

AD-A045 232

CALIFORNIA UNIV SANTA CRUZ INFORMATION SCIENCES
FORMULAS FOR GENERATING PLANS. (U)

F/G 12/2

UNCLASSIFIED

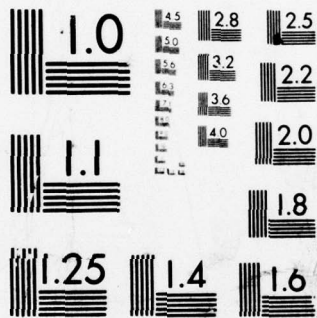
APR 77 S SICKEL
TR-77-8-002

N00014-76-C-0681
NL

| OF |
AD
AO45 232



END
DATE
FILMED
11-77
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA 045232

12

FORMULAS FOR GENERATING PLANS

by

Sharon Sickel

Technical Report No. 77-8-002 ✓

1977

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) TR-778-002	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) FORMULAS FOR GENERATING PLANS.	5. TYPE OF REPORT & PERIOD COVERED (9) Technical rept.	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Sharon/Sickel	8. CONTRACT OR GRANT NUMBER(s) (15) N00014-76-C-0681 new	9. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
9. PERFORMING ORGANIZATION NAME AND ADDRESS Information Sciences University of California Santa Cruz, California 95064	10. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia 22217	12. REPORT DATE (11) Apr 1977
11. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research University of California 553 Evans Hall Berkeley, California 94720	13. NUMBER OF PAGES 2 (125p)	15. SECURITY CLASS. (of this report) Unclassified
14. DISTRIBUTION STATEMENT (of this Report) Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) predicate calculus context-free grammars algorithms		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A closed form is discussed that provides an interface between formal specification of problems and the algorithms that solve them.		

D D C
RECEIVED
OCT 17 1977
LIBRARY
C

410350

James

Formulas for Generating Plans

Sharon Sickel
Information Sciences
University of California

Santa Cruz,
California

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
DI	SPECIAL
A	

It has been shown that any problem expressible as a theorem in the predicate calculus can be represented by a context-free attribute grammar such that the language of the grammar represents the plans that solve the problem.* A closed form for the language is often derivable; the notation for the closed form is a regular algebra -- an extension of regular expressions. Formal language theory can be used to simplify the grammar and the corresponding language.

Theorem proving has been used in the past for question answering and to generate or verify solutions to specific (ground case) problems. Here we generate algorithms. For example, solve factorial(n) (rather than factorial(6)), or answer subset(S,T) (rather than subset({1, 2}, {0, 1, 2, 5})). The closed form derivable from the grammar mentioned above gives the control structure of the algorithm. There may be more than one closed

*Formal Grammars as Models of Logic Derivations, Sharon Sickel, submitted to IJCAI 77.

form that will accomplish the task, and we choose among them. In this way we avoid the complexity of having to describe all solutions, and instead choose one that lends itself to execution. The data manipulation of the steps of the algorithm is given by the unification of components of the problem specification. Specifically, assignment statements in the algorithm assign values to the variables that correspond to values the variables are unified with in the specification.

The closed form also provides information about certain properties of the algorithm. The domain and range of the task are derivable from the closed form by replacing terminal symbols by substitutions and performing an operation on them similar to composition. The closed form may also describe how to compute recursively defined functions iteratively by determining a priori when loops will terminate and by discovering an upper bound on the amount of information required at any one time. The original specification may inherently imply an algorithm containing redundancies. We may be able to automatically improve such algorithms. For example, if the zero function is described recursively, the implied computation is inefficient. However our analysis shows that the range consists of a single element. Therefore the algorithm for the function can be transformed to one that maps directly onto the single range element. Another example of this simplification occurs in generating plans for travel on a Manhattan grid with no barriers. To go from point $(0,0)$ to point (m,n) , we could do an arbitrarily large amount of meandering. However,

it is possible to derive a plan that will accomplish the trip in $m+n$ steps by using a less general closed form that nevertheless achieves the task for all m and n .

Some problems may be so hard that finding the closed forms directly from the grammar is not practical (or even possible). In these cases, we may be able to induce a general plan by solving the problem for a small set of elements of the domain and generalizing on those solutions. The generalization is guessed from the examples and must then be verified for the entire domain, usually by mathematical induction on the construction operator of the domain. If the domain is finite or recursively defined, this proof should be automatic.

The closed form used here provides an interface between formal specification of problems and the algorithms that solve them. The automatic generation of these forms is a step toward mechanized plan formation.

OFFICIAL DISTRIBUTION LIST

Contract N00014-76-C-0681

Defense Documentation Center
Cameron Station
Alexandria, VA 22314
12 copies

Office of Naval Research
Information Systems Program
Code 437
Arlington, VA 22217
2 copies

Office of Naval Research
Code 102IP
Arlington, VA 22217
6 copies

Office of Naval Research
Code 200
Arlington, VA 22217
1 copy

Office of Naval Research
Code 455
Arlington, VA 22217
1 copy

Office of Naval Research
Code 458
Arlington, VA 22217
1 copy

Office of Naval Research
Branch Office, Boston
495 Summer Street
Boston, MA 02210
1 copy

Office of Naval Research
Branch Office, Chicago
536 South Clark Street
Chicago, IL 60605
1 copy

Office of Naval Research
Branch Office, Pasadena
1030 East Green Street
Pasadena, CA 91106
1 copy

New York Area Office
715 Broadway - 5th Floor
New York, NY 10003
1 copy

Naval Research Laboratory
Technical Information Division
Code 2627
Washington, DC 20375
6 copies

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marina Corps (CodeRD-
Washington, D. C. 20380
1 copy

Naval Electronics Laboratory Center
Advanced Software Technology Division
Code 5200
San Diego, CA 92152
1 copy

Mr. E. H. Gleissner
Naval Ship Research & Development Cent.
Computation and Mathematics Department
Bethesda, MD 20084
1 copy

Captain Grace M. Hopper
NAICOM/MIS Planning Branch (OP-916D)
Office of Chief of Naval Operations
Washington, D. C. 20350
1 copy

Mr. Kin B. Thompson
Technical Director
Information Systems Division (OP-911G)
Office of Chief of Naval Operations
Washington, D. C. 20350
1 copy