1}{P_1 \rightarrow Q_1} \qquad\qquad \dfrac{t_2}{P_2 \rightarrow Q_2}}{X \rightarrow Y}\quad FD3$$

Hence, by the inductive hypothesis, $\not\rightarrow$ does not appear in either $\frac{t_i}{(P_i \rightarrow Q_i)}$ for $i = 1, 2$. ■

Corollary 1 (Independence Property: Corollary to Lemma 2) *Let Σ be a set of functional dependencies and Σ' be a set of functional independencies. If $\Sigma \cup \Sigma' \vdash (X \not\rightarrow Y)$ then there are some R, S such that $(R \not\rightarrow S)$ and $\Sigma \cup \{(R \not\rightarrow S)\} \vdash (X \not\rightarrow Y)$.*

Proof:
Suppose t is a proof of $(X \not\rightarrow Y)$ from $\Sigma \cup \Sigma'$. Then, by Lemma 2, t has a unique independency thread. Let $(R \not\rightarrow S)$ be at the head of this independency thread. Then, $(R \not\rightarrow S)$ is the only functional independency that is being used as an assumption in t. Hence t is a proof of $(X \not\rightarrow Y)$ from $\Sigma \cup \{(R \not\rightarrow S)\}$. ■

4 Incompleteness of Janas' System

In this section, using proof-theoretic arguments we show that proof rules of Janas are incomplete for functional independencies. In particular, as stated by Bell [Bel95a, Bel95b], we show that the following proof rule is sound, but cannot be derived in Janas' rule system.

$$\frac{X \to Y \qquad\qquad Z \not\to Y}{Z \not\to X}$$

In order to justify our claim, we need some properties about Janas' system, which are in the following lemmas.

Lemma 3 *1. The following proof rule is sound.*

$$\frac{X \to Y \qquad\qquad Z \not\to Y}{Z \not\to X}$$

2. Janas' system has the single independency thread property.

3. It has the dependence property.

Proof:
For the proof of (1), which is a rather trivial fact, see [Bel95b], and [Bel95a]. Proof of (2) and (3) are similar to the corresponding proofs in our FI system.
■

Lemma 4 (Incompleteness of Janas' Proof System) *The following proof rule cannot be derived in Janas' system of rules.*

$$\frac{X \to Y \qquad\qquad Z \not\to Y \qquad \text{F}3}{Z \not\to X}$$

Proof: This can be easily seen semantically. A detailed syntactic proof appears in the appendix.
■.

5 Normal Forms for Proofs

In this section, we prove a normal form theorem for proofs in our system. We show that every proof in our system is equivalent to one in which there are at most three applications of FI axioms in the order *FI3*, *FI1*, *FI2*.

Towards this end we need some auxiliary facts, which are summarized below.

- Repeated applications of any independency rule can be replaced by a single application of the same rule.
- The order of *FI2* and *FI3* can be interchanged.
- The order of applications of independency rules *FI2*, *FI1* or *FI1*, *FI3* can be reversed, but not vice-versa.

Section 5.1 is devoted to precise statements of these facts, which are proved in the appendix.

5.1 Auxiliary Facts

Lemma 5 (Proof Rule Merging) *The following facts hold about repeated applications of proof rules.*

1. *A sequence of successive applications of* FI*1 is equivalent to a single application of* FI*1, i.e. given a proof t where the single independency thread has a sequence of applications of* FI*1, is equivalent to a proof that has a single application of* FI*1.*
2. *A sequence of successive applications of* FI*2 is equivalent to a single application of* FI*2, i.e. given a proof t where the single independency thread has a sequence of applications of* FI*2 is equivalent to a proof that has a single application of* FI*2.*
3. *A sequence of successive applications of* FI*3 is equivalent to a single application of* FI*3, i.e. given a proof t where the single independency thread has a sequence of applications of* FI*3 is equivalent to a proof that has a single application of* FI*3.*

Proof:
See Appendix A. ∎

Lemma 6 (Proof Rule Interchangeability) *The following facts hold about the interchangeability of inference rules in FI proofs.*

- *For every proof fragment in which* FI*3 is applied immediately after* FI*2, there is an equivalent proof fragment in which* FI*2 is applied after* FI*3.*
- *The following hold for the reversal of application orders of rules* FI*1,* FI*2 and* FI*3.*
  1. *For every proof fragment in which* FI*1 is applied immediately after* FI*2, there is an equivalent proof fragment in which* FI*2 is applied after* FI*1.*
  2. *For every proof fragment in which* FI*3 is applied immediately after* FI*1, there is an equivalent proof fragment in which* FI*1 is applied after* FI*3.*

Proof:
See Appendix A. ∎

Using Lemmas 5 and 6, we show that every proof in our system can be reduced to a normal form. In this normal form, every proof has at most three applications of functional independency rules, and furthermore they are applied in the order *FI* 3, *FI* 1 and *FI* 2. Accordingly, we define *normal forms* for proofs.

5.2 Proof of the Normal Form Theorem

In this section we state and prove the normal form theorem.

Definition 8 (Normal Form) *A proof is said to be in Normal Form if and only if its unique independency thread has atmost three applications of independency rules in the order* FI *3,* FI *1,* FI *2, if they do appear at all.*

Now, we show a weak normalization theorem, namely that every proof in in our system has a normal form. The proof of the normal form theorem, while syntactic in nature, consists of three main steps. In the first step, we use the lemma 5, and reduce successive applications of the same independency rule to a single application of the rule, resulting in a proof without successive applications of the same rule. This lets us visualize the independency thread as consisting of a sequence of *blocks* where each block begins by an application of *FI1* , and is followed by an application of either *FI2* or *FI3*, followed by the other rule. Then we show that interchangeability lemmas can be used to reduce such a proof segment to the order *FI3*, *FI1*, *FI2*. Lastly, we show that any two successive blocks can be reduced to a single block.

Definition 9 (Block) *A fragment of a proof is said to be a block if it has an independency thread in which either:*

- *There are at most three applications of distinct independency rules, of which the first one is* FI*1, and the other two are applications of distinct independency rules* FI*2 and* FI*3 in any order.*
- *Or there are at most two applications of distinct independency rules* FI*2 and* FI*3 in any order.*

Definition 10 (Normal Block) *A fragment of a proof is said to be a normal block if it is a block in which the independency rules are applied in the order* FI*3,* FI*1,* FI*2.*

Lemma 7 (Blocking of Proofs) *Suppose* $\Sigma \cup \Sigma' \vdash (X \not\rightarrow Y)$*. Then there is a proof t of* $(X \not\rightarrow Y)$ *in which the unique independency thread consists of a sequence of blocks, of which only the first block (i.e. the block at the top of the independency thread) may miss an application of* FI*1.*

Proof:
Suppose $\Sigma \cup \Sigma' \vdash (X \not\rightarrow Y)$. Then there is a proof t_1 of $(X \not\rightarrow Y)$ from $\Sigma \cup \Sigma'$. By applying lemma 5 to t_1, we obtain a proof t_2 of $(X \not\rightarrow Y)$ from $\Sigma \cup \Sigma'$, that does not contain successive applications of *FI1*, *FI2* or *FI3*.

Then, define the blocks in t_2 as the proof segments starting with any application of *FI1* and extending up to, but excluding the next application of *FI1* along the unique independency thread. If the first rule of application is not *FI1*, then the first block may contain *FI2* and/or *FI3* in any order. ■

Lemma 8 (Block Normalization) *For every block there is an equivalent normal block.*

Proof:
Suppose b is a block. Then, by definition, the unique independency thread of b does not have an application of *FI1*, in which case (if need be) Lemma 6 can be used interchange the application order of rules *FI2* and *FI3* to make it a normal block, or it has an application of *FI1* at the top of the independency thread, i.e. at the beginning of the independency thread.

If the first rule of application in the independency thread is *FI1*, and the order of application of other rules is *FI2*, *FI3*, then by Lemma 6, there is an equivalent proof fragment b_1 where the rules are applied in the order *FI1*, *FI3*, *FI2*. Then, by Lemma 6 there is an equivalent normal block b_{normal}. ■

Lemma 9 (Normal Block Merging) *A sequence of two normal blocks can be reduced to a normal block.*

Proof:
Suppose a proof fragment t consists of two successive normal blocks a and b. Let the blocks a and b both have the application of all three independency rules *FI3*, *FI1* and *FI2*, respectively denoted as a_3, a_1, a_2 and b_3, b_1, b_2. By using the interchangeability lemmas and merging they can be transformed into a proof in normal form, as given below.

1. Apply Lemma 6 to get a proof segment in which the order is $a_3, a_1, b_3, a_2, b_1, b_2$.

2. Apply Lemma 6 to get a proof segment in which the order is $a_3, b_3, a_1, a_2, b_1, b_2$.

3. Apply Lemma 6 to get a proof segment in which the order is $a_3, b_3, a_1, b_1, a_2, b_2$.

4. Apply Lemma 5 to respectively merge successive applications of rules *FI1*, *FI2* and *FI3* in a_3, b_3, a_1, b_1, and a_2, b_2 to a single application of respective proof rules.

In the cases of degenerate blocks, i.e., where application of one or two independency rules are missing, we can still apply the same procedure to group application of similar rules together. Some steps in the process will have become redundant because of the absence of some of the independency rule applications. The details are given in the appendix. ∎

Theorem 1 (Normal Form Theorem for Proofs) *Suppose* $\Sigma \cup \Sigma' \vdash (X \not\twoheadrightarrow Y)$. *Then there is a normal form proof of* $(X \not\twoheadrightarrow Y)$ *from* $\Sigma \cup \Sigma'$.

Proof:
Suppose t is a proof of $(X \not\twoheadrightarrow Y)$ from $\Sigma \cup \Sigma'$. Then:

1. Apply transformations given in the Lemma 7 to obtain a proof t_1 in which the independency thread consists of blocks.

2. Apply transformations given in the Lemma 8 to every block in t_1 to obtain an equivalent proof t_2 in which every successive block is a normal block.

3. Inductively apply the transformation given in Lemma 9 to blocks of t_2 to obtain an equivalent proof t_3, which consists of a single block.

∎

5.3 Proof-Theoretic Properties of Functional Dependencies

In this section, we show some proof-theoretic properties that are used in constructing Armstrong relations.

Lemma 10 (Merging Lemma for Functional Dependencies) *Successive applications of the Augmentation Rule (i.e.* FD2*) is equivalent to a single application of* FD2*; i.e., given a proof t in which there are two successive applications of* FD2 *on a proof thread, they can be replaced with a single application of* FD2.

Proof:
See Appendix A.

Lemma 11 (Interchange Lemma for Functional Dependencies) *Any proof fragment in which the order of application is* FD*3*, FD*2 can be replaced by a proof fragment in which the order of application is* FD*2*, FD*3*.

Proof:
See Appendix A.

The results in Lemmas 11 and 10 can be combined to show that all proofs for FD's can be transformed in to a standard form called *semi-normal form.*

Definition 11 (Semi-Normal Form for Functional Dependency Proofs) *We say that a proof t of a functional dependency is in semi-normal form if it satisfies the following properties.*

- *The application of* FD*2 in t is limited to once per proof thread in t.*
- *If* FD*2 is applied in a proof thread in t, then it is applied to the top sequent of the thread.*

We now show that every proof in FD has a semi-normal form.

Theorem 2 (Semi-Normal Form Theorem for FD Proofs) *Any proof of a functional dependency* $(X \to Y)$ *can be transformed to a proof in semi-normal form.*

Proof:
By applying Lemma 10, successive applications of *FD*2 can be replaced by a single application of *FD*2, and by applying Lemma 11, applications of *FD*2 can be pushed upto the top sequents of proof threads. ■

In the next theorem we show that for any proof in which any given dependency $X \to Y$ appears more than once as an assumption can be replaced with an equivalent proof in which it appears only once as an assumption. To prove this result, the following definition is in order.

Definition 12 (Transitive Envelope of a FD Proof Tree) *Consider a FD proof t in semi-normal form. Suppose* $\Gamma_1, \ldots, \Gamma_n$ *is a left-to-right listing of all proof threads of t. Then a listing* $(X_1 \to Y_1), \ldots, (X_n \to Y_n)$ *of functional dependencies satisfying the following properties is called the transitive envelope of t.*

- $(X_i \to Y_i)$ *is on* Γ_i *for all* $i \leq n$. *Suppose the position at which* $(X_i \to Y_i)$ *appears in* Γ_i *is* γ_i.
- γ_i *is the farthest position from the conclusion of t where there are no application of* FD*2 between* γ_i *and the conclusion of t.*

The following definition states properties of transitive envelopes, which are needed in later proofs.

Definition 13 (Chains of Dependencies) *A listing of dependencies of the form* $(X \to X_1), (X_1 \to X_2), \ldots (X_n \to Y)$ *is said to be a chain of dependencies. We say that X is the head and Y is the tail of the chain.*

The next theorem proves an important property of a transitive envelope of a FD proof tree.

Theorem 3 (Structural Property of Transitive Envelopes) *Suppose t is a semi-normal form proof of* $(X \to Y)$ *and T is the transitive envelope of t. If T is non-null, then it is a chain with head X and tail Y; i.e.* $(X_1 \to X_2), \ldots (X_n \to X_{n+1})$ *where X is* X_1 *and Y is* X_{n+1}.

Proof:
See Appendix A. ■

As Theorem 3 states, the transitive envelope of a FD proof is a chain. The next theorem shows that repeated assumptions in this chain can be removed, i.e. that cycles can be removed.

Theorem 4 (Repetition Removal from Transitive Envelopes) *Every proof t of a functional dependency $(X \to Y)$ in which the transitive envelope T has a repetition of some functional dependency $(A \to B)$, can be reduced to a proof t' in which $(A \to B)$ is not repeated in its transitive envelope.*

Proof:
See Appendix A. ■

In the next theorem we show that proofs in FD can be reduced to a form where assumptions that are functional dependencies are used atmost once. In order to prove it, we need the following lemmas.

Lemma 12 (Some Useful Proof fragments) *Following are auxiliary facts.*

1. *There is a FD proof of $(Y \to WY)$ from assumptions $(Y \to A), (A \to XV)$ and $W \subseteq V$.*
2. *There is a proof of $(YV \to YB)$ from assumptions $W \subseteq V, (YW \to XA)$ and $B \subseteq A$.*

Proof:
See Appendix A.

The results such Lemma 12 state some obvious monotonicity facts about $\to$ and $\not\to$ with respect to $\subseteq$.

Lemma 13 (Fusing FD Proofs) *Suppose $t_1, \dots t_n$ is a sequence of proofs that have respectively, $(A_1 \to A_2), \dots, (A_n \to A_{n+1})$ as their conclusions; then there is a proof of $(A_1 \to A_{n+1})$ that has the same assumptions as those of $t_1, \dots t_n$.*

Proof:
By applying *FD*3 repeatedly, we can create a proof of $(A_1 \to A_{n+1})$ from the chain $(A_1 \to A_n), \dots (A_n \to A_{n+1})$. By fusing the proof trees $t_i - 1$ on top of $(A_i \to A_{i+1})$ for all $2 \leq i \leq n$, we get the desired result. ■

Now, we use lemma 12 to generalize Theorem 4.

Theorem 5 (Repetition Removal from Assumptions) *For every proof t of $(E \to F)$ in FD and every assumption $(X \to Y)$ used in t, there is an equivalent proof t' in which the assumption $(X \to Y)$ is used at most once.*

Proof:
See Appendix A. ■

5.4 Proof Inversions

In this section, we show that proofs that assert a functional dependency can be constructively transformed into proofs that assert functional independencies, and vice versa. Specifically, we show that if $\sum \cup \{X \not\to Y\} \vdash (P \not\to Q)$ then $\sum \cup \{P \to Q\} \vdash (X \to Y)$ and vice versa. This fact is later used in the proof of the completeness theorem. The results contained in this section seemed trivial from semantic consideration. They are stated for the sake of completeness sake and to show that the syntactic method used throughout this paper is capable of showing all necessary facts.

Definition 14 (Inverse Fragments) *Consider the following proof fragments.*

1. $$\frac{\dfrac{t_1}{X \to Y} \qquad \dfrac{t_2}{Y \to Z}}{X \to Z}\ \text{FD}\mathit{3}$$

2. $$\frac{\dfrac{t_1}{X \to Y} \qquad X \not\to Z}{Y \not\to Z}\ \text{FI}\mathit{2}$$

3. $$\frac{\dfrac{t_2}{Y \to Z} \qquad X \not\to Z}{X \not\to Y}\ \text{FI}\mathit{3}$$

4. $$\frac{A \supseteq B \qquad X \to Y}{AX \to BY}\ \text{FD}\mathit{2}$$

5. $$\frac{AX \to X \qquad \dfrac{\dfrac{A \supseteq B}{AX \to B} \qquad AX \not\to BY}{AX \not\to Y}\ \text{FI}\mathit{1}}{X \not\to Y}\ \text{FI 2}$$

6. $$\frac{\dfrac{t}{V \to W} \qquad V \not\to YW}{V \not\to Y}\ \text{FI}\mathit{1}$$

7. $$\frac{\dfrac{V \subseteq V \qquad \dfrac{t}{V \to W}}{V \to VW}\ \text{FD}\mathit{2} \qquad \dfrac{W \subseteq W \qquad V \to Y}{VW \to YW}\ \text{FD}\mathit{2}}{V \to YW}\ \text{FD}\mathit{3}$$

In these proof fragments, (2) and (3) are said to be respectively the left and the right inverse of (1), and conversely (1) is said to be the inverse of (2) and (3). Similarly, (4) and (5) are said to be inverses of each other and (6) and (7) are said to be the inverses of each other.

We denote the inverse of proof fragment, left inverse and right inverse proof fragment of f respectively as f^{-1}, f_L^{-1} and f_R^{-1}.

Now we show the following properties about inverse fragments.

Lemma 14 (Properties of Inverse Fragments) *The proof fragments listed in Definition 14 have the property that if the fragment prove $(P \to Q)$ from $(X \to Y)$, possibly using t_1, then its inverse fragment (if applicable, left and right inverses) proves $(X \not\to Y)$ from $(P \not\to Q)$ (the inverse* *uses t_1 if the original fragment used t_1). Conversely, if a proof fragment listed in Definition 14 proves $(P \not\rightarrow Q)$ from $(X \not\rightarrow Y)$ using the assumption t_1, then its inverse proves $(X \rightarrow Y)$ from $(P \rightarrow Q)$ using the assumption t_1.*

Proof:

The fragments and their inverses (left and right, if applicable) are listed in Definition 14, with the corresponding proof rules used to justify the fragment. ■

Definition 15 *Suppose t is a proof in FD and γ is a thread in t where the topmost sequent of γ is a functional dependency, say $(X \rightarrow Y)$. Then define the γ inverse of t (Notation $t^{-1}(\gamma)$) inductively as follows.*

Base Case:

Suppose t consist of only $(X \rightarrow Y)$. Then define $t^{-1}(\gamma)$ as $(X \not\rightarrow Y)$.

Inductive Case:

Let t' be the proof that uses the consequent of the first application of a proof rule to $(X \rightarrow Y)$ as its assumption. Let γ' be the proof thread in t' that is obtained by removing the first sequent from γ.

- *Suppose the first proof rule applied on γ is* FD*2, and let f be the proof fragment that constitute the application of* FD*2, say $\frac{B \subset A \; X \rightarrow Y}{AX \rightarrow BY}$. If f constitutes all of t, then define $t^{-1}(\gamma)$ as f^{-1}.*

  Otherwise, define $t^{-1}(\gamma)$ as the proof obtained by fusing the consequent of $t'^{-1}(\gamma')$ to the assumption that is the only functional independency in f^{-1}. The next theorem shows that this functional independency is $(AX \not\rightarrow BY)$, so that they can be fused, and the resulting tree constitutes a valid proof.

- *Suppose the first proof rule applied on γ is* FD*3, and that $(X \rightarrow Y)$ is the left antecedent of that application of* FD*3. Let f be the proof fragment corresponding to this application. If f constitutes all of t, then define $t^{-1}(\gamma)$ as f_L^{-1}.*

  Otherwise, define $t^{-1}(\gamma)$ to be the proof obtained by fusing the consequent of $t'^{-1}(\gamma')$ to the assumption that is the only functional independency in f_L^{-1}, say $(A \not\rightarrow B)$. The next theorem shows that $t'^{-1}(\gamma')$ proves $(A \not\rightarrow B)$, so that they can be fused, and the resulting tree constitutes a valid proof.

- *Suppose the first proof rule applied on γ is* FD*3, and that $(X \rightarrow Y)$ is the right antecedent of that application of* FD*3. Let f be the proof fragment corresponding to this application. If f constitutes all of t, then define $t^{-1}(\gamma)$ as f_R^{-1}.*

  Otherwise, define $t^{-1}(\gamma)$ to be the proof obtained by fusing the consequent of $t'^{-1}(\gamma')$ to the assumption that is the only functional independency in f_R^{-1}, say $(A \not\rightarrow B)$. The next theorem show that $t'^{-1}(\gamma')$ proves $(A \not\rightarrow B)$, so that they can be fused, and the resulting tree constitutes a valid proof.

Let t be a proof in which functional independency $(X \not\rightarrow Y)$ is at the top of the unique independency thread. Then define the inverse of t (Notation t^{-1}) inductively as follows.

Base Case:

Suppose t consist of only $(X \not\rightarrow Y)$. Then define t^{-1} as $(X \rightarrow Y)$.

Inductive Case:
Let t' be the proof that takes the consequent of the first application of the appropriate proof rule to $(X \not\to Y)$. Then define t^{-1} as follows.

*Let f be the proof fragment corresponding to the first application of a proof rule (*FI*1,* FI*2 or* FI*3). If f constitutes all of t, then define the inverse of t to be f^{-1}*

Otherwise, define t^{-1} to be the proof obtained by fusing the consequent of t'^{-1} to the right assumption of proof fragment of f^{-1}. The next theorem shows that they can be fused, and the resulting tree constitutes a valid proof.

In the next theorem, we show that the results shown in Lemma 14 about inversions of proof fragments carry over to inversions of complete proofs.

Theorem 6 (Properties of Inverse Proofs) *The proofs listed in Definition 15 have the property that if t proves $(P \to Q)$ from $(X \to Y)$, possibly using a set of functional dependencies, say Σ, then its inverse proof indexed by a thread γ, $t^{-1}(\gamma)$ proves $(X \not\to Y)$ from $(P \not\to Q)$, possibly using Σ. Conversely, if t proves $(P \not\to Q)$ from $(X \not\to Y)$ and other dependencies Σ, then, t^{-1} proves $(X \to Y)$ from $\Sigma \cup \{P \to Q\}$.*

Proof:
See Appendix A.

6 Consequences of the Normal Form Theorem

In this section, we explore the consequences of the normal form theorem which are relevant in the completeness proof. In order to do so, we need to define *consistency* for a set of sentences.

Definition 16 *We say that $\Sigma \cup \Sigma'$ is consistent if $\Sigma \cup \Sigma' \not\vdash (W \not\to V), (W \to V)$ for some sets of attributes W and V. Here Σ is a set of dependencies and Σ' is a set of independencies.*

Lemma 15 (Inconsistency Test) *Suppose Σ is a set of dependencies and Σ' is a set of independencies. $\Sigma \cup \Sigma'$ is inconsistent if and only if there is an independency $(P \not\to Q) \in \Sigma'$ such that $\Sigma \vdash (P \to Q)$*

Proof:
See Appendix A. ∎

Lemma 16 (Consistency when adding a dependency) *Suppose Σ is a set of dependencies and Σ' is a set of independencies. If $\Sigma \cup \Sigma'$ is consistent and $\Sigma \cup \Sigma' \not\vdash (X \not\to Y)$ then $\Sigma \cup \Sigma' \cup \{X \to Y\}$ is consistent.*

Proof:
See Appendix A. ∎

Lemma 17 (Consistency when adding an independency) *Suppose Σ is a set of dependencies and Σ' is a set of independencies. If $\Sigma \cup \Sigma'$ is consistent and $\Sigma \cup \Sigma' \not\vdash (X \to Y)$ then $\Sigma \cup \Sigma' \cup \{X \not\to Y\}$ is consistent.*

Proof:
See Appendix A. ∎

7 Completeness of the Proof System

In this section, we present the consistency properties for the *FI-system.* Then we show that every consistency property yields a model. Finally we show the completeness theorem by proving that if $\Sigma \cup \Sigma' \not\vdash \psi$, then there is a complete consistency property that satisfies $\Sigma \cup \Sigma'$ but not ψ. This kind of proofs are common in model theory of first order logic.

Definition 17 (Consistency Property) *We say that a set C is a consistency property if following hold.*

- **Non-Contradictory Nature**
  For every $R, S \subseteq U$ not both $(R \to S) \in C$ and $(R \not\to S) \in C$ hold.
- **Closure Under Proof Rules**
  C is closed under proof rules; i.e

  1. *If $X \subseteq Y$ then $(Y \to X) \in C$*
  2. *If $(B \to A) \in C$ and $X \subseteq Y$ then $(YB \to XA) \in C$*
  3. *If $(X \to Y)$, $(Y \to Z) \in C$ then $(X \to Z) \in C$.*
  4. *If $(B \to A)$, $(B \not\to Y) \in C$ then $(A \not\to Y) \in C$.*
  5. *If $(Y \to Z)$, $(X \not\to Z)$ $\in C$ then $(X \not\to Y)$ $\in C$.*
  6. *If $(X \to Y)$, $(X \not\to Z)$ $\in C$ then $(Y \not\to Z)$ $\in C$.*

- **Disjunctive Nature of $\not\to$**
  If $S = \{S_i : 1 \leq i \leq n\}$ where each S_i is a single attribute, and $(R \not\to S) \in C$, then $(R \not\to S_i) \in C$ for some $i \leq n$.

Definition 18 (Complete Consistency Property) *We say that a set C is a complete consistency property if following properties hold.*

- *C is closed under the proof rules given in Definition 17.*
- *For every $R, S \subseteq U$ one and only one of $(R \to S) \in C$, $(R \not\to S) \in C$. hold.*

The next lemma show an important property of complete consistency properties.

Lemma 18 (Conjunctive and Disjunctive Nature of Consistency Properties) *Suppose C is a set of dependencies and independencies.*

- **Conjunctive Nature of $\to$**
  If C is a consistency property, then it satisfies the conjunctive nature of $\to$. i.e., if $S = \{S_i : 1 \leq i \leq n\}$ and $(R \to S) \in C$, then $(R \to S_i) \in C$ for all $i \leq n$.
- **Disjunctive Nature of $\not\to$**
  If C is a complete consistency property, then it satisfies the disjunctive nature of $\not\to$. i.e., if $S = \{S_i : 1 \leq i \leq n\}$ where each S_i is a single attribute, and $(R \not\to S) \in C$, then $(R \not\to S_i) \in C$ for some $i \leq n$.

Proof:
The conjunctive nature of $\rightarrow$ holds in a consistency property $\mathcal{C}$ because $(R \rightarrow S) \vdash_{Armstrong} (R \rightarrow S_i)$, and $\mathcal{C}$ is closed under deduction.

If the disjunctive nature of $\not\rightarrow$ is not true in a complete consistency property $\mathcal{C}$, then there is $S = \{S_i : 1 \leq i \leq n\}$ where each S_i is a single attribute, and $(R \not\rightarrow S) \in \mathcal{C}$, but $(R \not\rightarrow S_i) \notin \mathcal{C}$ for all $i \leq n$. Then $(R \rightarrow S_i) \in \mathcal{C}$. But $\{R \rightarrow S_i : 1 \leq i \leq n\} \vdash (R \rightarrow S)$, leading to a contradiction, because now $(R \rightarrow S)$, $(R \not\rightarrow S) \in \mathcal{C}$.

■

Lemma 19 (Complete Consistency Property and Consistency Property) *Every complete consistency property is a consistency property.*

Proof:
Suppose $\mathcal{C}$ is a complete consistency property. By Lemma 18, $\mathcal{C}$ satisfies the disjunctive nature of $\not\rightarrow$. Hence, C is a consistency property. ■

7.1 Constructing Models from Consistency Properties

Theorem 7 (Constructing Models) *If $\mathcal{C}$ is a consistency property, then there is a model $M(\mathcal{C})$ with the following properties*

1. *If $(R \not\rightarrow S) \in \mathcal{C}$ then $M(\mathcal{C}) \models (R \not\rightarrow S)$.*
2. *If $(R \rightarrow S) \in \mathcal{C}$ then $M(\mathcal{C}) \models (R \rightarrow S)$.*
3. *If $\mathcal{C}$ is a complete consistency property then $M(\mathcal{C}) \models (R \not\rightarrow S)$ implies $(R \not\rightarrow S) \in \mathcal{C}$.*
4. *If $\mathcal{C}$ is a complete consistency property then $M(\mathcal{C}) \models (R \rightarrow S)$ implies $(R \rightarrow S) \in \mathcal{C}$.*

Proof:
In this construction, we assume that the domain of every attribute can take at least countably many values. First, we construct the model $M(\mathcal{C})$ as follows.
Construction :
Let U be the set of all attributes. We construct the model $M(\mathcal{C})$ in stages, i, called the i^{th} segment $M_i(\mathcal{C})$ of $M(\mathcal{C})$. Each $M_i(\mathcal{C})$ consisting of two rows of a table (model) as follows.

1. Suppose $A = \bigcup\{B \subseteq U : (\emptyset \rightarrow B) \in \mathcal{C}\}$ For each attribute $A_i \in A$, let a_i be an attribute value valid in its domain.
2. For each attribute $S \notin A$ (where $1 \leq i \leq n$) where there is some set of attributes R satisfying the condition $R \not\rightarrow S \in \mathcal{C}$, let $\hat{S}$ be the set of all such maximal attribute sets R. Formally $\hat{S}$ can be defined to satisfy the following properties.
   - Any $R' \in \hat{S}$ satisfies $(R' \not\rightarrow S) \in \mathcal{C}$.
   - If any attribute set R'' satisfies $(R'' \not\rightarrow S) \in \mathcal{C}$ then there is some attribute set $R' \in \hat{S}$ satisfying $R' \supseteq R''$.

- If $R' \in \hat{S}$, then $R'' \notin \hat{S}$ for any proper subset R'' of R.

Let W be $\bigcup_{S \in (U \setminus A)} \hat{S}$ and $\{W_i : 1 \leq i\}$ be a listing of elements of W.

3. $\overline{W_i} = \{S : (W_i \not\to S) \in \mathcal{C}\}$. Now we construct $M_i(\mathcal{C})$ consisting of two rows (say row 0 and row 1) by filling in the attributes as follows.

   (a) For every attribute that appear in A, say A_k, fill in its value by a_k.

   (b) For each $S \in \overline{W_i}$, fill the value of S in rows 0 and 1 with $s_{i,0}$ and $s_{i,1}$ where they satisfy:
   (1) Both $s_{i,0}$ and $s_{i,1}$ are valid for their domains.
   (2) $s_{i,0} \neq s_{i,1}$.
   (3) They do not appear in any table segments created so far.

   (c) Let $W_i^+ = \bigcup\{V \subseteq U \ : \ (W_i \to V) \in \mathcal{C}\}$ Fill all the corresponding attribute values of W_i^+ in both rows 0 and 1 with the same set of values that have not appeared in any other table segment created so far.

   (d) Fill other (unfilled thus far; i.e. $U \setminus W_i^+ \setminus \overline{W_i} \setminus A$) attribute values of in both rows 0 and 1 with two sets of values that satisfy:

      - None of them have appeared in any other table segment created so far.
      - None of the corresponding component values in two rows are equal.

   These choices are possible because of the assumption that every domain of attribute values in U is countable.

4. Notice that except for attribute values filled in for W_i and A in rows 0 and 1 of the same table segment $M_i(\mathcal{C})$, none of the other attribute values are equal.

We show that our construction satisfies the required properties in the following lemma.

Lemma 20 *$M(\mathcal{C})$ constructed in Theorem 7 satisfies following properties.*

1. If $(R \not\to S) \in \mathcal{C}$ then $M(\mathcal{C}) \models (R \not\to S)$.

2. If $(R \to S) \in \mathcal{C}$ then $M(\mathcal{C}) \models (R \to S)$.

3. If $\mathcal{C}$ is a complete consistency property
$M(\mathcal{C}) \models (R \not\to S)$ implies $(R \not\to S) \in \mathcal{C}$.

4. If $\mathcal{C}$ is a complete consistency property
$M(\mathcal{C}) \models (R \to S)$ implies $(R \to S) \in \mathcal{C}$.

Proof :

To show (1):
Suppose $(R \not\to S) \in \mathcal{C}$. Then by definition of $\mathcal{C}$, $(R \not\to S_i) \in \mathcal{C}$ for some singleton subset S_i of S. Hence, by construction of $M(\mathcal{C})$, there is a maximal W_k such that $R \subseteq W_k$ and $(W_k \not\to S_i) \in \mathcal{C}$. Consequently, in $M_k(\mathcal{C})$, attribute values of W_k in rows 0 and 1 have the same values and the attribute values of S_i are distinct. Hence $M(\mathcal{C}) \models (W_k \not\to S_i)$. Hence, $M(\mathcal{C}) \models (R \not\to S_i)$. Therefore, $M(\mathcal{C}) \models (R \not\to S)$.

To show (2):
Suppose $(R \to S) \in \mathcal{C}$. Then by definition of $M(\mathcal{C})$, for all singleton subsets S_i of S, $(R \to S_i) \in \mathcal{C}$, because $\mathcal{C}$ is closed under deduction and $(R \to S) \vdash_{Armstrong} (R \to S_i)$. We show that $M(\mathcal{C}) \models (R \to S_i)$.

Notice that there is no $P \supseteq R$ with $(P \not\to S_i) \in \mathcal{C}$. For if not, then $(P \to R) \in \mathcal{C}$, (because $P \supseteq R \vdash (P \to R)$) and hence by *FI2*, $(R \not\to S_i) \in \mathcal{C}$, (because $\mathcal{C}$ is closed under deduction), contradicting $(R \to S_i) \in \mathcal{C}$. Hence, $R \not\subseteq W_k$ for any W_k used in the construction of $M_k(\mathcal{C})$. Furthermore the attribute values were chosen so that, except for the attributes from A, no two rows across distinct segments have the same attribute values. Now, if $R \subseteq A$, then $(A \to R) \in \mathcal{C}$ and hence $(\emptyset \to S_i) \in \mathcal{C}$, and hence S_i has the same attribute value across all rows, satisfying $M(\mathcal{C}) \models (R \to S_i)$. Conversely, if $R \not\subseteq A$, then there is an attribute of R that is not in A. Two distinct rows in $M(\mathcal{C})$ with the property that they have same values for attributes in R happens only when $R \subseteq W_k^+$. In this case $W_k \to R \in \mathcal{C}$ and hence $W_k \to S_i \in \mathcal{C}$ implies $M_k(\mathcal{C})$. Consequently, $M(\mathcal{C}) \models (R \to S_i)$. Because $M(\mathcal{C}) \models (R \to S_i)$ for every i, we get that $M(\mathcal{C}) \models (R \to S)$

To show (3):
Suppose $M(\mathcal{C}) \models (R \not\to S)$. Then there is a singleton subset S_k of S satisfying $M(\mathcal{C}) \models (R \not\to S_k)$. Then there are two rows in $M(\mathcal{C})$ that have the same value for attributes in R and different values for attributes of S_k. By construction, except for attributes from A, only pairs of rows from the same segment of $M(\mathcal{C})$ have equal value vectors.

Now suppose $R \subseteq A$. Then $(\phi \to R) \in \mathcal{C}$, and thus $S_k \not\subseteq A$, for if not, then $S_k \subseteq A$, an hence, by construction all rows of $M(\mathcal{C})$ have the same value for S_k. Therefore, $(A \to S_k) \notin \mathcal{C}$, for, if not, then $(A \to S_k) \in \mathcal{C}$, and hence $(\emptyset \to S_k) \in \mathcal{C}$, and hence by the definition of A, $S_k \subseteq A$. Because $\mathcal{C}$ is a complete consistency property $(A \not\to S_k) \in \mathcal{C}$. By applying *FI2* to $(A \not\to S_k) \in \mathcal{C}$ and $(A \to R) \in \mathcal{C}$, we get $(R \not\to S_k) \in \mathcal{C}$. By the deductive closure of $\mathcal{C}$, we get that $(R \not\to S) \in \mathcal{C}$.

Now suppose $R \not\subseteq A$. Hence, both rows of attributes in R that have equal values vectors must come from the same segment, (say) $M_l(\mathcal{C})$. Then $R \subseteq W_l$. To show that $(W_l \not\to S_k) \in \mathcal{C}$, notice that S_k has distinct values in rows 0 and 1 in W_l imply that $S_k \notin W_l^+$, and hence $(W_l^+ \to S_k) \notin \mathcal{C}$. Consequently, because $\mathcal{C}$ is a complete consistency property $(W_l^+ \not\to S_k) \in \mathcal{C}$. But $(W_l \not\to S_k) \vdash (R \not\to S_k)$ and $(W_l \not\to S_k) \vdash (R \not\to S)$. Because $\mathcal{C}$ is closed under deduction, we get $(R \not\to S) \in \mathcal{C}$.

To show (4):
Suppose $M(\mathcal{C}) \models (R \to S)$ and $(R \to S) \notin \mathcal{C}$. Because $\mathcal{C}$ is a complete consistency property, $(R \not\to S) \in \mathcal{C}$. By Part(2), $M(\mathcal{C}) \models (R \not\to S)$, contradicting the assumption $M(\mathcal{C}) \models (R \to S)$. ∎

7.2 Constructing Consistency Properties

In this section, we show how to produce a consistency property from an underivable sentence.

Theorem 8 *Suppose Σ is a set of functional dependencies and Σ' is a set of functional independencies and $\Sigma \cup \Sigma' \not\vdash \psi$. If $\Sigma \cup \Sigma'$ is consistent, then there is a complete consistency property $\mathcal{C}$, satisfying $\Sigma \cup \Sigma' \subseteq \mathcal{C}$ and $\psi \notin \mathcal{C}$.*

Proof:
Let $L = \{(P_i, Q_i) : 0 \leq i\}$ be a list of all pairs of subsets of U such that P_0 is X and Q_0 is Y, where ψ is either $(X \to Y)$ or $(X \not\to Y)$. By stages $\{i < \omega\}$ construct the consistency property $\mathcal{C}$ as follows:

At Stage 0:

1. If ψ is the dependency $(X \to Y)$. By Lemma 17, $\Sigma \cup \Sigma' \cup \{X \not\to Y\}$ is consistent. Then define $\mathcal{C}(0) = Cn(\Sigma \cup \Sigma' \cup \{X \not\to Y\})$.

2. If ψ is the independency $(X \not\to Y)$. By Lemma 16, $\Sigma \cup \Sigma' \cup \{X \to Y\}$ is consistent. Then define $\mathcal{C}(0) = Cn(\Sigma \cup \Sigma' \cup \{X \to Y\})$.

Notice that in both cases, $\mathcal{C}(0)$ is consistent.

At Stage $i + 1 > 0$:

1. If $\mathcal{C}(i) \vdash (P_i \to Q_i)$, define $\mathcal{C}(i+1)$ as $Cn(\mathcal{C}(i))$.

2. If $\mathcal{C}(i) \not\vdash (P_i \to Q_i)$, by Lemma 17, $\mathcal{C}(i) \cup \{P_i \not\to Q_i\}$ is consistent. Hence define $\mathcal{C}(i+1)$ as $Cn(\mathcal{C}(i) \cup \{P_i \not\to Q_i\})$

Notice that at every stage i, following hold.

- If $\mathcal{C}(i)$ is consistent, then $\mathcal{C}(i+1)$ is consistent. Therefore, at every stage i, we have that $\mathcal{C}(i)$ is consistent.

- $(P_i \to Q_i) \in \mathcal{C}(i+1)$ or $(P_i \not\to Q_i) \in \mathcal{C}(i+1)$.

Let $\mathcal{C} = \bigcup_{0 \leq i} \mathcal{C}(i)$. Then $\mathcal{C}$ is a consistency property. This is true because of the following facts.

1. $\mathcal{C}$ is non contradictory.
   There are no attribute sets R and S satisfying $(R \not\to S), (R \to S) \in \mathcal{C}$ because, $\mathcal{C}(i)$ satisfies that property for each i, due to the consistency of $\mathcal{C}(i)$, and $\mathcal{C}(i+1) \supseteq \mathcal{C}(i)$ for all $i \geq 0$. The construction at stage $i+1$ ensures that either $(P_i \to Q_i) \in \mathcal{C}(i)$ or $(P_i \not\to Q_i) \in \mathcal{C}(i)$ for all $i \geq 0$.

2. $\mathcal{C}$ is closed under deduction.
   This is because of the finitary nature of the proof rules. For suppose $\mathcal{C} \vdash \theta$, then because any proof of θ uses finitely many assumptions from $\mathcal{C}$, there is a stage i where $\mathcal{C}(i) \vdash \theta$. Hence $\theta \in \mathcal{C}(i+1)$, as $\mathcal{C}(i+1) \supseteq Cn(\mathcal{C}(i))$ implying $\theta \in \mathcal{C}$

Finally, $\psi \notin \mathcal{C}$ because of the following reasons.

- If ψ is a dependency $(X \to Y)$, then $(X \not\to Y) \in \mathcal{C}$, and because of the non-contradictory nature of $\mathcal{C}$, $(X \to Y) \notin \mathcal{C}$.

- If ψ is a independency $(X \not\to Y)$, then $(X \to Y) \in \mathcal{C}$, and because of the non-contradictory nature of $\mathcal{C}$, $(X \not\to Y) \notin \mathcal{C}$.

■

7.3 Proving the Completeness Theorem

In this section, we prove the completeness theorem.

Theorem 9 (Completeness of the Proof System) *Our proof rules are complete for functional dependencies and independencies; i.e if $M \models \psi$ whenever $M \models \Sigma \cup \Sigma'$, then $\Sigma \cup \Sigma' \vdash \psi$. Here Σ is a set of functional dependencies and Σ' is a set of functional independencies.*

Proof:

Suppose not, then there are sets Σ, Σ' and a sentence ψ satisfying $\Sigma \cup \Sigma' \not\vdash \psi$. Then we produce a model M satisfying $M \models \Sigma \cup \Sigma'$, but $M \not\models \psi$.

1. If $\Sigma \cup \Sigma' \not\vdash \psi$, then by Theorem 8, there is a complete consistency property $\mathcal{C}$ satisfying $\Sigma \cup \Sigma' \in \mathcal{C}$ and $\psi \notin \mathcal{C}$.

2. By Theorem 7, there is a model $M(\mathcal{C})$ satisfying $M(\mathcal{C}) \models \Sigma \cup \Sigma'$, and $M(\mathcal{C}) \not\models \psi$. This is true because $M(\mathcal{C}) \models \pi$ if and only if $\pi \in \mathcal{C}$ for any dependency or independency π.

■

8 Armstrong's Relations

In this section, we show the connection between Armstrong's relations [BDFS84] and our proof of completeness. For a given set of functional dependencies Armstrong's relations are relations that satisfy all those and only those functional dependencies that are logical consequences of the given set.

8.1 The Classical Case: Another Construction

Given a set of functional dependencies Σ, [BDFS84] shows how to produce an Armstrong relation. Their construction is as follows.

- Let A be defined as $\cup\{B \subseteq U : (\phi \rightarrow B) \in \mathcal{C}\}$.

- For each dependency ψ where $\Sigma \not\vdash \psi$, construct a model M_ψ with two rows that satisfy Σ but not ψ, satisfying the property that for each attribute in A take the same value across models M_ψ for all such $\Sigma \not\vdash \psi$.

- Let $M = \biguplus_{\Sigma \not\vdash \psi} M_\psi$, where $\biguplus$ is the disjoint union; i.e., M is constructed by taking all rows of all M_ψ's.

For a given set of consistent functional dependencies Σ, we can create an Armstrong relation by using our construction as follows. Suppose $\{X_i \rightarrow Y_i : i \geq 1\} = \{\psi : \Sigma \not\vdash \psi\}$. Then, we show that $\Sigma \cup \{X_i \not\rightarrow Y_i : i \geq 1\}$ is consistent. Suppose $\Sigma_n = \Sigma \cup \{X_i \not\rightarrow Y_i : i \leq n\}$. Due to lemma 1, $\Sigma_n \not\vdash_{Armstrong} (X_i \rightarrow Y_i)$. Hence, by Lemma 17, $\Sigma_n \cup \{X_i \not\rightarrow Y_i\}$ is consistent. Therefore, by induction, $\Sigma \cup \{X_i \not\rightarrow Y_i : i \geq 1\}$ is consistent and in fact it is a complete consistency condition. Therefore by Theorem 9, $\Sigma \cup \{X_i \not\rightarrow Y_i : i \geq 1\}$ has a model, say $\mathcal{M}$. Then $\mathcal{M} \models \psi$ if and only if $\Sigma \vdash \psi$. Hence $\mathcal{M}$ is an Armstrong relation. A careful examination of the construction of Theorem 20 shows that $\mathcal{M}$ has the potential of a model with a smaller number of rows than the construction given in [BDFS84].

8.2 Armstrong Relations in the presence of Independencies

In light of known results and the importance of Armstrong relations, a natural question that arises is: *given a set of functional dependencies and independencies, is there a relation that satisfies all those and only those that are logical consequences of the given set ?* The answer to this question is that in general there is no relation that can satisfy above stated requirements as shown in the following example. Suppose a relation schema has attributes A, B, C, D and E, and let the set of dependencies and independencies be $\Sigma \cup \Sigma' = \{(A \to B), (B \not\to C)\}$, where Σ is the set of dependencies and Σ' is the set of independencies. Then $\{(A \to B), (B \not\to C)\} \not\vdash (D \to E), (D \not\to E)$. Notice that there is no relation that satisfies both $(D \to E)$ and $(D \not\to E)$. The reason for the failure above is that $\Sigma \cup \Sigma' \cup \{(D \to E), (D \not\to E)\}$ is inconsistent in our proof system.

In general, following is possible. Suppose Σ and Σ' are respectively a set of dependencies and independencies where $\Sigma \cup \Sigma'$ is consistent. Then, any consistent complete extension Σ'' of $\Sigma \cup \Sigma'$ has a model, where complete means for any sets of attributes X, Y either $(X \to Y) \in \Sigma''$ or $(X \not\to Y) \in \Sigma''$. This is derivable from our theorems.

9 Use of Inference Rules

In [Bel95a] and [Bel95b] it is shown how to use proof rules in the inference of functional dependencies from data values. In this work, a Prolog-based inference engine is interleaved with the mining engine. The inference engine adds newly mined dependencies and independencies to an existing known set. When the mining engine is used, it omits mining for facts in that have already been derived, or rejected on the basis of derived independencies.

In [GJS+96] it is shown that for probabilistic functional dependencies, the performance of a mining algorithm can be enhanced based on inference rules to reject and accept already derived dependencies.

In other general data mining work such as, [AIS93] and [AS95], proof rules are not explicitly used, but based on the properties of the dependencies that is being mined for, some facts are automatically accepted or rejected. Since all that proof rules do is generate new facts from already known facts, those usage of properties can be considered as using an inference engine to some extent. One thing that a complete proof procedure does to such a process is to provide a complete set of properties that can be used in such circumstances.

10 Conclusions

In this paper, we have presented a sound and completeness axiomatization of functional independencies. We have also outlined the proof of completeness of this system using a syntactic method. One of the consequences of the completeness proof is that a straightforward method for generating the Armstrong relation for a given set of dependencies is obtained. The second advantage of this axiomatization is that we have shown that every proof in this system has a normal form with atmost three levels of application of FI-rules, and this can be used to search for proofs with very high efficiency. Consequently, as shown by Bell [Bel95a], these rules can be profitably used to mine for functional dependencies. Mining of FDs can prove useful in various situations such as semantic query optimization, database design, and database restructuring. Other applications [PK95] where the search space can be pruned using both positive and negative knowledge can also take advantage of this axiomatization. In the same vein, other data mining applications domains such as association rules, sequential patterns can also take advantage of negative knowledge of relationships, as well as the positive knowledge. Search mechanisms will only be reinforced with such capabilities.

Our goal in this direction is to produce a general mining engine which utilizes both positive and negative knowledge as discussed above. Functional dependencies present themselves as a prime candidate for this application due to their highly structured nature and well-known properties. For constructing a mining system for FDs, we have to have highly efficient and mechanizable proof methods. Such a proof method using tableaux are presented in [WGSN97].

A Appendix

Proof of the Incompleteness of Janas' System (Lemma 4)
(By Contradiction)
Consider the case where X, Y, and Z each consists of one attribute and they are all different from each other. Assume there is a proof t of $(Z \not\to X)$ from $\{X \to Y, Z \not\to Y\}$.

The plan of our proof is as follows. By induction on the number of applications of $J3$ in the independency thread, we show that there is a proof of $(Z \not\to X)$ from $\{X \to Y, Z \not\to Y\}$ that does not use $J3$. Next we show that it is impossible to prove $(Z \not\to X)$ from $\{X \to Y, Z \not\to Y\}$ only by using $J1$ and $J2$.

Suppose t is a proof of $Z \not\to X$ that uses $J3$. Then the first application of $J3$ in the independency thread must be of the following form:

$$\frac{Z \to T \qquad\qquad Z \not\to P}{T \not\to P}\quad J3$$

Hence, by By part (3) of Lemma 3, $\{X \to Y\} \vdash_{FD} (Z \to T)$. This is impossible by the completeness of Armstrong's rules and the existence of counter models for $\{X \to Y\} \not\vdash_{FD} (Z \to T)$, except when $T \subseteq Z$ or $Z \supseteq X$ and $T \supseteq Y$. But notice that $Z \supseteq X$ is impossible by our choices of attributes. In this case, the application of $J3$ is superfluous. By induction, we can argue that all subsequent applications of $J3$ are superfluous. Hence there is a proof of $F2$ in Janas' system that does not use $J3$.

Now, to show that this is impossible, suppose t does not have an application of $J3$. By part (1) of Lemma ??, t has a single independency thread. In order to apply $J2$ non-trivially, the left hand side of the independency must have more than one attribute. But, in our application we start with a single attribute set X and it does not change if the only rules applied are $J1$ and $J2$. Hence the rule $J2$ cannot be applied to our situation. The only rule applicable is $J1$. But then $X \supseteq Y$, which is a contradiction because X and Y are distinct single attribute sets. ∎

Proof of the Merging Lemma (Lemma 5):
Case 1:
To show the merging of Rule $FI1$, suppose a proof segment of t is as follows:

$$\frac{V \to U_1 \qquad \dfrac{V \to U_2 \qquad \dfrac{t'}{V \not\to YU_1U_2}\ FI1}{V \not\to YU_1}\ FI1}{\dfrac{V \not\to Y}{t''}}$$

This proof fragment is equivalent to the following:

$$\frac{V \to U_1U_2 \qquad\qquad \dfrac{t'}{V \not\to YU_1U_2}}{\dfrac{V \not\to Y}{t''}}\quad FI1$$

Case 2:
To show the merging of $FI2$, suppose there is a fragment of t of the following form.

$$\dfrac{\dfrac{\dfrac{t_2}{Y_1 \to Y_2} \qquad \dfrac{\dfrac{t_1}{X \to Y_1} \qquad \dfrac{t}{X \not\to Z}}{Y_1 \not\to Z}\ \mathit{FI2}}{Y_2 \not\to Z}\ \mathit{FI2}}{t_3}$$

Then it is equivalent to the following fragment:

$$\dfrac{\dfrac{\dfrac{\dfrac{t_1}{X \to Y_1} \qquad \dfrac{t_2}{Y_1 \to Y_2}}{X \to Y_2} \qquad \dfrac{t}{X \not\to Z}}{Y_2 \not\to Z}\ \mathit{FI2}}{t_3}$$

Case 3:
To show the merging of *FI3*, suppose there is a proof fragment with two successive applications of *FI3* of the following form:

$$\dfrac{\dfrac{\dfrac{t_3}{U_1 \to Y} \qquad \dfrac{\dfrac{t_1}{Y \to Z} \qquad \dfrac{t_2}{X \not\to Z}}{X \not\to Y}\ \mathit{FI3}}{X \not\to U_1}\ \mathit{FI3}}{t_4}$$

It is equivalent to the following proof fragment with a single application of *FI3*:

$$\dfrac{\dfrac{\dfrac{\dfrac{t_3}{U_1 \to Y} \qquad \dfrac{t_1}{Y \to Z}}{U_1 \to Z} \qquad \dfrac{t_2}{X \not\to Z}}{X \not\to U_1}\ \mathit{FI3}}{t_4}$$

■

Proof of the Interchangeability Lemma (Lemma 6):
Suppose there is a proof fragment of the following form, where *FI3* is applied following an application of *FI2*:
Case 1: (Interchangeability of *FI2* and *FI3*

$$\dfrac{\dfrac{\dfrac{t_3}{W \to Z} \qquad \dfrac{\dfrac{t_1}{X \to Y} \qquad \dfrac{t_2}{X \not\to Z}}{Y \not\to Z}\ \mathit{FI2}}{Y \not\to W}\ \mathit{FI3}}{t_4}$$

This proof fragment is equivalent to the following proof fragment.

$$\dfrac{\dfrac{\dfrac{t_1}{X \to Y} \qquad \dfrac{\dfrac{t_3}{W \to Z} \qquad \dfrac{t_2}{X \not\to Z}}{X \not\to W}\ \mathit{FI3}}{Y \not\to W}\ \mathit{FI2}}{t_4}$$

Conversely, a proof fragment of the later form is equivalent to a proof fragment of the earlier form.

Case 2: (Partial Interchangeability of *FI1* and *FI2*)

Suppose there is a proof fragment of the following form, where *FI1* is applied following an application of *FI2*:

$$\dfrac{\dfrac{\dfrac{t_1}{V \to X} \qquad \dfrac{\dfrac{t_2}{W \to V} \qquad \dfrac{t_3}{W \not\to YX}}{V \not\to YX}\; FI\,2}{V \not\to Y}\; FI\,1}{t_4}$$

Above proof fragment is equivalent to the following proof fragment in which the order of application *FI1* and *FI2* are reversed.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{t_2}{W \to V} \qquad \dfrac{t_1}{V \to X}}{W \to X} \qquad \dfrac{t_3}{W \not\to YX}}{W \not\to Y}\; FI\,1 \qquad \dfrac{t_2}{W \to V}}{V \not\to Y}\; FI\,2}{t_4}$$

■

Case 3: (Partial Interchangeability of *FI1* and *FI3*)

Suppose there is a proof fragment of the following form, where *FI3* is applied following an application of *FI1*:

$$\dfrac{\dfrac{\dfrac{t_3}{W \to Y} \qquad \dfrac{\dfrac{t_1}{V \to X} \qquad \dfrac{t_2}{V \not\to YX}}{V \not\to Y}\; FI\,1}{V \not\to W}\; FI\,3}{t_4}$$

Above proof is equivalent to the following proof fragment, in which the order of application of *FI* 1 and *FI* 3 are reversed.

$$\dfrac{\dfrac{V \to V \qquad \dfrac{\dfrac{\dfrac{\dfrac{t_1}{V \to X}}{VW \to XW} \qquad \dfrac{\dfrac{t_3}{W \to Y}}{XW \to YX}}{VW \to YX} \qquad \dfrac{t_2}{V \not\to YX}}{V \not\to VW}\; FI\,3}{V \not\to W}\; FI\,1}{t_4}$$

■

Proof of the Normal Block Merging Lemma (Lemma 9):

In the degenerate cases, reduction of all possible combinations of two normal blocks to a single normal block is shown below. The blocks are assumed to be in the order a followed by b. Due to the large number of cases we use the following notation. The application of functional independency rules in each block is denoted by the block name subscripted by the rule number; for example a_3, a_1, a_2 means that in the independency thread of a, the application of independency rules are in the order *FI3*, *FI1* and *FI2*. In this notation, we can denote all possible types of normal blocks a and b can be in, for example, where they may or may not contain applications of all the independency rules in their independency threads. We denote these cases by the digital equivalent of the binary pattern where a 1 denotes the application of a rule and a 0 denotes its absence in the normal block. For example, Pattern 56 corresponding to the binary pattern - 101 110, stands for the case where block a has *FI*3, does not have *FI*1, and has *FI*2, and block b has *FI*3, and *FI*1, but no *FI*2. We show the application of lemmas 5 and 6 through the stages in the reduction of the two normal blocks into an equivalent normal block.

Pattern 77:
$a_3a_1a_2b_3b_1b_2 \overset{Lemma6}{\longmapsto} a_3a_1b_3a_2b_1b_2 \overset{Lemma6}{\longmapsto} a_3b_3a_1a_2b_1b_2 \overset{Lemma6}{\longmapsto} a_3b_3a_1b_1a_2b_2 \overset{Lemma5}{\longmapsto} c_3c_1c_2$

Pattern 76:
$a_3a_1a_2b_3b_1 \overset{Lemma6}{\longmapsto} a_3a_1b_3a_2b_1 \overset{Lemma6}{\longmapsto} a_3b_3a_1a_2b_1 \overset{Lemma6}{\longmapsto} a_3b_3a_1b_1a_2 \overset{Lemma5}{\longmapsto} c_3c_1a_2$

Pattern 75:
$a_3a_1a_2b_3b_2 \overset{Lemma6}{\longmapsto} a_3a_1b_3a_2b_2 \overset{Lemma6}{\longmapsto} a_3b_3a_1a_2b_2 \overset{Lemma5}{\longmapsto} c_3a_1c_2$

Pattern 74:
$a_3a_1a_2b_3 \overset{Lemma6}{\longmapsto} a_3a_1b_3a_2 \overset{Lemma6}{\longmapsto} a_3b_3a_1a_2 \overset{Lemma5}{\longmapsto} c_3a_1a_2$

Pattern 73:
$a_3a_1a_2b_1b_2 \overset{Lemma6}{\longmapsto} a_3a_1b_1a_2b_2 \overset{Lemma5}{\longmapsto} a_3c_1c_2$

Pattern 72:
$a_3a_1a_2b_1 \overset{Lemma6}{\longmapsto} a_3a_1b_1a_2 \overset{Lemma5}{\longmapsto} a_3c_1a_2$

Pattern 71:
$a_3a_1a_2b_2 \overset{Lemma5}{\longmapsto} a_3a_1c_2$

Pattern 67:
$a_3a_1b_3b_1b_2 \overset{Lemma6}{\longmapsto} a_3b_3a_1b_1b_2 \overset{Lemma5}{\longmapsto} c_3c_1b_2$

Pattern 66:
$a_3a_1b_3b_1 \overset{Lemma6}{\longmapsto} a_3b_3a_1b_1 \overset{Lemma5}{\longmapsto} c_3c_1$

Pattern 65:
$a_3a_1b_3b_2 \overset{Lemma6}{\longmapsto} a_3b_3a_1b_2 \overset{Lemma5}{\longmapsto} c_3a_1b_2$

Pattern 64:
$a_3a_1b_3 \overset{Lemma6}{\longmapsto} a_3b_3a_1 \overset{Lemma5}{\longmapsto} c_3a_1$

Pattern 63:
$a_3a_1b_1b_2 \overset{Lemma5}{\longmapsto} a_3c_1b_2$

Pattern 62:
$a_3a_1b_1 \overset{Lemma5}{\longmapsto} a_3c_1$

Pattern 61:
$a_3a_1b_2$

Pattern 57:
$a_3a_2b_3b_1b_2 \overset{Lemma6}{\longmapsto} a_3b_3a_2b_1b_2 \overset{Lemma6}{\longmapsto} a_3b_3b_1a_2b_2 \overset{Lemma5}{\longmapsto} c_3b_1c_2$

Pattern 56:
$a_3a_2b_3b_1 \overset{Lemma6}{\longmapsto} a_3b_3a_2b_1 \overset{Lemma6}{\longmapsto} a_3b_3b_1a_2 \overset{Lemma5}{\longmapsto} c_3b_1a_2$

Pattern 55:
$a_3a_2b_3b_2 \overset{Lemma6}{\longmapsto} a_3b_3a_2b_2 \overset{Lemma5}{\longmapsto} c_3c_2$

Pattern 54:
$a_3a_2b_3 \overset{Lemma6}{\longmapsto} a_3b_3a_2 \overset{Lemma5}{\longmapsto} c_3a_2$
Pattern 53:
$a_3a_2b_1b_2 \overset{Lemma6}{\longmapsto} a_3b_1a_2b_2 \overset{Lemma5}{\longmapsto} a_3b_1c_2$
Pattern 52:
$a_3a_2b_1 \overset{Lemma6}{\longmapsto} a_3b_1a_2 \overset{Lemma5}{\longmapsto} a_3b_1a_2$
Pattern 51:
$a_3a_2b_2 \overset{Lemma5}{\longmapsto} a_3c_2$
Pattern 47:
$a_3b_3b_1b_2 \overset{Lemma5}{\longmapsto} c_3b_1b_2$
Pattern 46:
$a_3b_3b_1 \overset{Lemma5}{\longmapsto} c_3b_1$
Pattern 45:
$a_3b_3b_2 \overset{Lemma5}{\longmapsto} c_3b_1$
Pattern 44:
$a_3b_3 \overset{Lemma5}{\longmapsto} c_3$
Pattern 43:
$a_3b_1b_2$
Pattern 42:
a_3b_1
Pattern 41:
a_3b_2
Pattern 37:
$a_1a_2b_3b_1b_2 \overset{Lemma6}{\longmapsto} a_1b_3a_2b_1b_2 \overset{Lemma6}{\longmapsto} b_3a_1a_2b_1b_2 \overset{Lemma6}{\longmapsto} b_3a_1b_1a_2b_2 \overset{Lemma5}{\longmapsto} b_3c_1c_2$
Pattern 36:
$a_1a_2b_3b_1 \overset{Lemma6}{\longmapsto} a_1b_3a_2b_1 \overset{Lemma6}{\longmapsto} b_3a_1a_2b_1 \overset{Lemma6}{\longmapsto} b_3a_1b_1a_2 \overset{Lemma5}{\longmapsto} b_3c_1a_2$
Pattern 35:
$a_1a_2b_3b_2 \overset{Lemma6}{\longmapsto} a_1b_3a_2b_2 \overset{Lemma6}{\longmapsto} b_3a_1a_2b_2 \overset{Lemma5}{\longmapsto} b_3a_1c_2$
Pattern 34:
$a_1a_2b_3 \overset{Lemma6}{\longmapsto} a_1b_3a_2 \overset{Lemma6}{\longmapsto} b_3a_1a_2 \overset{Lemma5}{\longmapsto} b_3a_1a_2$
Pattern 33:
$a_1a_2b_1b_2 \overset{Lemma6}{\longmapsto} a_1b_1a_2b_2 \overset{Lemma5}{\longmapsto} c_1c_2$
Pattern 32:
$a_1a_2b_1 \overset{Lemma6}{\longmapsto} a_1b_1a_2 \overset{Lemma5}{\longmapsto} c_1a_2$
Pattern 31:
$a_1a_2b_2 \overset{Lemma5}{\longmapsto} a_1c_2$
Pattern 27:
$a_1b_3b_1b_2 \overset{Lemma6}{\longmapsto} b_3a_1b_1b_2 \overset{Lemma5}{\longmapsto} b_3c_1b_2$
Pattern 26:
$a_1b_3b_1 \overset{Lemma6}{\longmapsto} b_3a_1b_1 \overset{Lemma5}{\longmapsto} b_3c_1$
Pattern 25:
$a_1b_3b_2 \overset{Lemma6}{\longmapsto} b_3a_1b_2$
Pattern 24:
$a_1b_3 \overset{Lemma6}{\longmapsto} b_3a_1$
Pattern 23:
$a_1b_1b_2 \overset{Lemma5}{\longmapsto} c_1b_2$
Pattern 22:
$a_1b_1 \overset{Lemma5}{\longmapsto} c_1$

Pattern 21:
a_1b_2
Pattern 17:
$a_2b_3b_1b_2 \overset{Lemma6}{\longmapsto} b_3a_2b_1b_2 \overset{Lemma6}{\longmapsto} b_3b_1a_2b_2 \overset{Lemma5}{\longmapsto} b_3b_1c_2$
Pattern 16:
$a_2b_3b_1 \overset{Lemma6}{\longmapsto} b_3a_2b_1 \overset{Lemma6}{\longmapsto} b_3b_1a_2$
Pattern 15:
$a_2b_3b_2 \overset{Lemma6}{\longmapsto} b_3a_2b_2 \overset{Lemma5}{\longmapsto} b_3c_2$
Pattern 14:
$a_2b_3 \overset{Lemma6}{\longmapsto} b_3a_2$
Pattern 13:
$a_2b_1b_2 \overset{Lemma6}{\longmapsto} b_1a_2b_2 \overset{Lemma5}{\longmapsto} b_1c_2$
Pattern 12:
$a_2b_1 \overset{Lemma6}{\longmapsto} b_1a_2$
Pattern 11:
$a_2b_2 \overset{Lemma5}{\longmapsto} c_2$

■

Proof of the Merging Lemma for Functional Dependencies (Lemma 10)

We first show that two successive applications of *FD*2 can be reduced to a single application of *FD*2. Suppose two successive applications of *FD*2 are as follows:

$$\frac{W_1 \subseteq V_1 \qquad \dfrac{W \subseteq V \qquad \dfrac{t'}{X \to Y}}{XV \to YW}\ FD2}{\dfrac{XVV_1 \to YWW_1}{t''}}\ FD2$$

This can be replaced by the following proof fragment, which has only one application of *FD*2.

$$\frac{WW_1 \subseteq VV_1 \qquad \dfrac{t'}{X \to Y}}{\dfrac{XVV_1 \to YWW_1}{t''}}\ FD2$$

Then, by induction, the general result follows. ■

Proof of the Interchange Lemma for Functional Dependencies (Lemma 11)

Suppose the following proof fragment is an application of *FD*3, *FD*2.

$$\frac{W \subseteq V \qquad \dfrac{\dfrac{t_1}{X \to Y} \qquad \dfrac{t_2}{Y \to Z}}{X \to Z}\ FD3}{\dfrac{XV \to ZW}{t''}}\ FD2$$

The it can be replaced by the following proof fragment, in which the proof rules appear in the reverse order *FD*2, *FD*3.

$$\cfrac{\cfrac{W \subseteq V \qquad \cfrac{t_1}{X \to Y}}{XV \to YW}\ FD2 \qquad \cfrac{W \subseteq W \qquad \cfrac{t_2}{Y \to Z}}{YW \subseteq WZ}\ FD2}{\cfrac{XV \to WZ}{t''}}\ FD3$$

■

Proof of Structural Property of Transitive Envelopes (Theorem 3):

The proof is by induction on the structure of the proof tree, T.

For the base case, if the last proof rule applied is *FD*2, then T is null. If the last proof rule is *FD*3, and the rules that are applied at levels immediately higher are not *FD*3, then T is of the following form.

$$\cfrac{\cfrac{t_1}{X \to Y} \qquad \cfrac{t_2}{Y \to Z}}{X \to Z}\ FD3$$

Then T is $(X \to Y), (Y \to Z)$, which satisfies Theorem 3.

Now, for the inductive case assume that proof trees above t_1 and t_2 have transitive envelopes $(A_1 \to A_2), \ldots, (A_{n-1} \to A_n)$ and $(A_{n+1} \to A_{n+2}), \ldots, (A_{m-1} \to A_m)$. By the inductive hypothesis, A_1 is X, A_n is Y, A_{n+1} is Y and A_m is Z. Hence $(A_1 \to A_2), \ldots, (A_{m-1} \to A_m)$ is the transitive envelope of t. ■

Proof of Repetition Removal from Transitive Envelopes (Theorem 4):

Suppose the chain $(A_1 \to A_2), \ldots, (A_n \to A_{n+1})$ is a transitive envelope of a proof t of $(A_1 \to A_{n+1})$ and $(X \to Y)$ appears in $(A_1 \to A_2), \ldots, (A_n \to A_{n+1})$ more than once. The aim is to get a proof t' of $(A_1 \to A_{n+1})$ using of the same set of assumptions (possibly a subset thereof) as that of t, but without repeated occurrences of $(X \to Y)$ in the transitive envelope of t'. Also suppose that t_i is the sub-proof tree that has $(A_i \to A_{i+1})$ as its root (conclusion) in the proof tree t.

Suppose the first and last occurrences of $(X \to Y)$ in the transitive envelope $(A_1 \to A_2), \ldots, (A_n \to A_{n+1})$ are respectively $(A_a \to A_{a+1})$ and $(A_b \to A_{b+1})$.

Case 1: $(a > 1$ and $b < n-1)$
Then, there is a FD proof t' of $(A_1 \to A_{n+1})$ that uses only *FD*3 as proof rules and $(A_1 \to A_2), \ldots, (A_{a-1} \to A_a), (A_{b+1} \to A_{b+2}) \ldots (A_n \to A_{n+1})$ as assumptions. Then the following is a proof of $(A_1 \to A_{n+1})$ from the same set of assumptions (possibly less).

$$\cfrac{\cfrac{\cfrac{t_1}{A_1 \to A_2}}{\cdots} \quad \cdots \quad \cfrac{\cfrac{t_{a-1}}{A_{a-1} \to X}}{\cdots} \quad \cfrac{\cfrac{t_b}{X \to Y}}{\cdots} \quad \cfrac{\cfrac{t_{b+1}}{Y \to A_{b+2}}}{\cdots} \quad \cdots \quad \cfrac{\cfrac{t_n}{A_n \to A_{n+1}}}{\cdots}}{\cfrac{t'}{A_1 \to A_{n+1}}}$$

Case 2: $(a = 1$ and $b < n)$
Following is a proof of $(A_1 \to A_{n+1})$ form the same set of assumptions (possibly less).

$$
\begin{array}{cccc}
\dfrac{t_b}{\dfrac{X \to Y}{\cdots}} & \dfrac{\dfrac{\dfrac{t_{b+1}}{Y \to A_{b+2}}}{\cdots}}{\dfrac{t'}{A_1 \to A_{n+1}}} & \begin{array}{c}\cdots\\ \cdots\\ \cdots\end{array} & \dfrac{t_n}{\dfrac{A_n \to A_{n+1}}{\cdots}}
\end{array}
$$

Case 3: $(a > 1$ **and** $b = n)$
Then the following is a proof of $(A_1 \to A_{n+1})$ form the same set of assumptions (possibly less).

$$
\dfrac{\begin{array}{ccc}
\dfrac{t_1}{\dfrac{A_1 \to A_2}{\cdots}} & \begin{array}{c}\cdots\\ \cdots\\ \cdots\end{array} & \dfrac{t_a}{\dfrac{X \to Y}{\cdots}}
\end{array}}{\dfrac{t'}{A_1 \to A_{n+1}}}
$$

Case 4: $(a = 1$ **and** $b = n)$
Then $(A_1 \to A_{n+1})$ is the proof of itself, as A_1 is X and A_{n+1} is Y. ■

Proof of Lemma 12 :
(1)
Following proof in semi-normal form suffices.

$$
\dfrac{\dfrac{\dfrac{Y \subseteq Y \qquad Y \to A}{Y \to YA} \qquad \dfrac{Y \subseteq Y \qquad A \to XV}{YA \to XVY}}{Y \to XVY} \qquad \dfrac{\dfrac{W \subseteq V}{YW \subseteq XVY}}{XVY \to YW}}{Y \to YW}
$$

(2)
The following proof in semi-normal form suffices.

$$
\dfrac{\dfrac{\dfrac{\dfrac{W \subseteq V}{YW \subseteq YV}}{YV \to YW} \qquad \dfrac{Y \subseteq Y \qquad YW \to XA}{YW \to YXA}}{YV \to YXA} \qquad \dfrac{\dfrac{B \subseteq A}{YB \subseteq YXA}}{YXA \to YB}}{YV \to YB}
$$

■

Proof of Repetition Removal from Assumptions (Theorem 5):

Suppose that t is a proof of $(E \to F)$ in FD and that the assumption $(X \to Y)$ is used more than once in t. Let t' be the semi-normal form of t. Notice that the set of assumptions used in t' is a subset of the set of assumptions used in t. Suppose that $(X \to Y)$ appears more than once in t'.

If all multiple uses of $(X \to Y)$ as assumptions occur in the transitive envelope of t', then by Lemma 4, there is a proof t'' that does not repeatedly use $(X \to Y)$ as an assumption. Hence, we need to prove that such repeated occurrences can be eliminated only when not all of them occur in the transitive envelope. To prove so, we consider pairs of such duplicates, where not both of them occur in the transitive envelope, and reduce the proof so that the reduced proof contains only one occurrence, instead of two of them. In order to do so, let T', TE' be the ordered listing of the assumptions and the transitive envelope of t' respectively.

Case 1: Suppose $(X \to Y)$ occurs successively in T', once as an antecedent to $FD2$ and next as an antecedent to $FD3$ (hence this occurrence is included in the transitive envelope) in that order. Hence there is some subsequence $(X \to Y), (Y \to A_1), (A_1 \to A_2) \dots (A_{n-1} \to A_n), (A_n \to XV), (XV \to YW)$ in TE' where the dependency $(XV \to YW)$ is a consequent of the rule $FD2$ applied to some $W \subseteq V$ and $(X \to Y)$, as shown below.

$$\dfrac{\dfrac{t_x}{X \to Y} \quad \dfrac{t_y}{Y \to A_1} \;\cdots\; \dfrac{t_i}{A_i \to A_{i+1}} \;\cdots\; \dfrac{t_n}{A_n \to XV} \quad \dfrac{W \subseteq V \qquad X \to Y}{XV \to YW}}{\dfrac{\cdots}{E \to F}}$$

Let the subsequence $(Y \to A_1), (A_1 \to A_2), \dots (A_{n-1} \to A_n), (A_n \to XV), (XV \to YW)$ in TE' be denoted by SUB. Also, let $t_y, t_1, \dots, t_n, t''$ be the subproofs of t' that have these dependencies as conclusions. Then $(A_1 \to A_2), \dots (A_{n-1} \to A_n), (A_n \to XV)$ is a chain, and thus by Lemma 13, there is a proof t_T of $(A_1 \to XV)$ from $t_1, \dots t_n$. By Lemma 12 (1), there is a proof, say t_S of $(Y \to YW)$ from $(Y \to A_1), (A_1 \to XV)$ and $W \subseteq V$, that uses t_T. Notice that the proof t_T does not use $(X \to Y)$ as an assumption. Now suppose $TE' = TE_1, SUB, TE_2$. Then $TE'' = TE_1, (Y \to YW), TE_2$ is a chain of dependencies. Therefore, by Lemma 13 there is a proof, say t_{final} of $(E \to F)$ from dependencies in TE'', using proof fragments of t' that have dependencies of TE_1, TE_2 as conclusions and t_T. Notice that since t_T does not use $(X \to Y)$ as an assumption, t_{final} use one less instance of $(X \to Y)$ as an assumption than t'. The structure is shown below.

$$\underbrace{\overbrace{\cdots}^{TE_1}\overbrace{(Y \to A_1)\ (A_1 \to A_2) \dots (A \to XV)\ (XV \to YW)}^{SUB}\overbrace{\cdots}^{TE_2}}_{TE}$$

$$\underbrace{\overbrace{\cdots}^{TE_1}\overbrace{(Y \to YW)}^{t_S}\overbrace{\cdots}^{TE_2}}_{TE''}$$

Case 2: Suppose $(X \to Y)$ occurs successively in T', once as an antecedent to $FD3$ (and thus is included in the transitive envelope) and again as an antecedent to $FD2$ in that order. Then there is some subsequence $(XV \to YW), (YW \to A_1), (A_1 \to A_2), \dots, (A_{n-1} \to A_n), (A_n \to X), (X \to Y)$, say SUB in TE', where the dependency $(XV \to YW)$ is a consequence of applying $FD2$ to some $W \subseteq V$ and $(X \to Y)$, as shown below.

$$\dfrac{\dfrac{W \subseteq V \qquad X \to Y}{XV \to YW} \quad \dfrac{YW \to A_1}{\cdots} \quad \dfrac{A_1 \to A_2}{\cdots} \;\cdots\; \dfrac{A_n \to X}{\cdots} \quad \dfrac{X \to Y}{}}{E \to F}$$

Then $(XV \to X)$ is derivable by $FD1$, because $X \subseteq XV$. Suppose $TE' = TE_1, SUB, TE_2$. Then $TE_1, (XV \to X), (X \to Y), TE_2$, is a chain. The proof fragments that prove dependencies in TE_1, TE_2 are subproofs in t'. By Lemma 13, we get a proof, say t_{final} of $(E \to F)$ that has one less occurrence of $(X \to Y)$ as an assumption, because proofs fragments that had conclusions in SUB used $(X \to Y)$ as an assumption, which is not there in t_{final}.

Case 3: Suppose $(X \rightarrow Y)$ occurs successively in t', both as antecedents to $FD2$. Then there is a subsequence, $(XV \rightarrow YW), (YW \rightarrow A_1), (A_1 \rightarrow A_2) \ldots (A_{n-1} \rightarrow A_n), (A_n \rightarrow XA), (XA \rightarrow YB)$ in TE', say SUB, where $(XV \rightarrow YW)$ is a consequence of applying $FD2$ to some $W \subseteq V$ and $(X \rightarrow Y)$ and $(XA \rightarrow YB)$ is a consequence of applying $FD2$ to some $B \subseteq A$ and $(X \rightarrow Y)$, as follows.

$$\frac{W \subseteq V \quad X \rightarrow Y}{XV \rightarrow YW} \qquad \cfrac{\cfrac{t_1}{YW \rightarrow A_1}}{\ldots} \qquad \cfrac{\cfrac{\cfrac{t_2}{A_1 \rightarrow A_2}}{\ldots}}{E \rightarrow F} \;\ldots\; \cfrac{\cfrac{t_n}{A_n \rightarrow XA}}{\ldots} \qquad \frac{X \rightarrow Y}{XA \rightarrow YB}$$

Then, by Lemma 13, there is a proof t_T of $(YW \rightarrow XA)$ from $(YW \rightarrow A_1), (A_1 \rightarrow A_2), \ldots, (A_{n-1} \rightarrow A_n), (A_n \rightarrow XA)$ that uses the same proof fragments $t_1, \ldots t_n$.

Then by Lemma 12 (2), there is a proof, say t_T of $(YV \rightarrow YB)$ from $(YW \rightarrow XA), W \subseteq V$ and $B \subseteq A$. Hence the following, say t_V, is a proof of $(XV \rightarrow YB)$ using $(X \rightarrow Y)$ as an assumption only once.

$$\cfrac{\cfrac{V \subseteq V \qquad X \rightarrow Y}{XV \rightarrow YV} \qquad \cfrac{t_T}{YV \rightarrow YB}}{XV \rightarrow YB}$$

$TE'' = TE_1, (XV \rightarrow YB), TE_2$ is a chain of dependencies. Hence by Lemma 13, there is a proof, say t_{final} of $(E \rightarrow F)$ from the subproofs of t' that have dependencies in TE_1, TE_2, and t_V as assumptions. Notice that t_{final} has one less occurrence of $(X \rightarrow Y)$ than t' because t_T used it as an assumption only once. The situation is as follows.

$$\underbrace{\overbrace{\ldots}^{TE_1} \overbrace{(XV \rightarrow YW)\ (YW \rightarrow A_1)(A_2 \rightarrow A_2) \ldots (A_n \rightarrow XA)\ (XA \rightarrow YB)}^{SUB} \overbrace{\ldots}^{TE_2}}_{TE}$$

$$\underbrace{\overbrace{\ldots}^{TE_1} \overbrace{(XV \rightarrow YB)}^{t_V} \overbrace{\ldots}^{TE_2}}_{TE''}$$

■

Proof of the Theorem on Properties of Inverse Proofs (Theorem 6):

Case 1: t Proves the functional dependency $(X \rightarrow Y)$

The proof is by induction. For the base case where t is $(X \rightarrow Y)$, t^{-1} is $(X \not\rightarrow Y)$. In case t is a single application of $FD2$ or $FD3$, the result follows from the proof of Lemma 14.

In case t consists of more than one application of a rule, let the proof fragment which corresponds to the application of the first proof rule be f, and its conclusion be $(A \rightarrow B)$, and, say $(P \rightarrow Q)$ is at the top of γ. Then $(A \rightarrow B)$ is the head of γ' and t' proves $(X \rightarrow Y)$. By the inductive hypotheses, $t'^{-1}(\gamma')$ proves $(A \not\rightarrow B)$ where $(X \not\rightarrow Y)$ is at the top of its independency thread. By Lemma 14 the inverse of f, say f^{-1} has $(A \not\rightarrow B)$ as the head of the independency thread and $(P \not\rightarrow Q)$ as the consequent. Hence consequent of f^{-1} can be fused to the head of the independency thread of $t'^{-1}(\gamma')$, to derive the desired result $t^{-1}(\gamma)$.

Case 2: t proves functional independency $(X \not\to Y)$

Once again, the proof is by induction. For the base case where t is $(X \not\to Y)$, t^{-1} is $(X \to Y)$. In case where the result where t is a single application of $FI1$, $FI2$ or $FD3$, the result follows from the proof of Lemma 14.

For the inductive case, the proof is similar to Case 1, except that we use the unique independency thread. ■

Proof of the Inconsistency Test (Lemma 15):

If $\Sigma \vdash (X \to Y)$ for some independency $(X \not\to Y) \in \Sigma'$, then by Definition 16, $\Sigma \cup \Sigma'$ is inconsistent.

To prove the converse, suppose $\Sigma \cup \Sigma'$ is inconsistent. Then by definition 16, there are attribute sets P and Q such that $\Sigma \cup \Sigma' \vdash (P \not\to Q), (P \to Q)$. Then, by the normal form theorem, there is a normal proof t of $(P \not\to Q)$ from $\Sigma \cup \Sigma'$. In the following case analysis we show that this always leads to the desired result.

Case 1: Suppose all rules $FI3$, $FI1$, and $FI2$, are applied in the independency thread of t. Then the proof is of the following form.

$$\frac{\dfrac{\Sigma}{X \to P} \qquad \dfrac{\dfrac{\Sigma}{X \to A} \qquad \dfrac{\dfrac{\Sigma}{AQ \to Y} \qquad X \not\to Y}{X \not\to AQ}\ FI3}{X \not\to Q}\ FI1}{P \not\to Q}\ FI2$$

Hence we get $\Sigma \vdash (X \to P)$, $(P \to Q)$, $(X \to A)$, $(AQ \to Y)$. Consequently, we get $\Sigma \vdash (X \to Y)$ from Armstrong's axioms, where $(X \not\to Y) \in \Sigma'$.

Case 2: Suppose the application of independency rules were restricted to only $FI3$ and $FI1$. Then only the first two proof rules are relevant. Therefore we get that P is X, and therefore $\Sigma \vdash (AQ \to Y)$, $(X \to A)$, $(X \to Q)$. Consequently, we get $\Sigma \cup \Sigma' \vdash (X \to Y)$, for $(X \not\to Y) \in \Sigma'$.

Case 3: Suppose the application of independency rules are $FI3$, $FI2$. Then the proof is of the following form.

$$\frac{\dfrac{\Sigma}{X \to P} \qquad \dfrac{\dfrac{\Sigma}{Q \to Y} \qquad X \not\to Y}{X \not\to Q}\ FI3}{P \not\to Q}\ FI2$$

Hence we get $\Sigma \vdash (X \to P)$, $(P \to Q)$, $(Q \to Y)$. Consequently, we get $\Sigma \vdash (X \to Y)$ from Armstrong's axioms, for $(X \not\to Y) \in \Sigma'$.

Case 4: Suppose the application of independency rules are $FI1$, $FI2$. Then the proof of $(P \not\to Q)$ is of the form given below. In the proof we see that Y must be of the form TQ for some attribute set T.

$$\frac{\dfrac{\Sigma}{X \to P} \qquad \dfrac{\dfrac{\Sigma}{X \to T} \qquad X \not\to TQ}{X \not\to Q}\ FI1}{P \not\to Q}\ FI2$$

Hence we get $\Sigma \vdash (X \to P),\ (P \to Q),\ (X \to T))$. Consequently, we get $\Sigma \vdash (X \to TQ)$ from Armstrong's axioms, for $(X \not\to TQ) \in \Sigma'$.

Case 5: Suspose the only independency rule applied is *FI3*.
Then only the first application of the proof rule in case 1 is relevant and we get that P is X, AQ is Y, and $\Sigma \vdash (QU \to Y),\ (X \to Q)$. Consequently by applying Armstrong's axioms we get $\Sigma \cup \Sigma' \vdash (X \to Y)$, for $(X \not\to Y) \in \Sigma'$.

Case 6: Suppose the only independency rule applied is *FI1*.
Then only the first two lines of the proof of case 4 are relevant. Then P is X and Y is TQ. Hence we get that $\Sigma \vdash (P \to T),\ (P \to Q)$, and by Armstrong's axioms we get $\Sigma \vdash (P \to TQ)$, i.e. $\Sigma \vdash (X \to Y)$, for $(X \not\to Y) \in \Sigma'$.
Case 7: Suppose the only independency rule applied is *FI2*.
Then the proof of $(P \not\to Q)$ is of the following form.

$$\frac{X \to P \qquad\qquad X \not\to Q}{P \not\to Q}\quad FI2$$

Then, we get that Y is Q and that $\Sigma \vdash (X \to P),\ (P \to Q)$, and by Armstrong's axioms get $\Sigma \vdash (X \to Y)$. ■

Proof of Consistency when adding a dependency (Lemma 16):

Suppose $\Sigma \cup \Sigma' \cup \{X \to Y\}$ is not consistent. Then by Lemma 15, there is an independency $(P \not\to Q) \in \Sigma'$ such that $\Sigma \cup \{X \to Y\} \vdash (P \to Q)$. Because $\Sigma \cup \Sigma'$ is consistent, the proof of $(P \to Q)$ must use $(X \to Y)$ as an assumption. Then, by Theorem 5, there is a proof of $(P \to Q)$ from assumptions $\Sigma \cup \{X \to Y\}$ that uses $(X \to Y)$ only once as an assumption. Call this proof t. Let γ be the proof thread of t that begins at the assumption $(X \to Y)$. Then, by Theorem 6, $t^{-1}(\gamma)$ is a proof of $(X \not\to Y)$ from the assumptions $\Sigma \cup \{P \not\to Q\}$. Because $(P \not\to Q) \in \Sigma'$, $\Sigma \cup \Sigma' \vdash (X \not\to Y)$, contradicting the hypotheses of the Lemma. ■

Proof of Consistency when adding an independency (Lemma 17):

Suppose $\Sigma \cup \Sigma' \not\vdash (X \to Y)$ and $\Sigma \cup \Sigma' \cup \{X \not\to Y\}$ is inconsistent. Then there a dependency $(P \to Q)$ such that $\Sigma \vdash (P \to Q)$ with $\Sigma \cup \Sigma' \cup \{X \not\to Y\} \vdash (P \not\to Q)$. By the normal form theorem (i.e. Theorem 1), there is a normal proof t of $(P \not\to Q)$ from $\Sigma \cup \Sigma' \cup \{X \not\to Y\}$. Notice that a normal proof applies independence rules in the order *FI3*, *FI1*, *FI2*.

Case 1: Suppose all rules *FI3*, *FI2*, *FI1* are applied in the independency thread of t.
Then the proof is of the following form.

$$\cfrac{\cfrac{\Sigma}{X \to P} \qquad \cfrac{\cfrac{\Sigma}{X \to A} \qquad \cfrac{\cfrac{\Sigma}{AQ \to Y} \qquad X \not\to Y}{X \not\to AQ}\ FI3}{X \not\to Q}\ FI1}{P \not\to Q}\ FI2$$

Hence we get $\Sigma \vdash (X \to P),\ (P \to Q),\ (X \to A),\ (AQ \to Y)$. Consequently, we get $\Sigma \vdash (X \to Y)$ from Armstrong's axioms, contradicting $\Sigma \cup \Sigma' \not\vdash (X \to Y)$.

Case 2: Suppose the application of independency rules are *FI3* and *FI1* in that order. Then only the first three lines of the above proof are relevant, and hence we get that P is X, and therefore $\Sigma \vdash (QA \rightarrow Y),\ (X \rightarrow A),\ (X \rightarrow Q)$. Consequently, we get $\Sigma \cup \Sigma' \vdash (X \rightarrow Y)$ for a contradiction.

Case 3: Suppose the application of independency rules are *FI3*, *FI2* in that order. Then the proof is as follows.

$$\dfrac{\dfrac{\Sigma}{X \rightarrow P} \qquad \dfrac{\dfrac{\Sigma}{Q \rightarrow Y} \qquad X \nrightarrow Y}{X \nrightarrow Q}\ FI3}{P \nrightarrow Q}\ FI2$$

Hence we get $\Sigma \vdash (X \rightarrow P),\ (P \rightarrow Q),\ (Q \rightarrow Y)$. Consequently, we get $\Sigma \vdash (X \rightarrow Y)$ from Armstrong's axioms, contradicting $\Sigma \cup \Sigma' \nvdash (X \rightarrow Y)$.

Case 4: Suppose the application of independency rules are *FI1*, *FI2* in that order. Then the proof of $(P \nrightarrow Q)$ is of the form given below. In the proof we see that Y must be of the form TQ for some attribute set T.

$$\dfrac{\dfrac{\Sigma}{X \rightarrow P} \qquad \dfrac{\dfrac{\Sigma}{X \rightarrow T} \qquad X \nrightarrow TQ}{X \nrightarrow Q}\ FI1}{P \nrightarrow Q}\ FI2$$

Hence we get $\Sigma \vdash (X \rightarrow P),\ (P \rightarrow Q),\ (X \rightarrow T))$. Consequently, we get $\Sigma \vdash (X \rightarrow TQ)$ from Armstrong's axioms, contradicting $\Sigma \cup \Sigma' \nvdash (X \rightarrow Y)$.

Case 5: Suppose the only independency rule applied is *FI3*. Then only the first two lines of the proof in case 1 are relevant and we get that P is X and $\Sigma \vdash (QA \rightarrow Y),\ (X \rightarrow Q)$. Consequently by applying Armstrong's axioms we get $\Sigma \cup \Sigma' \vdash (X \rightarrow Y)$ for a contradiction.

Case 6: Suppose the only independency rule applied is *FI1*. Then only the first two lines of the proof of case 4 are relevant. Then P is X and Y is TQ. Hence we get that $\Sigma \vdash (P \rightarrow T),\ (P \rightarrow Q)$, and by Armstrong's axioms get $\Sigma \vdash (P \rightarrow TQ)$, i.e. $\Sigma \vdash (X \rightarrow Y)$ for a contradiction.

Case 7: Suppose the only independency rule applied is *FI2*. Then the proof of $(P \nrightarrow Q)$ is of the following form.

$$\dfrac{X \rightarrow P \qquad X \nrightarrow Q}{P \nrightarrow Q}\ FI2$$

Then, we get that Y is Q and that $\Sigma \vdash (X \rightarrow P),\ (P \rightarrow Q)$, and by Armstrong's axioms we get $\Sigma \vdash (X \rightarrow Y)$ for a contradiction. ■

References

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining Association Rules Between Sets of Items In Large Databases. In *ACM SIGMOD Conference on Management of Data*, volume 1, pages 207–216. SIGMOD, 1993.

[Arm74] W. W. Armstrong. Dependency Structure of Database Relationships. In *Proc. IFIP 74*, pages 580–583. North Holland, Amsterdam, 1974.

[AS95] Rakesh Agrawal and Ramakrishnan Srikant. Mining Generalized Association Rules. In *Proc. of the 21st VLDB Conference*, Zurich, Switzerland, 1995.

[BBG78] C. Beeri, P.A. Bernstein, and N. Goodman. A Sophisticate's Introduction to Database Normalisation Theory. *Proc. of the Fourth Int'l. Conf. on Very Large Data Bases*, pages 113–124, 1978.

[BDFS84] Cartriel Beeri, Martin Dowd, Ronald Fagin, and Richard Statman. On The Structure of Armstrong Relations for Functional Dependencies. *Journal of the ACM*, 31(1):30–46, January 1984.

[Bel95a] Siegfired Bell. Discovery and Maintainence of Functional Dependencies. In *Proc. of the First KDD Conference*, pages 27–32, Montréal, Canada, 1995.

[Bel95b] Siegfred Bell. The Expanded Implication Problem of Data Dependencies. Technical Report LS-8 Report 16, Dept of Computer Science, University of Dortmund, 1995.

[Bel96] Siegfred Bell. Deciding distinctness of query results by discovered constraints. Technical Report To Be Published, Dept of Computer Science, University of Dortmund, 1996.

[BMR85] Dina Bitton, Heikki Mannila, and Kari-Jouko Räihä. Design-By-Example: A design Tool for Relational Databases. Technical Report TR 85-692, Dept of Computer Science, Cornell University, 1985.

[BMT89] Dina Bitton, Jeffrey Millman, and Solveig Torgersen. A Feasibility and Performance Study of Dependency Inference. In *Proc. of the 5th Int'l Conf. on Data Engg.*, pages 635–641, Los Angeles, CA, 1989.

[CAdS84] M. A. Casanova and J. E. Amarald de Sa. Mapping uninterpreted schemes into entity-relationship diagrams. *IBM Journal of Research and Development*, 28-1:82–94, 1984.

[Cod70] E. F. Codd. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13:377–387, 1970.

[Dec87] R. Dechter. Decomposing an n-ary relation into a tree of binary relations. In *Proceedings of the Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, volume 1. ACM-SIGMOD, March 1987.

[EN94] Ramez A. Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems.* Benjamin Cummings, Redwood City, CA, 2nd edition, 1994.

[Fit83] Melvin Fitting. *Proof Methods for Modal and Intuitionistic Logic.* Riedel, Amsterdam, 1983.

[FV86] Ronald Fagin and Moshe Vardi. The theory of data dependencies - a survey. *Symposia in Applied Mathematics*, 34:19–72, 1986.

[GJS+96] M Ganesh, Namit Jain, Jaideep Srivastava, Sujal Parikh, Travis Richardson, and Duminda Wijesekera. Efficient mining for probabilistic functional dependencies. Technical Report In Preparation, Dept of Computer Science, University of Minnesota, 1996.

[GL90] G Gottlib and L Libkin. Investigations on armstrong relations , dependency inference, and excluded functional dependencies. *Acta Cybernetica*, 9:385–402, 1990.

[Gra95] J. N. Gray. Database systems: A textbook case of research paying off. In *http://cra.org/research.impact*, 1995.

[Jan88] J. M. Janas. Covers for Functional Independencies. In *Conference on Database Theory*, volume 338. Springer-Verlag, Lecture Notes in Computer Science, 1988.

[Kan90] Paris C. Kanellakis. *Elements of relational database Theory*, chapter 17, pages 1074–1156. The MIT Press and Elsevier, Amsterdam, 1990.

[Mai83] David Maier. *The Theory of Relational Databases.* Computer Science Press, Rockville, Maryland, 1983.

[MR87] Heikki Mannila and Kari-Jouko Räihä. Dependency inference. In *Proc. of 13th VLDB Conference*, pages 155–158, Brighton, UK, 1987.

[MR92a] Heikki Mannila and Kari-Jouko Räihä. *The Design of Relational Databases.* Addison-Wesley, Workingham, U.K., 1992.

[MR92b] Heikki Mannila and Kari-Jouko Räihä. On the complexity of inferring functional dependencies. *Discrete Applied Mathematics*, 40:237–243, 1992.

[MR94] Heikki Mannila and Kari-Jouko Räihä. Algorithms for inferring functional dependencies from relations. *Data and Knowledge Engineering*, 12:83–99, 1994.

[PK95] Bernhard Pfahringer and Stefan Kramer. Compression-Based Evaluation of Partial Determinations. In *Proc. of the First KDD Conference*, pages 234–239, Montréal, Canada, 1995.

[Pra65] Dag Prawitz. *Natural Deduction.* Almqvist and Wiksell, Stockholm, 1965.

[PS92] Gregory Piatetsky-Shapiro. Probabilistic data dependencies. In *Machine Learning Workshop*, volume 1, 1992.

[SF96] Iztok Savnik and Peter. A. Falch. Bottom-up induction of functional dependencies from relations. In *PDIS*, volume 1. PDIS, 1996.

[Tak91] Gaisie Takeuti. *Proof Theory.* North Holland, Amsterdam, 2 edition, 1991.

[Tha88] B. Thalheim. Logical relational detabase design tools using different classes of dependencies. *Journal of New Genaration Computing Systems*, 13(3):211–228, 1988.

[Ull88] Jeffrey D. Ullman. *Database and Knowledge-Base Systems.* Computer Science press, Rockville, Maryland, 1988.

[Var88] Moshe Vardi. Fundamentals of Dependency Theory. pages 171–224. Computer Science Press, 1988.

[WGSN97] Duminda Wijesekera, M. Ganesh, Jaideep Srivastava, and Anil Nerode. Tableaux for Functional Dependencies and Independencies. In *Tableaux'97*, 1997.

[Yan95] Mihalis Yannakakis. Perspectives on Database Theory. In *Foundations of Computing*, pages 224–246. IEEE FOCS, 1995.