Recursive Rules - An Outside Challenge

by

Donald P. McKay

SNeRG

Technical Note Number 1

November 13, 1980

Introduction

Recursive rules are rules or sets of rules one of whose antecedents matches at least one of its consequents. For example, if we have the set of rules $A_1 \rightarrow B_1$, $B_1 \rightarrow B_2$, ..., $B_n \rightarrow C$ where $n \geq 0$ and A_1 matches C then the set of rules can be said to be recursive. A single recursive rule is one in which one of the antecedents matches one of the consequents. The SNePS inference system can handle recursive rules or recursive rule sets as a part of the normal deductive process. See [Shapiro and McKay, 1980] for details on how the system uses recursive rules.

In the course of work on DADM at SDC [Kellogg, et al; Klahr, 1978], a deductive processor intended to be a data base frontend/extender, the researchers involved have developed a robust system which they claim handles recursive rules. However, the system handles them by considering several special cases which they claim handle most recursive rules. The method employed by SNePS is considerably different in that <u>no</u> special case processing is required - directly and indirectly recursive rules are used productively until no further new deductions can be made from them. In the course of some discussions with the SDC research group, John Olney, a contributor to the group, sent us some sample data from their domain which contains recursive rules and a small set of questions (or deductions) which they wished to see run with SNePS.

The remainder of this technical note gives the ATN generator grammar used for generating english surface descriptions of network structures and the SNePS run using the SDC data.

The ATN Generator Grammar

The ATN generator grammar uses the general connective framework with a simple extension for domain specific information. The first section generates surface expressions for the SNePS connectives and quantifiers and as such can be used in any application. The second section of the grammar handles the generation from the representation chosen for the SDC data.

```
;=============== RULEGRM 6/6/80 ======================
;========== WRITTEN BY STUART C. SHAPIRO ==============
;
;      THIS IS A GRAMMAR FOR GENERATING ENGLISH FROM SNEPS DEDUCTION
;   RULES.  TO USE IT DO THE FOLLOWING --
;
;     1.  ADD A GENERATION GRAMMAR FOR ATOMIC ASSERTIONS.
;         THIS GRAMMAR MUST START AT THE STATE 'GS', MUST BUILD A
;         SENTENCE IN THE REGISTER 'STRING', AND MUST TRANSFER TO
;         THE STATE 'END', HAVING CONSUMED ITS NODE.
;         THE REGISTER 'NEG' WILL BE SET TO 'T', IF THE SENTENCE
;         IS TO BE NEGATED.
;
;     2.  START THE PARSER AT STATE 'G', OR USE THE SNEPS FUNCTION,
;         'SURFACE'.
;
;     3.  TO TRACE INFERENCES IN ENGLISH, (SETQ INFERTRACE 'SURFACE).
```

```
;
(G (JUMP GRULE T (SETR CONJ 'AND) (SETR TAB 0)))

(G1 (GROUP (JUMP GRULE COUNT (ADDR STRING (GETR COUNT) '/))
                 (SETR COUNT (ADD1 (GETR COUNT))))
            (JUMP GRULE (NULLR COUNT))))

(GRULE(GROUP (JUMP GORENT (GETA ANT))
       (JUMP GNUMQUANT (GETA PEVB) (SETR EMAX (GETA EMAX))
                                   (SETR EMIN (GETA EMIN)))
       (JUMP G&ENT (OR (GETA CQ) (GETA DCQ)))
       (JUMP GTHRESH (GETA THRESH) (SETR TOT (LENGTH (GETA ARG)))
                                   (SETR THRESH (GETA THRESH)))
       (JUMP GANDOR (GETA MAX) (SETR TOT (LENGTH (GETA ARG)))
                    (SETR MIN (GETA MIN)) (SETR MAX (GETA MAX)))
       (JUMP GS T))
      (TO (END) T (PRIN3 <> "======ERROR -- NO GENERATION GRAMMAR FOR" <>)
                  (DESCRIBE (^ (GETR /*))))))

(GANDOR (GROUP (JUMP GANDORN NEG)
        (JUMP GANDORP (NULLR NEG))))

(GANDORN(GROUP(JUMP GANDORP-N1N (OVERLAP MIN TOT) (SETR MIN 1))
         (JUMP GANDORP-NNN (AND (OVERLAP MIN 1) (OVERLAP MAX TOT))
                           (SETR MIN TOT))
         (CALL GANDORP (CONCAT (GETA ARG) '<>) T
               (SENDR CONJ 'AND)
                (SENDR TAB (PLUS (GETR TAB) 3)) STR
                 (ADDR STRING '"IT IS NOT THE CASE THAT" '<>
                       '% (PLUS 3 (GETR TAB)) STR) (TO END))))

(GANDORP(GROUP(CALL GRULE (CONCAT (GETA ARG) '<>) (OVERLAP MAX 0)
            (SENDR CONJ 'AND)(SENDR NEG (NULLR NEG))
            (SENDR TAB) * (ADDR STRING *) (TO END))
       (JUMP GANDORP-NNN (OVERLAP MIN TOT))
       (JUMP GANDORP-N1N (AND (OVERLAP MIN 1) (OVERLAP MAX TOT)))
         (CALL GRULE (CONCAT (GETA ARG) '<>)
                     (AND (OVERLAP MIN 1) (OVERLAP MAX 1))
                     (SENDR CONJ 'OR) (SENDR TAB (PLUS 7 (GETR TAB)))
                     (SENDR NEG) STR
                     (ADDR STRING '"EITHER" STR) (TO END))
    (CALL G1 (CONCAT (GETA ARG) '<>) (OVERLAP MAX TOT) (SENDR NEG)
            (SENDR COUNT 1) (SENDR TAB (PLUS 3 (GETR TAB))) STR
         (ADDR STRING '"AT LEAST" MIN '"OF THE FOLLOWING:" '<>
          '% (PLUS 3 (GETR TAB)) STR) (TO END))
    (CALL G1 (CONCAT (GETA ARG) '<>) (OVERLAP MIN 0) (SENDR COUNT 1)
           (SENDR TAB (PLUS 3 (GETR TAB))) (SENDR NEG) STR
           (ADDR STRING '"AT MOST" MAX
'"OF THE FOLLOWING" '<>
             '% (PLUS 3 (GETR TAB)) STR) (TO END))
    (CALL G1 (CONCAT (GETA ARG) '<>) (OVERLAP MIN MAX) (SENDR COUNT 1)
          (SENDR TAB (PLUS 3 (GETR TAB))) (SENDR NEG) STR
          (ADDR STRING '"EXACTLY" MIN '"OF THE FOLLOWING:*" '<>
               '% (PLUS 3 (GETR TAB)) STR) (TO END))
    (CALL G1 (CONCAT (GETA ARG) '<>) T(SENDR COUNT 1)
       (SENDR TAB (PLUS 3 (GETR TAB))) (SENDR NEG) STR
       (ADDR STRING 'BETWEEN MIN 'AND MAX
                  '"OF THE FOLLOWING:" '<>
                 '% (PLUS 3 (GETR TAB)) STR) (TO END))))

(GANDORP-NNN (CALL GRULE (CONCAT (GETA ARG) '<>) T
          (SENDR CONJ 'AND) (SENDR NEG)
          (SENDR TAB) * (ADDR STRING *) (TO END)))
(GANDORP-N1N (CALL GRULE (CONCAT (GETA ARG) '<>)T
          (SENDR CONJ 'OR) (SENDR TAB (PLUS 7 (GETR TAB)))
                 (SENDR NEG) STR
                (ADDR STRING '"EITHER" STR '"(OR"
                                        (COND ((OVERLAP MAX 2) '"BOTH)")
                                              (T (LIST 'ALL (GETR MAX) '/)))))
                (TO END)))

(GNUMQUANT (CALL GRULE (CONCAT(GETA &ANT)(GETA CQ) '<>) (NULLR NEG)
                 (SENDR CONJ 'AND)
                (SENDR TAB (PLUS 3 (GETR TAB))) STR
                (ADDR STRING '"THERE IS AT MOST" (GETR EMAX)
                (GETA PEVB) '"SUCH THAT" '<> '% (PLUS 3(GETR TAB)) STR)
                (TO END)))
(G&ENT(GROUP(CALL GRULE (CONCAT (GETA &ANT) '<>) (GETA &ANT) (SENDR CONJ 'AND)
```

```
              (SENDR TAB (PLUS 3 (GETR TAB)))(SENDR STRING 'IF)STR
             (ADDR STRING (IF (GETR NEG)  '"IT IS NOT THE CASE THAT" '<>) STR
                   '<> '% (PLUS 3 (GETR TAB))) (JUMP GCQ))
         (CALL GRULE (CONCAT (GETA CQ) '<>) T (SENDR TAB) (SENDR CONJ 'AND)
              * (ADDR STRING *) (TO END))))
 (GORENT (CALL GRULE (CONCAT (GETA ANT) '<>) T (SENDR CONJ 'OR)
              (SENDR TAB (PLUS 3 (GETR TAB))) (SENDR STRING 'IF) STR
              (ADDR STRING (IF (GETR NEG) '"IT IS NOT THE CASE THAT" '<>) STR
                   '<> '% (PLUS 3 (GETR TAB))) (JUMP GCQ)))

(GCQ (CALL GRULE (CONCAT (GETA CQ) '<>) (GETA CQ)
          (SENDR CONJ 'AND)(SENDR STRING 'THEN)(SENDR TAB(PLUS 8(GETR TAB)))
          * (ADDR STRING *) (TO END)))

(GTHRESH
   (GROUP(JUMP GANDORP NEG (SETR MIN THRESH) (SETR MAX (SUB1 (GETR TOT))))
       (CALL GRULE (CONCAT (GETA ARG) '<>)
           (AND (OVERLAP TOT 2) (OVERLAP THRESH 1))
          (SENDR TAB (PLUS 3 (GETR TAB))) (SENDR CONJ '"IF AND ONLY IF")
           STR (ADDR STRING STR) (TO END))
       (CALL G1 (CONCAT (GETA ARG) '<>) (OVERLAP THRESH 1)
           (SENDR COUNT 1) (SENDR TAB (PLUS 3 (GETR TAB)))
           (SENDR CONJ 'AND)
         STR(ADDR STRING '"THE FOLLOWING ARE EQUIVALENT:" '<> STR) (TO END))
       (CALL G1 (CONCAT (GETA ARG) '<>) T (SENDR COUNT 1)
        (SENDR CONJ 'AND)
           (SENDR TAB (PLUS 3 (GETR TAB)))
           STR (ADDR STRING '"IF ANY OF THE FOLLOWING ARE TRUE,"
                            '"THEY ALL ARE:" '<> STR) (TO END))))

(END (GROUP (WRD <> T (TO ENDPOP))
          (JUMP G1 T (ADDR STRING CONJ '<> '% (GETR TAB)))
           (POP STRING T)))

(ENDPOP (POP STRING T))

;================== END OF RULE GRAMMAR =======================

;*********************************  GRAMMAR  *********************************
;                FOR TEST FROM SDC
;            11/7/80    DPM
;
(GS (GROUP
          (TO (END)(EQ (GETA REL) 'CONCAT)
                   (ADDR STRING (GETA A3) '"IS THE CONCATENATION OF"
                                (GETA A1) '"AND" (GETA A2)))
          (TO (END)(EQ (GETA REL) 'GOES-THROUGH)
                   (ADDR STRING (GETA A1) '"GOES THROUGH" (GETA A2)))
          (TO (END)(EQ (GETA REL) 'ROUTE-PART)
                   (ADDR STRING (GETA A1) '"IS A ROUTE PART OF"
                                (GETA A2)))
          (TO (END)(EQ (GETA REL) 'EFFICIENCY)
                   (ADDR STRING '"THE EFFICIENCY OF" (GETA A1)
                                '"IS" (GETA A2)))
          (TO (END)(EQ (GETA REL) 'SPAN)
                   (ADDR STRING '"THE SPAN OF" (GETA A1) '"IS"
                                (GETA A2)))
          (TO (END) (EQ (GETA REL) 'ISGREATER)
                  (ADDR STRING (GETA A1) '"IS GREATER THAN"
                               (GETA A2)))
          (TO (END)(EQ(GETA NAME:) 'SUM)
                (ADDR STRING (GETA  RESULT) '"IS THE SUM OF"
                      (GETA A1) '"AND" (GETA A2)))
          (TO (END)(EQ (GETA NAME:) 'GREATER)
                  (ADDR STRING '"(GREATERP" (GETA A1)(GETA A2) '")"))
          (TO (END)(EQ (GETA NAME:) 'WTDAVG)
                   (ADDR STRING '"THE WEIGHTED AVERAGE OF"
                                (GETA A2) '"AND" (GETA A4)
                                '"(USING WEIGHTS OF" (GETA A1) '"AND"
                                (GETA A3) '"RESPECTIVELY) IS" (GETA RESULT)))
        (TO (END) (EQ (GETA NAME:) 'GREATER)
                  (ADDR STRING (GETA A1) '"IS GREATER THAN" (GETA A2)))
))
```

Sample Run

This section shows the sample run of SNePS using the SDC data and the generator grammar above. The output below was generated by a run using ALISP on the CYBER 174 at SUNY/Buffalo in batch with a field length of 50K (decimal) words. (The maximum field length of an interactive job is approximately 47K (decimal) words.) All source code was run interpretted, substantial space and time improvements would be obtained using a compiled SNePS system.

```
 ALISP  VERSION 3.2    11 NOVEMBER 1980   2:15 PM
;***************************************************************************
;*
;*   Input SNePS system
;*
 $ ?$$$$$ (LENGTH(LIST(INPUT(SNEPSRC CSDLIB))(INPUT(MATCH CSDLIB))(INPUT(MULTI
CSDLIB))))
 3
 $ ?$$$$$ (LENGTH(LIST(INPUT INFER)))
 1
 $ ?$$$$$ (LENGTH(INPUT PATCH))
 2
 $ ?$$$$$ (LENGTH(INPUT NEWUPDT))
 18
 $ ?$$$$$ (SNEPS T)
 SNEPS
 $*:$$$$$ (^(LENGTH(INPUT (PARSER CSDLIB))))
 (77)
 3149 MSECS

 $*:$$$$$ (^(LENGTH(ATN-IN 'SHOWOFF)))
 (16)
 1545 MSECS

 $*:$$$$$ (^(SETQ INFERTRACE 'SURFACE))
 (SURFACE)
 5 MSECS
;***************************************************************************
;*
;*   Declare arc names to be used in run
;*
 $*:$$$$$ (DEFINE REL REL- A1 A1- A2 A2- A3 A3- A4 A4- RESULT RESULT-)
 (REL REL-)
 (A1 A1-)
 (A2 A2-)
 (A3 A3-)
 (A4 A4-)
 (RESULT RESULT-)
 (DEFINED)
 39 MSECS
;***************************************************************************
;*
;*   Assert SDC data and rules
;*
 $*:$$$$$ (SURFACE (BUILD REL CONCAT A1 R1 A2 R2 A3 R3))
 R3 IS THE CONCATENATION OF R1 AND R2
 (DUMPED)
 241 MSECS

 $*:$$$$$ (SURFACE (BUILD REL CONCAT A1 R3 A2 R4 A3 R5))
 R5 IS THE CONCATENATION OF R3 AND R4
 (DUMPED)
 249 MSECS

 $*:$$$$$ (SURFACE (BUILD REL GOES-THROUGH A1 R1 A2 W))
 R1 GOES THROUGH W
 (DUMPED)
 231 MSECS
```

```
$*:$$$$$ (SURFACE (BUILD REL EFFICIENCY A1 R1 A2 /5))
THE EFFICIENCY OF R1 IS 5
(DUMPED)
261 MSECS

$*:$$$$$ (SURFACE (BUILD REL EFFICIENCY A1 R2 A2 /7))
THE EFFICIENCY OF R2 IS 7
(DUMPED)
262 MSECS

$*:$$$$$ (SURFACE (BUILD REL EFFICIENCY A1 R4 A2 /3))
THE EFFICIENCY OF R4 IS 3
(DUMPED)
259 MSECS

$*:$$$$$ (SURFACE (BUILD REL SPAN A1 R1 A2 /3))
THE SPAN OF R1 IS 3
(DUMPED)
1157 MSECS

$*:$$$$$ (SURFACE (BUILD REL SPAN A1 R2 A2 /4))
THE SPAN OF R2 IS 4
(DUMPED)
274 MSECS

$*:$$$$$ (SURFACE (BUILD REL SPAN A1 R4 A2 /2))
THE SPAN OF R4 IS 2
(DUMPED)
274 MSECS

$*:$$$$$ (SURFACE (BUILD AVB ($X $Y $Z)
$*:$$$$$                 ANT (BUILD REL CONCAT A1 *X A2 *Y A3 *Z)
$*:$$$$$                 CQ  ((BUILD REL ROUTE-PART A1 *X A2 *Z)
$*:$$$$$                      (BUILD REL ROUTE-PART A1 *Y A2 *Z)) ))
IF V3 IS THE CONCATENATION OF V1 AND V2
   THEN V2 IS A ROUTE PART OF V3 AND
        V1 IS A ROUTE PART OF V3
(DUMPED)
2194 MSECS

$*:$$$$$ (SURFACE (BUILD AVB ($X $Y $Z)
$*:$$$$$                 &ANT ((BUILD REL ROUTE-PART A1 *X A2 *Y)
$*:$$$$$                       (BUILD REL GOES-THROUGH A1 *X A2 *Z))
$*:$$$$$                 CQ    (BUILD REL GOES-THROUGH A1 *Y A2 *Z)) )
IF V4 GOES THROUGH V6 AND
   V4 IS A ROUTE PART OF V5
   THEN V5 GOES THROUGH V6
(DUMPED)
1293 MSECS

;**************************************************************************
;*
;*   Q1: Does R3 go through W?
;*
$*:$$$$$ (SURFACE (DEDUCE 1 REL GOES-THROUGH A1 R3 A2 W))

SINCE
R3 IS THE CONCATENATION OF R1 AND R2

WE INFER
R1 IS A ROUTE PART OF R3

SINCE
R3 IS THE CONCATENATION OF R1 AND R2

WE INFER
R2 IS A ROUTE PART OF R3

SINCE
R3 IS THE CONCATENATION OF R1 AND R2

WE INFER
R1 IS A ROUTE PART OF R3

SINCE
R1 IS A ROUTE PART OF R3 AND
R1 GOES THROUGH W

WE INFER
```

```
R3 GOES THROUGH W


R3 GOES THROUGH W
(DUMPED)
7837 MSECS

;************************************************************************
;*
;*   Q2: Does R5 go through W?
;*
;
$*:$$$$$ (SURFACE (DEDUCE 1 REL GOES-THROUGH A1 R5 A2 W))

SINCE
R1 IS A ROUTE PART OF R3 AND
R1 GOES THROUGH W

WE INFER
R3 GOES THROUGH W

SINCE
R5 IS THE CONCATENATION OF R3 AND R4

WE INFER
R3 IS A ROUTE PART OF R5

SINCE
R3 IS A ROUTE PART OF R5 AND
R3 GOES THROUGH W

WE INFER
R5 GOES THROUGH W


R5 GOES THROUGH W
(DUMPED)
5062 MSECS

$*:$$$$$ (CLEAR-INFER)
(CLEARED)
16 MSECS

;************************************************************************
;*
;*   Assert further rules and define functions for weighted average and sum
;*
;
$*:$$$$$ (^(LENGTH(INPUT NFNS)))
(2)
51 MSECS

$*:$$$$$ (^ (DP WTDAVG (A1 A2 A3 A4 RESULT)
$*:$$$$$      (SETQ RESULT
$*:$$$$$          (UN-IZE (QUOTIENT
$*:$$$$$                      (PLUS (TIMES (N-IZE A1)(SUB1 (N-IZE A2)))
$*:$$$$$                            (TIMES (N-IZE A3)(SUB1 (N-IZE A4))) )
$*:$$$$$                      (PLUS (N-IZE A1)(N-IZE A3)))
$*:$$$$$                      ))
$*:$$$$$      (SUCCEED TRUE))
$*:$$$$$                          )
(WTDAVG)
7 MSECS

$*:$$$$$ (SURFACE (BUILD AVB ($X $Y $Z $W1 $N1 $W2 $N2)
$*:$$$$$        ANT (BUILD REL CONCAT A1 *X A2 *Y A3 *Z)
$*:$$$$$      CQ  (BUILD &ANT  (
$*:$$$$$               (BUILD REL EFFICIENCY A1 *X A2 *W1)
$*:$$$$$               (BUILD REL SPAN A1 *X A2 *N1)
$*:$$$$$               (BUILD REL EFFICIENCY A1 *Y A2 *W2)
$*:$$$$$               (BUILD REL SPAN A1 *Y A2 *N2))
$*:$$$$$        CQ  (BUILD AVB $TEMP
$*:$$$$$                   ANT (BUILD NAME: WTDAVG A1 *W1 A2 *N1 A3 *W2 A4 *N2
$*:$$$$$                                              RESULT *TEMP)
$*:$$$$$                   CQ  (BUILD REL EFFICIENCY A1 *Z A2 *TEMP))))) )
IF V9 IS THE CONCATENATION OF V7 AND V8
   THEN IF THE SPAN OF V8 IS V13 AND
           THE EFFICIENCY OF V8 IS V12 AND
           THE SPAN OF V7 IS V11 AND
           THE EFFICIENCY OF V7 IS V10
           THEN IF THE WEIGHTED AVERAGE OF V11 AND V13 (USING WEIGHTS OF V10 AND V12
```

```
RESPECTIVELY) IS V14
                     THEN THE EFFICIENCY OF V9 IS V14
 (DUMPED)
 5998 MSECS

 $*:$$$$$ (SURFACE (BUILD AVB ($X $Y $Z $N1 $N2 $N3)
 $*:$$$$$                 ANT (BUILD REL CONCAT A1 *X A2 *Y A3 *Z)
 $*:$$$$$                 CQ  (BUILD
 $*:$$$$$                        &ANT ((BUILD REL SPAN A1 *X A2 *N1)
 $*:$$$$$                              (BUILD REL SPAN A1 *Y A2 *N2))
 $*:$$$$$                        CQ    (BUILD
 $*:$$$$$                               ANT (BUILD NAME: SUM A1 *N1 A2 *N2
 $*:$$$$$                                                    RESULT *N3)
 $*:$$$$$                               CQ  (BUILD REL SPAN A1 *Z A2 *N3)))))
 IF V17 IS THE CONCATENATION OF V15 AND V16
    THEN IF THE SPAN OF V16 IS V19 AND
            THE SPAN OF V15 IS V18
            THEN IF V20 IS THE SUM OF V18 AND V19
                    THEN THE SPAN OF V17 IS V20
 (DUMPED)
 3874 MSECS

 $*:$$$$$ (^ (DP SUM (A1 A2 RESULT)
 $*:$$$$$         (SETQ RESULT (UN-IZE
 $*:$$$$$                        (PLUS (N-IZE A1)(N-IZE A2))))
 $*:$$$$$         (SUCCEED TRUE) ))
 (SUM)
 7 MSECS

;*************************************************************************
;*
;*  Q3: What is the efficiency of R3?
;*
 $*:$$$$$ (SURFACE (DEDUCE 1 REL EFFICIENCY A1 R3 A2 %E1))

 SINCE
 R3 IS THE CONCATENATION OF R1 AND R2

 WE INFER
 IF THE EFFICIENCY OF R1 IS V10 AND
    THE SPAN OF R1 IS V11 AND
    THE EFFICIENCY OF R2 IS V12 AND
    THE SPAN OF R2 IS V13
    THEN IF THE WEIGHTED AVERAGE OF V11 AND V13 (USING WEIGHTS OF V10 AND V12
RESPECTIVELY) IS V14
            THEN THE EFFICIENCY OF R3 IS V14

 SINCE
 THE EFFICIENCY OF R1 IS 5 AND
 THE SPAN OF R1 IS 3 AND
 THE EFFICIENCY OF R2 IS 7 AND
 THE SPAN OF R2 IS 4

 WE INFER
 IF THE WEIGHTED AVERAGE OF 3 AND 4 (USING WEIGHTS OF 5 AND 7 RESPECTIVELY) IS V14
    THEN THE EFFICIENCY OF R3 IS V14

 SINCE
 THE WEIGHTED AVERAGE OF 3 AND 4 (USING WEIGHTS OF 5 AND 7 RESPECTIVELY) IS 2

 WE INFER
 THE EFFICIENCY OF R3 IS 2


 THE EFFICIENCY OF R3 IS 2
 (DUMPED)
 20916 MSECS
;*************************************************************************
;*
;*  Q4: What is the efficiency of R5?
;*
 $*:$$$$$ (SURFACE (DEDUCE 1 REL EFFICIENCY A1 R5 A2 %E2))

 SINCE
 R5 IS THE CONCATENATION OF R3 AND R4

 WE INFER
 IF THE EFFICIENCY OF R3 IS V10 AND
```

```
    THE SPAN OF R3 IS V11 AND
    THE EFFICIENCY OF R4 IS V12 AND
    THE SPAN OF R4 IS V13
    THEN IF THE WEIGHTED AVERAGE OF V11 AND V13 (USING WEIGHTS OF V10 AND V12
RESPECTIVELY) IS V14
            THEN THE EFFICIENCY OF R5 IS V14

 SINCE
 R3 IS THE CONCATENATION OF R1 AND R2

 WE INFER
 IF THE SPAN OF R1 IS V18 AND
    THE SPAN OF R2 IS V19
    THEN IF V20 IS THE SUM OF V18 AND V19
            THEN THE SPAN OF R3 IS V20

 SINCE
 THE SPAN OF R2 IS 4 AND
 THE SPAN OF R1 IS 3

 WE INFER
 IF V20 IS THE SUM OF 3 AND 4
    THEN THE SPAN OF R3 IS V20

 SINCE
 7 IS THE SUM OF 3 AND 4

 WE INFER
 THE SPAN OF R3 IS 7

 SINCE
 THE SPAN OF R3 IS 7 AND
 THE EFFICIENCY OF R4 IS 3 AND
 THE SPAN OF R4 IS 2 AND
 THE EFFICIENCY OF R3 IS 2

 WE INFER
 IF THE WEIGHTED AVERAGE OF 7 AND 2 (USING WEIGHTS OF 2 AND 3 RESPECTIVELY) IS V14
    THEN THE EFFICIENCY OF R5 IS V14

 SINCE
 THE WEIGHTED AVERAGE OF 7 AND 2 (USING WEIGHTS OF 2 AND 3 RESPECTIVELY) IS 3

 WE INFER
 THE EFFICIENCY OF R5 IS 3


 THE EFFICIENCY OF R5 IS 3
 (DUMPED)
 33872 MSECS
;**************************************************************************
;*
;*   Asssert remaining rule and comparison function
;*
 $*:$$$$$ (^ (DP GREATER (A1 A2)
 $*:$$$$$        (COND ((GREATERP (N-IZE A1)(N-IZE A2))(SUCCEED TRUE))
 $*:$$$$$              (T (SUCCEED FALSE)) )))
 (GREATER)
 7 MSECS

 $*:$$$$$ (SURFACE (BUILD AVB ($N1 $N2)
 $*:$$$$$                 ANT (BUILD NAME: GREATER A1 *N1 A2 *N2)
 $*:$$$$$                 CQ  (BUILD REL ISGREATER A1 *N1 A2 *N2)))
 IF (GREATERP V21 V22 )
    THEN V21 IS GREATER THAN V22
 (DUMPED)
 2008 MSECS
;**************************************************************************
;*
;*   Q5: Is the efficiency of R5 greater than the efficiency of R3?
;*
 $*:$$$$$ (SURFACE (DEDUCE REL ISGREATER A1 (FIND A2- (FIND REL EFFICIENCY
 $*:$$$$$                                                    A1  R5))
 $*:$$$$$                                  A2 (FIND A2- (FIND REL EFFICIENCY
 $*:$$$$$                                                    A1  R3))))

 SINCE
```

```
(GREATERP 3 2 )

WE INFER
3 IS GREATER THAN 2

3 IS GREATER THAN 2
(DUMPED)
2919 MSECS

;*****************************************************************************
;*
;*  Q6: Is the efficiency of R3 greater than the efficiency of R5?
;*
$*:$$$$$ (SURFACE (DEDUCE REL ISGREATER A1 (FIND A2- (FIND REL EFFICIENCY
$*:$$$$$                                                   A1  R3))
$*:$$$$$                                A2   (FIND A2- (FIND REL EFFICIENCY
$*:$$$$$                                                   A1  R5)) ))

(DUMPED)
923 MSECS

$*:$$$$$ (LISP)
END SNEPS
$ ?$$$$$ (EXIT)
END ALISP RUN
```

BIBLIOGRAPHY

Kellogg, C. and Travis, L. Reasoning with data in a deductively augmented data management system. In Advances in Data Base Theory - Volume 1. Gallaire and Minker (eds.). To appear.

Klahr, P. Planning techniques for rule selection in deductive question-answering. In Pattern-Directed Inference Systems, Waterman, D.A. and Hayes-Roth, F. (eds.), Academic Press, New York, 1978.

Shapiro, S.C. and McKay, D.P. Inference with recursive rules, Proceedings of the First National Conference on Artificial Intelligence, AAAI, August, 1980.