RES Primer

Version 1 - Sept 2010

Bud Scott

PREFACE

This primer presupposes a prior, basic familiarity with the Logos Model's approach to machine translation.
Not every term or concept is defined therefore. Where needed, the reader should consult
the Glossary of Terms in Appendix B
Those unfamiliar with the Logos Model are forewarned that rules in OpenLogos are
in the form of numbers rather than symbols and take some getting used to.
It is suggested that the interested reader consult the RES commentary in the diagnostic file
for the sentence, *The sky is blue*, where an effort is made to explain the numbers.
This diagnostic is available as Appendix A to this Primer.
Basic information relating to the Logos Model is available on the OpenLogos Archives site at:
http://logossystemarchives.homestead.com
The serious reader will eventually want to become familiar with RES-related materials
available in the RES folder of the Archives site at:
http://logossystemarchives.homestead.com/RES/RES_Index.html
.

I. Introduction

1. . The function of RES is two-fold: (1) to eliminate POS ambiguities as found in the dictionary, and (2) to detect and record boundaries of all clauses and "nested" materials in a sentence. Nested materials include parenthetical and absolute constructions.
    a. In many respects RES rules look like TRAN rules, but their functions differ considerably.
    b. RES is the most brain-like module of the Logos Model and it takes some effort to understand its complexities.
        i. A good overview of RES function is given at the following websites:
        ii. http://logossystemarchives.homestead.com/Logos_SystemWEB_as_a_Biological_Neural_Net.pdf

iii. And on p. 24f of:
http://logossystemarchives.homestead.com/Logos_SystemWEB_--_Principles_and_Motivations.pdf

2. There are two RES modules in the OpenLogos pipeline, RES1 and RES2.
   a. Both RES modules have their own distinct set of rules, viz., RES1 rules and RES2 rules.
      i. Rather than a small number of deep, powerful rules, the Logos Model employs large numbers of shallow rules.
      ii. RES1 and RES2 collectively have c 7-8 thousand rules.
      iii. This primer assumes the reader is reasonably familiar with the philosophy and motivations of the Logos Model.

   b. To repeat, RES rules effect a macro-parse of the input sentence, consisting of:
      i. part-of-speech resolution, and
      ii. clause boundary recognition, including nested clauses.

   c. RES also supplies the TRANs with marco-level information about sentence clausal structure.
      i. Macro-parse information is passed to the TRANs in a 100-cell Scon record associated with each SWORK.
         1. This macro-parse information is summarized in the diagnostics in the Clausal Status Array (CSA) which appears at the end of RES2. The CSA shows data for both major and minor paths (clauses).
      ii. Reminder: SWORK stands for semantico-syntactic work record (or, equivalently, SAL work record).
      iii. For new readers, an SWORK is a numeric entity that represents (takes the place of) a natural language word or phrase. All RES and TRAN rules operate not on NL entities but on the corresponding SWORK entities that represent them.
         1. Of course a hash code for the literal NL word or phrase accompanies its SWORK and can be referenced where needed.
      iv. SAL is the representation language used throughout the system for both the input stream and rules. SAL stands for Semantico-Syntactic Abstraction Language.
         1. SAL is an actual language to which NL easily maps.
         2. Consult the Archives website for extensive info about SAL and its key role in the Logos Model

   d. The RES parse is deterministic, i.e., a single parse is produced rather than a parse forest.
   e. The RES modules are purely source-related and have no direct target functions.

3. RES1 is executed upon completion of dictionary look-up and the creation of the SWORK array. This SWORK array serves as input to RES1. The array contains all parts of speech found in the dictionary for the words or phrases of the given sentence.
    a. Output of RES! is input to RES2.
    b. Output of RES2 is input to TRAN1.
    c. This output from RES2 to TRAN1 is an SWORK array that has been pruned of all part-of-speech ambiguities.

4. The SWORK array which serves as input to RES1 is illustrated below, in simplified form.
    a. Note that this SWORK array below contains
        i. each literal word or phrase of the input sentence, as matched in the dictionary
        ii. the SAL codes for these words or phrases (WC, TYP, F, SBS, SPS)
        iii. up to 3 possible parts of speech (POS) found associated with the word or phrase in the dictionary.
            1. Note, however, that in the case of homographs, a particular part of speech associated with the word in the dictionary may already have been eliminated via morphological analysis during dictionary look-up
                a. E.g., the N/V homograph "building" is resolved to N when inflected for the plural, "buildings"
        iv. BOS stands for Beginning of Sentence, EOS for End of Sentence.

SWORK RECORDS

EL	**1st POS**					**2nd POS**					**3rd POS**					**Word or Phrase**
1	**WC**	**TYP**	**F**	**SBS**	**SPS**	**WC**	**TYP**	**F**	**SBS**	**SPS**	**WC**	**TYP**	**F**	**SBS**	**SPS**	
2	20	1	1	0	1	-	-	-	-	-	-	-	-	-	-	BOS
3	14	1	3	101	1	-	-	-	-	-	-	-	-	-	-	the
4	1	29	1	29	9	-	-	-	-	-	-	-	-	-	-	sky
5	2	60	3	886	11	12	60	3	886	1	-	-	-	-	-	is
6	1	40	13	609	6	2	21	1	546	7	-	-	-	-	-	blue
7	20	10	1	10	10	-	-	-	-	-	-	-	-	-	-	EOS

WC = Word Class; TYP = Semantic Type (i.e., SAL Set); F = Form; SBS = SAL Subset; SPS = SAL Superset

5. In this particular sentence, only two words are homographs, i.e. ambiguous with respect to part of speech, viz., *is* and *blue*, both of which entail two possible parts of speech.

6. As stated, one of the two purposes of the RES modules is to resolve these ambiguities to a single POS.
    a. RES generally resolves POS ambiguities to an accuracy of c. 97-98% for reasonably well-written, discursive text.
    b. Very occasionally, RES decisions regarding POS resolution may be reversed by subsequent TRAN analysis.
        i. Information about all original POS's is provided by RES to the TRANs via the Scon record associated with each SWORK.

7. How RES accomplishes POS resolution is explained in what follows:

II. How RES Resolves ambiguous POS to a single POS.

1. **SWORK Array**. The SWORK array sent to RES1 contains, in principle, every conceivable combination of SWORKs (conceived of as "paths") inherent in a given sentence as it comes out of dictionary look-up.
    a. The actual sentence, of course, can be represented by only one SWORK combination, i.e., by a single path through the possible POS combinations latent in the array. It is the function of RES to determine this path.
    b. Each of the possible paths through the SWORK array will becomes a search argument to the RES rulebase, one path or path portion after the other. In the process, the rules effectively delete some POS and retain others.
    c. In the sentence *The sky is blue*, there are four such possible path combinations. (Here we only show WC, omitting the semantico-syntactic codes and the form fields.) Each of these path patterns, in effect, becomes a search argument to the RES rulebase, seeking a RES rule with a corresponding SWORK pattern in the rule's SP line.
    d. These are the four combinations for *The sky is blue*:
        i. WC20 + WC14 + WC1 + WC2 + WC1 + WC20
        ii. WC20 + WC14 + WC1 + WC12 + WC1 + WC20
        iii. WC20 + WC14 + WC1 + WC12 + WC2 + WC20
        iv. WC20 + WC14 + WC1 + WC2 + WC2 + WC20
            1. NOTE: In longer sentences, the number of path combinations can quickly become quite large. E.g., in one 57-word sentence, the number of possible SWORK combinations reached well above a million.
            2. Of course, the system does not actually create an explicit table or list of a million combinations. (The above combination list is merely for illustration purposes.) But, indeed, all these path combinations are latent in the SWORK array, and each combination comes into play as RES works its way through

the SWORK array, seeking a single correct path among the million plus possible paths (in the case of the cited 57-word sentence).

e. In sum, then, each one of these SWORK path combinations (or portion thereof, in longer sentences) is employed, one after the other, as search argument in seeking to march on rules in the RES rulebase, looking for one (or more) with a corresponding SWORK pattern in its SP line.
   i. Matches are made against the database much in the way NL words are looked up in a dictionary. Note that each SWORK constitutes an object much like a NL word.
   ii. The two RES rulebases (one for each of the two RES modules) can be thought of as a pattern dictionary.
      1. See the diagnostic file in the Appendix A for a description of the four components of an OpenLogos rule, viz., (1) Comment line; (2) SP line (for semantico-syntactic or SAL pattern); (3) Constraint Line; (4) Action line.
   iii. Matches against the RES rulebases are conducted on the longest match principle.
      1. Most typically, rules both in the RES and TRAN modules tend to be quite short, mostly two or three SWORKs in length.
      2. Accordingly, RES works its way down the possible paths in small incremental steps as matches are made.
      3. By the end of RES2, all POS ambiguities will have been resolved, and all other path combinations but one will have been eliminated.
   iv. In sum, resolution of the correct path (among all possible SWORK combinations) is effected incrementally, working along the combinations from left to right. At the end of RES2, then, only one path combination is passed to TRAN1.
      1. The sentence has thus been parsed at the macro-level.
      2. TRAN1 will then begin a micro-parse.

f. Unlike TRAN, RES rules do not have to match every element (SWORK) in a sentence. This is an important difference.
   i. Homographic elements not explicitly dealt with by a rule in RES will automatically resolve to the first POS in the dictionary for the given word.
   ii. Thus, RES rules need only be written to deal with (1) POS ambiguities that must be resolved to an SWORK other than the first POS given by the dictionary for the word; and (2) to detect and record the beginning and end of all clauses, including relative clauses, parentheticals, and absolute constructions.

iii. Also unlike TRAN, RES cannot have a one-element rule, i.e., RES rules must be two or more elements in length. The reason for this is explained later.

g. Unlike RES2, RES1 rules do not automatically resolve homographs just by virtue of the match. For RES1 to resolve a homograph, it must have a -13 switch in the action line. This -13 switch has a single parameter.
   i. When the single parameter is a relational pointer (e.g., -81), only the element pointed to is locked, i.e., fixed to the part of speech in the matched pattern. (E.g., if the homograph is DET N/V and the matching rule is DET N, and the action line of the rule has -13 -82, then the N/V homograph is resolved by this RES1 rule to N)..
   ii. When the parameter is the value 1, then all elements in the SP line are locked (except those affected by the backspace function (-41 switch)..

h. One key function of RES1 is to gather intelligence about individual SWORKs for subsequent use by RES2 rules in its final, definitive effort to effect homograph resolution.
   i. Note that RES2 must resolve all homographs not already resolved by morphology during dictionary look-up or by RES1.
      1. As stated, any homographs not explicitly dealt with by a RES1 or RES2 rule will be resolved by default to the first SWORK provided in the dictionary for the ambiguous element.
         a. The first position of a POS in the dictionary is generally assigned to the most probable POS
      2. Considerable reliance is made of this default function by the RES modules.
      3. In sum, RES rules need only be written where it is judged necessary to override this default resolution.
   ii. RES1 defers the more difficult (less clear-cut) resolution tasks to RES2, but often provides a key piece of intelligence to RES2 for this purpose. This is most especially true in the case of N/V homographs. To illustrate, take the example of the N/V homograph *blue* in our example sentence..
      1. In the present sentence, of course, *blue* could not be a verb, but if in some other sentence the N/V homograph *blue* had been followed by an article (determiner), then some RES1 rule would detect that it could possibly (but not necessarily) be a verb in such a context, and the RES1 rule would thus label the verb SWORK for *blue* as a possible transitive verb (PVT).
      2. RES1 informs RES2 of this possibility by changing the TYP field for *blue's* verb SWORK to 862, signifying PVT. In effect, 862 signifies tht the SWORK for *blue* as a verb satisfies necessary conditions that could allow RES2 to resolve it as a verb.
         a. RES1 labels possible intransitive verbs (PVI).863.

b. 862 and 863 are called SUBCHG codes. For a complete set of SUBCHG codes for all parts of speech, see the document *RES_Strategy_and Cell_Definition*, available at the Archives website

3. It is now up to some RES2 rule to decide whether in fact *blue* is indeed a verb in this instance, and if so, the RES2 rule resolves the N/V to V, or if the homograph is N/VI/VT, then to VT.
   a. Or, to repeat, in the absence of an explicit RES2 rule dealing with the several POS possibilities for *blue*, the system will default to the POS in the first POS position for *blue* in the SWORK array of this sentence.
4. This 862 and other related SUBCHG codes are only temporary to RES. The original TYP field (SAL Set code) of the SWORK is restored to its original value in the SWORK array that is output to TRAN1.
   a. Certain SUBCHG codes however, remain throughout RES and TRAN, viz., those relating to orthography (e.g., codes denoting Initial Cap word, etc).
5. As stated, TRAN1 has facilities to reverse a resolution made by RES but this is rarely done.
   a. In some few cases, however, RES intentionally leaves some POS decisions to the TRANs. For example, a PREP/ PARTICLE homograph like *down* (as in, e.g.,(a) *turn <u>down</u> the street* or (b) *turn <u>down</u> the offer*) is usually provisionally resolved by RES to VI PREP, leaving it to TRAN2 to reverse this decision if necessary.. TRAN2 makes this determination this by sending both possibilities to Semtab for resolution. In the case of (b), above, a match on a Semtab rule dealing with the verb construct *turn down* would cause TRAN2 to re-resolve VI PREP to VT PART
   b. This Semtab rule also have a target function, changing the target transfer(s) for the newly resolved construction.

iii. RES1 rules may also designate a N/V homograph as a <u>weak</u> possible verb, (WPVT, or WPVI).
   1. RES1 labels WPVT as TYP=864 and WPVI as 865
   2. A typical context for WPVT would be, e.g., "COMMA + VT + N(PL)". RES2, by now having other data to draw upon, might still decide this construction should resolve to "COMMA + N N(PL)" as in, e.g., *", steady states"*.

iv. As stated, by the end of RES2, all POS ambiguities have been resolved and each word is now represented by a single SWORK.

i. RES2 is the main workhorse of RES.
   i. Every RES2 rule just by virtue of its match automatically "fixes" the SWORK pattern it matches on..

ii. By "fixing" we mean that, in RES2, the matched upon SWORK now becomes the only SWORK for the given word. All other SWORKS for that word (assuming the word is a homograph) are automatically eliminated.
iii. In sum, this is how RES2 establishes the path through the network of possible combinations, i.e., how RES2 effects POS resolution, viz., simply by effecting a match between an SWORK input stream and a RES2 rule. I.e., the very match itself is sufficient basis for resolving the homograph.
    1. RES1 can also "fix" a homograph to a single SWORK but only by explicit use of a -13 switch (discussed later).
iv. Obviously, this being the case, RES2 rules have to be written with complete awareness of other rules that might also quality for a match in the given context, and potentially cause a different and possibly incorrect resolution.
v. Thus, rules in RES, and especially in RES2, have to be written with awareness of the total body of rules and how they compete with each other for the right to fire.

2. **Rule Application.** How are RES rules applied? How do they compete for the right to fire?

a. Four factors determine which rules (among those that match an input pattern) actually get to fire. These are:
    i. Rule's authority code This is the single most important factor in rule competition.
    ii. Rule's length. This is the second most important factor.
    iii. Rule's semantico-syntactic specificity. All other factors being equal, iii and iv now come into play.
    iv. Rule's extra "weight" factor, assigned arbitrarily by rule-writer to favor a rule.

b. In effect, then, each rule in the RES rulebase has an <u>authority code</u>, displayed (in diagnostics) at the head of the SP line, e.g., **32** (14 -1 31) (1 -2 44). The significance of **32** is explained hereunder:
    i. This authority code is a two-digit code (NN), each digit of which has a special meaning, as follows:
        1. N1 is a priority code, ranging from 0 to 3, zero being the lowest priority, 3 being the highest priority.
            a. This priority code is assigned by the rule-writer. It is the chief means available to him or her for prioritizing rule application
        2. N2 is the length code. A rule that encompasses a pattern of two SWORKS will have a 2 in N2
            a. N2 is also assigned by the rule writer but is verified by the system during rule generaition.
        3. All other things being equal, a rule with authority code 32 will execute before 22, etc.
            a. Note that shorter rules with higher two-digit authority codes will fire before longer rules with lower authority codes, e.g., 32 will take precedence over 24.

ii. Rules of equal authority code are further distinguished by their degree of semantico-syntactic specificity.
    1. Specificity is summed for all elements in the SP line:
        a. WC (if specified) = +1
        b. TYP (if specified, including -2) = +2
        c. Form (if specified) = +1
        d. Note that nothing is added whereever values for WC, TYP, Form are negative (except for the case of -2) i.e. are left unspecified. (-2 points to a constraint line where a string of TYPs may be found)

iii. Rule-writers can also add an additional "weight" to the priority of a rule.
    a. This additional rule "weight" is displayed at the head of the comment line (in the diagnostics).
    b. For example: `+4 (XMV)(NUM=0) ART (U/PV2RT) N(SG) = -2 EP1085 OERES2`
       `32 (14 -1 31) (1 -2 44`
    c. This "weight" is added into to the total priority code of a rule.

iv. Note lthat TRAN rules do not have authority codes. In the TRANs, rule specifity and rule lengthy alone determine firing.

c. In sum, rules with the most specific semantics will be looked at first. Within this class of rules, those with the highest authority code (including any added "weight') will fire first. All else being equal, rules with the greatest length will fire first.

d. All of the foregoing, of course, presupposes that the constraint line of the RES rule has been satisfied.
    i. Rules become "active" by virtue of the foregoing but do not "fire" unless their constraint line is satisfied.

3. **Constraint Line of a RES Rule**. What are they?

a. A rule's constraint line specifies conditions that must be met for the rule to fire (i.e, conditions beyond that of SWORK matching itself).
b. The 2nd line of a RES rule is the constraint line (just as in TRAN rules). ,
    i. Unlike in the rule's Action line (where everything has the 3-digit form NNN), constraint line elements have a 4-digit form NNNN

ii. In both RES and TRAN, leading zeroes are required when the linguist writes his or her rules, but leading zeroes are suppressed in diagnostic printouts.
iii. Leading 888888 at the beginning of a constraint line indicates that the constraint line is continued on the next line.

c. To repeat, constraint lines test for conditions that must be satisfied for the rule to fire.
d. This test is effected by a set of so called 6000 functions (sometimes also called switches), designated by numbers like 6012, 6050, 6399 6500, etc. These are briefly described below. (See *RES Switch Summary* file for more complete description of these functions and of others like them not covered here):
   i. 6010 P1 P2
      1. P1 is a relational pointer, indicating the element in the matched SWORK pattern to which P2 applies
      2. P2 is any number of SAL set, superset, or subset values, any one of which must be present in the SWORK for the rule to fire.
      3. 9000 is end of function sign
   ii. 6012 P1 P2
      1. P1 is a relational pointer, indicating the element to which P2 applies
      2. P2 is any number of SAL set, superset, or subset values, all of which must NOT be present for the rule to fire.
      3. 9000 is end of function sign
      4. Example : 6012 82 851 849 848 900 9000
   iii. 6013 P1 P2 (for use when WC of SWORK is -1) (i.e., when WC is unspecified)
      1. P1 is a relational pointer, indicating the element to which P2 applies
      2. P2 is any number of word classes, any one of which must be present in the WC field of the relevant SWORK for the rule to fire.
      3. 9000 is end of function sign.
   iv. 6014 P1 P2 (for use when WC of SWORK is -1) (i.e., WC is undefined)
      1. P1 is a relational pointer, indicating the element to which P2 applies
      2. P2 is any number of word classes, all of which must NOT be present in the WC field of the relevant SWORK for the rule to fire.
      3. 9000 is end of function sign.
      4. Example: 6014 82 12 14 6 9000
   v. 6025 P1 (ambiguity tests)

1. P1 is a relational pointer
2. Function tests if P1 is unambiguous (i.e., this matched upon SWORK is the only SWORK for the given word)
3. Example: 6025 81

vi. 6050 P1 P2 (tests for cell value(s))
1. Any number of cells may be tested
2. AND is implied, 777 signifies OR
3. 9000 is end of function sign
4. Example: 6050 1004 8000 1103 8021 9000

vii. 6055 P1 P2 P3 (compares two or more SWORKS for shared values in, set, subset, superset or form)
1. P1 is a relational pointer
2. P2 is a control field indicating what in particular this is to be a test for, viz., set, superset, etc.). The 4th digit P2 specifies the number of elements being compared to P1. (This 4th digit replaces the 9000 in other 6000 functions.)
3. P3 is one or more relational pointers to SWORK(s) to be matched against P1.

e. 6300 Function in the Constraint Line requires special explanation.
i. RES has a so-called "look ahead" capability, which means that at any point in the course of analysis, a RES1 or RES2 rule may ask the system to "look to the right" for some particular element (most typically a verb that is either unambiguously verb or, in the case of RES2, is a N/V homograph that has been previously marked by RES1 as PVT or PVI).
ii. The success or failure of this look-ahead will affect the ability of the originating rule to fire.
iii. The 6300 series contained hardwired searches for U/PVs.
1. 6353 searches for V (sg) and V(ed).
2. 6357 searches for V (pl) and V (ed).
3. 6360 has special criteria for helping to establish a V(ed) vs N Adj'd (past adjectival participle) .
iv. The 6300 searcj to the right entails SKIPs and QUITs which are rule-based.
1. The priority for rules matching for both 6300 is SKIP, QUIT, HIT.
v. By way of illustration, 6300 is the mechanism that allows RES to deal with so called "garden path" sentences like *The horses run past the barn fell*..

1. Without a look ahead capability, a RES2 rules would inevitably have to resolve *run* to an *intransitive* verb.
2. The 6300 look-ahead capability will allow RES to see run as the *past adjectival participle* of an *transitive* verb run and to see *run past the barn* as an elided relative clause (*that were run past the born*)

f. 6400 Function is the negative of 6300, i.e. a failure to find a hard-coded HIT constitutes TRUE.
   i. Both 6300 and 6400 have fairly extensive sub-functions, too numerous and complicated to discuss in this introductory primer. These may be found in the *RES Switch Summary* file available on the Archives website.
   ii. Looking-ahead for information that will help the decision of a present rule is obviously critical to any deterministic parser, and obviously entails complicated machinery, especially where all these operations must be accomplished by the matching of pattern-based rules.

g. 6500 function. The 6500 is a more flexible look-ahead function. The function entails a search to the right for specific patterns defined in the SP line of WC7 rules.
   i. In a 6500 look-ahead, the SWORKs to the right of the triggering rule become search arguments against rules stored in WC7 of the RES database.
   ii. 6500 searches are commonly employed to help define a series, or a clause transition
   iii. A successful match results in the function being TRUE, contributing to the ability of the originating rule to fire
   iv. A set of FILTER rules in WC10 and 11 are concurrently exercised to constrain WC 7 matching.
      1. The priority for rules matching is SKIP(WC11), QUIT(WC10), HIT(WC7).
         a. You may have noticed that WC 7, 10, and 11 have no counterpart in the SAL word class scheme, and thus are available for special purpose rules such as these
   v. Here's an example 6500 functioin:
      1. . 6500 81 8000 8052
         a. 81 is a relational pointer indicating the point from which matching against WC7 rules is to begin (in this case beginning to the right of the 1st SWORK)
            i. An 80 in this parameter would start the search with the 1st SWORK
         b. 8000 tells the system to use the standard SKIP and QUIT patterns defined for 6300
         c. Variations:
            i. 80XX - use Skip and Quit patterns defined in WC 11 065 XX (for SKIP) and 10 065 XX.(for UIT).

ii. 8052 search for the personal pronoun I (code: 01 795 11). If this pattern is found, the form field of the pronoun I is temporarily changed to 12 for RES (restored to 11 for TRAN).
    1. NOTE: RES needs to have the form field for this pronoun *faked* to plural for the sake of Subject/Verb agreement needed in subsequent potential N/V homograph resolution (e.g. *I / They very seldom work on Sunday* versus *He very seldom works on Sunday…)*

vi. 6800 Function. This is an infrequently invoked function having to do with searches to the right of a possible clause boundary (PCB) to determine whether this PCB is indeed a CB. P1 of this function is a pointer to the element beyond which searching is to start. P2 points to an element construed as a PCB.
    1. The 6800 function has fairly complex logic. It allows next SWORKS to the right of P1 to be processed provisionally, such that Action Line functions performed during this search can be undone upon encountering something that prevents this PCB from being a true CB, as, e.g., when the verb of a new clause is expected and EOS is encountered instead.

4. **Action Line of a RES rule**.
    a. The fourth and final component of a RES rule is its Action Line.
    b. See commentary in RES diagnostics in Appendix A for description of the four components of RES and TRAN rules, viz.,
        i. Comment Line;
        ii. SP Line,
        iii. Constraint Line;
        iv. Action Line

    c. The Action Line consists entirely of pre-programmed functions called switches, with 999 signifying end of Action Line.
        i. Switches are identified by negative numbers, e.g., -13, -18, -41 etc.
        ii. In RES, these numbered functions range from -13 to -46 (with many numbers unused)
        iii. See *RES Switch Summary* for comprehensive list of Action Line switches.

d. Below we show all four components of the RES2 rule that matches on "The sky.**"** The data shown has been extracted from the diagnostics file in Appendix A.
e. In what follows, text in **black** = original system output, text in **blue** = emphasis on original output, **green** = added commentary

```
Res2 rule #8358, ID: 8177
Comment Line     +4 (XMV)(NUM=0) ART (U/PV2RT) N(SG) = -2 EP1085 ORES2  +4 is a weight that increases rule priority
                                                                        XMV=expect main verb
                                                                        U/PV2RT = unambiguous or possible verb to
                                                                        right
                                                                        = -2 means the rule backspaces two elements.
                                                                        The equal sign indicates what the rule does.
                                                                        Recall however that, in RES2, any ambiguities
                                                                        associated with these elements would be
                                                                        resolved just by virtue of the match.

SP Line          32 (14 -1 31)  (1 -2 44)     32 signifies highest priority (3 is priority code, 2 is length within
                                                 that code). The two SWORKs here are for (1) any singular determiner and (2)
                                                 any singular N (forms 31 and 44 both represent sets of form codes denoting
                                                 singular).
                                                 The -2 in the TYP field of the second element merely tells the system that
                                                 a constraint line has been attached to this element.  Note that the tests
                                                 in this constrain line however need not pertain to just this element.

Constraint Line: { 6050  103 8021    4 8000 9000 6014   82    6 9000 6355   82 8000    0    0    0    0    0    0    0 }
      6050 in the Constraint line tests cell 03 for value of 21, and cell 4 for zero. 9000 delimits the function.
      6014 function excludes match on element 2 if it WC is 6.
      6355 looks to right for an unambiguous verb, or a verb marked by RES1 as PVI or PVT (possible verbs)
            All these conditional tests are successful.
            (See RES Switch Summary for fuller explanation of these Constraint line functions.)

VTR Line:   -22 -81 -82   0   0   0 -46 -81   0   0  10 -31  56 -41   2 999
      The -22 switch sends -81 and -82 (relationaol pointers to the first and second elements) to SEMRES. The switch
      has five parameters, allowing up to five SWORKs to be sent to SEMRES. SEMRES is also sometimes called RES22.  See
      RES22 Match below.
      The -46 switch relables the form field of the first element. The WC and TYP field are left untouched
      The -36 switch tells the system NOT to match on this rule immediately again (given the backspace all the way)
      The -41 switch tells the system to backspace 2 elements, i.e. all the way. In effect, an attempt will be makde to
      match this input pattern again, but not by this rule.
      999 is end-of-rule marker
```

*** **RES22 MATCH** **A successful match in SEMRES is found (see below). NOTE: SEMRES (RES22) is the RES version of SEMTAB (Semantic Table) extensive use of which is made in the TRANs) SEMRES and SEMTAB are special purpose pattern dictionaries whose chief purpose is to support homograph resolution in RES and semantic disambiguation in the TRANs. Both these special rulebases (pattern dictionaries) are invoked by the -22 switch. In RES, however, the -22 switch has an additional function: the rule invoking the -22 switch send to SEMRES will fail if the SEMRES match fails. This is not the case in the TRANs.**

Res22 rule #6367, ID: 6313 **This is the matched SEMRES rule.**
Comment: THE N JP584 ERES22
SP Line: 3 (14 101 1) (14 101 -1) (1 -1 42) **NOTE: The first SWORK, 14 101 1 (for "the"), is a "rule index" and is not actually part of the SP pattern. Rule indices are invariably based on the first element of the SP line.**
Constraint: **None.**
VTR Line: 999 **The 999 is end-of-rule marker. The sole action of this rule is effected by the match itself, which has the effect of confirming the match in the originating RES2 rule.**

5. **FINAL OBSERVATIONS AND SUMMARY:**
   a. RES1 resolves POS ambiguities via the -13 switch.
   b. A RES2 rule, on the other hand, resolves an ambiguous SWORK to a single meaning by the simple fact of matching.
      i. In other words, when RES2 rules matches on an SWORK pattern in the SWORK input array, that match in-and-of-itself effects resolution.
      ii. The single exception pertains to the last SWORK of the rule, which is partially left open. For example, if the WC of the last SWORK is -9 (a negative WC signifying either WC2 of WC12), all other WC's associated with the element are excluded but these two (e.g., the noun meaning of the word 'can', which is N/V/AUX homograph, in this case would be excluded from further matching.
         1. If the negative WC in terminal position is -1, (meaning any WC), then all WC's for that element remain open.

   c. It is clear therefore that RES rule writers have to be very mindful that rules not be allowed to match when they should not match.

d. In practice, this system, despite its complexity and delicacy, has worked fairly well.
    i. Homograph disambiguation has been shown to be correct 97-98% of the time.
        1. To a very minor extent, TRAN's subsequent micro-parse may contribute to this figure by correcting or refining the RES macro parse.
            a. For example, TRAN2 will sometimes find reason to change a RES macro parse of VI PREP N to VT PART N (i.e. altering the verb and changing a preposition to a verb particle). TRAN2 will do this on the basis of a send to SEMTAB.
    ii. RES is less effective when it comes to detecting clause boundaries, correctly identifying them only about 90% of the time in general purpose text.
        1. This 90% figure may improve in the case of carefully composed test, e.g. technical manuals composed by tech writers.

SOME FINE POINTS ABOUT RES PROCESSING

1. The SP line in RES rules must have at a minimum two elements (two SWORKS).
    - There are two reasons for this:
        - (1) Every RES rule automatically backspaces one element (SWORK). In effect there is an implicit -41 1 switch at the end of every RES rule, unless of course the rule writer intends to backspace more than one element, in which case the -41 switch is explicit.
        - (2) Unlike the TRAN rules, the function of RES rules is to create a path or transition from one SWORK to the next. In doing so it eliminates all the other POS possibilities of a given element in a given sentence and in this manner effects POS resolution. In contrast, TRAN rules need not concern themselves with transitions from one SWORK to another as RES has already accomplished this. TRAN rules thus can deal with a single element (SWORK) if that suits the purpose of the TRAN rule writer.
2. As you will have noted by now, the dictionary can have up to three parts of speech (POS) for a given word. (If a word has more than three parts of speech, and in English a number of words do, then it is the task of a TRAN rule to detect and supply the missing POS where necessary. TRAN does this via sends to SEMTAB.

- In this connection, it is very important to realize the following:
- When a word has more than one POS in the dictionary, the most probable part of speech is placed first.
  - This order is not word-specific however, but is based on generic combinations. Every ambiguous word falls into an ambiguity class as defined by a table (with over 200 classes of POS combinations). This table determines the POS order in the dictionary entry for the given class. Thus homographs of the same ambiguity class will have the same POS order in the dictionary.
  - In the case of N/V homographs, the N is always placed first.

3. If some RES rule does not explicitly select a POS for a homograph, then the system automatically selects the first (assumed to be the most probable) POS in the dictionary entry for that word.
   - This is the DEFAULT action of the system with respect of POS resolution, and RES relies upon it to a great extent.
   - What this means is that RES need NOT explicitly account for every POS resolution, i.e. RES rules need only be written to cover exceptions to this system default.
     - Given this, it is worth nothing that had there been no RES rules at all for our sentence, ***The sky is blue***, the default action of the system would have correctly resolved the POS ambiguities in the sentence, i.e., correctly identified the path through the SWORK array.
     - As may be seen, the only amhiguous words in this sentence, viz., ***is*** and ***blue***, both have "main verb" and "adjective" (respectively) as their default (most probably) POS. Thus, to repeat, the default action of the system would have provided the correct macro-parse.
   - To understand RES, it is important to realize that this default action is a background assumption of the RES rule writer.
     - In effect, rules were written whenever, in testing, the default action was shown to be inadequate.
     - Thus, rules were not written in the spirit of au augmenmted transition network (ATN), where all legitimate (well-formed) transitions from one POS to another must be specified in order for the parse to be accomplished.
     - For this reason, while well-formedness issues are certainly involved in the macro-parse, e.g., subject/predicate agreement in effecting N/V homograph resolution, and in detecting and identifying clause types and transitions, the

RES macro-parse offers no judgement as to the well-formedness of the sentence as a whole. RES has no idea therefore whether or not a sentence is grammatical.

4. The parts of speech that RES discards are preserved and stored in the Scons associated with the SWORK of each resolved homograph. This will allow a TRAN rule to override RES and reinstate the discarded POS of a given word. However, this happens rather rarely.

APPENDIX A

Intermediate OpenLogos System Diagnostics for the sentence “The sky is blue.” Commentary provided by B. Scott

NOTE to Reader: It is impossible for me to make intelligible all the numbers and codes of this diagnostic. Given its abundance and complexity, one can only deal cursorily with the material. Reader is advised to have first read the RES and TRAN primer before attempting to cope with this diagnostic.

Earlier commentary is in red, later commentary is in green. Text in black is system output. Blue is used to draw attention to system output.

```
*Original Input: Line #1*
The sky is blue.

*Pattern Matcher*

Pattern matcher has no rules

*Start Rules*
The sky is blue.

*Proper Name Recognition*

*Complete Lookup phase*
The sky is blue.

*Dictionary Match*
bos the sky is blue .
^   ^   ^   ^  ^    ^

INPUT: bos
DICT:  bos
INPUT: The
DICT:  the
INPUT: sky
DICT:  sky
INPUT: is
DICT:  is
INPUT: blue
DICT:  blue
INPUT: .
DICT:  .
```

SWORK RECORDS **SWORK = semantico-syntactic work (record)**

NOTE:Words in the dictionary may have up to three parts of speech (POS)> The most probably POS is normally placed in the first position. The RES modules make important use of this fact (See commentary below for RES).

```
 #  xx wc typ fr sbs sps patstm schg com smc            o2b o3b meaninID| wc typ fr sbs sps patstm schg com smc
o2b o3b meaninID| wc typ fr sbs sps patstm schg com smc            o2b o3b meaninID|
```

wc=word class; type=SAL set; fr=(morphological)form; sbs=SAL subset;sps=SAL superset
(SAL=semantico-syntactic abstraction language (the internal representation language))
patstm=the generic paradigmatic morphology table (PAT) that identifies stems and suffixes of all inflectable words. NOTE: In the case of dictionary entries for already inflected forms (e.g., 'is') this field is 0).
schg=subset change (refers to codes assigned terms that have initial caps, all caps, bold facing, etc.
smc=subject matter code, used to select between alternative entries for a given word, depending on user specification at run time

```
 1  -1 20   1  1   0   1   0  0    0 LOG 001              0   0       0 |  0   0  0   0   0   0  0    0     001
0   0       0 |  0   0  0   0   0   0  0    0     001              0   0       0 |  1 bos = beginning of sentence
```

wc for 'bos' is 20 (used for all punctuation); Type=SAL superset=1(unique for BOS);form=1 (no significance)

NOTE: initially, initially there are no SAL set and subset codes for BOS (some are assigned later during analysis).
NOTE: SAL superset codes range from 1 to 16, set codes from 17-99, subset codes from 100-998. If there is no SAL set or subset code for a given word, as in the case of BOS, the superset code will occupy the type field (as in the case of BOS.

```
 2   2 14   1  3 101   1   0  0    0 LOG 001              0   0   86452 |  0   0  0   0   0   0  0    0     001
0   0       0 |  0   0  0   0   0   0  0    0     001              0   0       0 |  1 the
```

wc for 'the' is 14; SAL superset is 1; form is 3; no SAL Set. SAL Subset is 101;

NOTE: the SAL superset occupies the type fields because there is no set code for this word. This lack of strict consistency reflects the fact that SAL was developed inductively and very gradually over a period of many years, not always with the issue of strict consistency in mind. Nevertheless SAL has proved to be of immense value to the process (as will be evident as we go along)

```
 3   3  1  29  1  29   9   0  0    0 LOG 001              0   0   80811 |  0   0  0   0   0   0  0    0     001
0   0      0 |  0   0  0   0   0   0  0    0     001              0   0      0 |  1 sky
```

wc for 'sky' is 1 (noun class); type is 29 (see SAL chart in Logos Archive Website for meaning); form=1 (for singular). Note that as there is no subset code for this class of nouns, therefore the set code is repeated in the subset field. Some noun supersets have both sets and subsets, some only have sets.

```
 4   4  2  60  3 886  11   0  0    0 LOG 001              0   0   52846 | 12  60  3 886   1   0  0    0 LOG 001
0   0  52845 |  0   0  0   0   0   0  0    0     001              0   0      0 |  1 is
```

wc for 'is' as a verb is 2 (verb class); type is 60; form is 3 (3rd per singular present tense); subset is 886; superset=11 (one of three intransitive classes)

NOTE: THERE ARE TWO PARTS OF SPEECH FOR 'IS'. The system only allows for a maximum of three POS. Words that have more than three POS have to be handled (disambiguated) during analysis.

wc for 'is' as an auxiliary is 12 (auxiliaries and modals); form=3; subset-886; superset-1 (differing from wc2)

NOTE: For verbs, the subset code, in combination with the set code, uniquely identifies each verb, allowing rules to specify individual verbs by this subset-set combination. Documentation exists (but is not presently available) specifying each verb and its set/subset combination. This of course is recoverable from the DB.

```
 5   5  1  40 13 609   6   0  0    0 LOG 001              0   0   18142 |  2  21  1 546   7   0  0    0 LOG 001
0   0  18143 |  0   0  0   0   0   0  0    0     001              0   0      0 |  1 blue
```

wc=1 (for nouns and n/adj homographs); type-40; form-13 (this form signifies a noun that may also be an adjective. If the adjective 'blue' were unambiguous, the form would be 23)(wc would be 4 later changed to 1); subset=609; superset=6 (these are the SAL codes for 'blue' as a noun).

NOTE: Noun/adjective homographs like 'blue" are resolved by TRAN rules (see TRAN2 below).
NOTE: In some graphics of the system, a distinction is made between PARSE and TRAN rules, reflecting the redesign of a newer version of the System that was never completed. In this diagnostic, TRAN refers to Transformation Rule and pertains to both analysis and synthesis functions. (See below in TRAN1 for fuller discussion)

wc=2 (for verb, i.e., verbal sense of 'blue'); type = 21 (simple transitive);form=3 (as above for 'is'); subset=546; superset-7 (transitive)

NOTE: Superset 546 in combination with superset 7 uniquely identifies this verb, as explained above.

```
 6   6 20  10  1  10  10   0  0    0 LOG 001              0   1       0 |  0   0  0   0   0   0  0    0     001
0   0       0 |  0   0  0   0   0   0  0    0     001              0   0       0 |  0 .
```

wc=20 (punctuation); type=10 (EOS - end of sentence); form=1 (no meaning). The superset is repeated in the other SAL fields.

SentenceUnits **all literal strings have hash codes that allow rules to specify the string. Except for verbs, very few open class words have unique codes and therefore analysis must rely upon hash codes to specify a word uniquely. There are two different hash codes, the second one is called a hen number (after a man named Hennesey, a Logos programmer who designed the code). Hen hash codes give a hash value for an entire entry (incl phrases). Regular hash codes only hash the head word.**

```
   hash-1 hash-2 roothash-1 roothash-2 hen-1 hen-2 roothen-1 roothen-2 wct CQDUBI
 1      0      0          0          0    -1    -1         0         0   0
 2      0      0          0          0  8302     5         0         0   1 I
 3   8490      4       8490          4  8490     4         0         0   1
 4      0      0          0          0  8030     0         0         0   1
 5   1505   2001       1505       2001  1505  2001         0         0   1
 6      0      0          0          0  9000     4         0         0   0

*Sentence Summary*
   Case          = Lower Case
   Translatable  = Yes
   Bold          = No
   Italic        = No
   Underlined    = No
   Single Quoted = No
   Double Quoted = No
```

EOS* **(= end of sentence)**

RES START **RES=resolution. There are two RES modules. In English these modules effect a macro parse of the sentence, resolving syntactic homographs and identifying clause transitions, chiefly.**

SWORK RECORDS **These records, below, are the SWORK records after dictionary look up which serve as input to RES1. *(This is the same as the SWORK array shown above)**

NOTE: Dictionary look-up, where possible, effects homograph resolution based on morphological clues. None took place in this case, however.)

NOTE: As previously noted, SWORK 4 ('is') and SWORK 5 ('blue') are syntactic homographs. It is the responsibility of the RES modules to resolve these homographs.

HOWEVER, it is important to note the following facts about RES and its process:

5. **The SP line in RES rules must have at a minimum two elements (two SWORKS).**
    - **The reason for ths is that RES must create a path or transition from one SWORK to the next. (This is not the case in the TRANs becauwe in the output of RES to the TRANs (TRAN1 to be more exact), the path through all the POS possibilities of a given sentence has already been established. Accordingly TRAN rules need not concern themselves with transitions from one element (SWORK) to another and thus can deal with a single element (SWORK) if that suits the purpose of the TRAN rule writer.**
6. **As you will have noted, the dictionary can have up to three parts of speech (POS) for a given word. (If a word has more than three parts of speech, and in English a number of words do, then it is the task of a TRAN rule to detect and supply the missing POS where necessary. TRAN does this via sends to SEMTAB.**
7. **It is very important to realize the following:**
    - **When a word has more than one POS in the dictionary, the most probable part of speech is placed first.**
        - **In the case of N/V homographs, the N is always placed first.**
    - **If RES does not explicitly select a POS for a homograph, then the system automatically selects the first (and most probably) POS provided by the dictionary as the correct POS for a given homograph in a given sentence.**
        - **This is the DEFAULT action of the system with respect of POS resolution, and RES relies upon it.**
    - **What this means is that RES need NOT explicitly account for every POS resolution, i.e. RES rules need only be written to cover exceptions to this system default.**
        - **Interestingly, given this, had there been no RES rules at all for this particular sentence, *The sky is blue*, the default action of the system would have correctly resolved the POS ambiguities in the sentence.**
            - **As may be seen, the two amhiguous words, *is* and *blue*, have "main verb" and "adjective" (respectively) as their default (most probably) POS. Thus, the default action of the system would have provided the correct macro-parse.**

- **To understand RES, it is important to realize that this default action is going on as a background understanding and assumption of the RES rule writer.**
  - **To repeat, then, the RES rule writer only writes rules to cover the (endless) instances where the default action (based on probability) would not apply in a given context.**

```
         xx  wc typ fr sbs sps pat stm schg com o2b o3b meaningID| wc typ fr sbs sps pat stm schg com o2b o3b
meaningID| wc typ fr sbs sps pat stm schg com o2b o3b meaningID|
  1 1__   -1  20   1  1   0   1   0   0    0 LOG    0   0        -1|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0         0|  1 bos
  2 1__    2  14   1  3 101   1   0   0    0 LOG    0   0     86452|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0         0|  1 the
  3 1__    3   1  29  1  29   9  18   1    0 LOG    0   0     80811|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0         0|  1 sky
  4 11_    4   2  60  3 886  11  13   4    0 LOG    0   0     52846| 12  60  3 886   1  13   4    0 LOG    0   0
52845|  0   0  0   0   0   0   0    0        0   0         0|  1 is
  5 11_    5   1  40 13 609   6  16   1    0 LOG    0   0     18142|  2  21  1 546   7 233   1    0 LOG    0   0
18143|  0   0  0   0   0   0   0    0        0   0         0|  1 blue
  6 1__    6  20  10  1  10  10   0   0    0 LOG    0   1        0|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0         0|  0 .
```

```
   WSTRNG             HENUM        root      HASHCOD      root
    bos                 -1   -1     0    0       0    0     0    0
    the               8302    5     0    0       0    0     0    0
    sky               8490    4     0    0    8490    4  8490    4
    is                8030    0     0    0       0    0     0    0
    blue              1505 2001     0    0    1505 2001  1505 2001
    .                 9000    4     0    0       0    0     0    0
```

WSTRNG=literal input word string

RES1 START **(NOTE: If the Reader has not already done so, he or she is advised to consult the RES primer before proceeding further)**

The two RES modules are quite complicated, far more so than the TRAN modules. Their function is to resolve syntactic homographs and recognize clausal transitions (effecting a top-down, deterministic macroparse of the sentence). In effect, a single path is created through the multiple nodes of the SWORK array, if any multinoded (i.e. ambiguous)entries exist. (As stated earlier, an SWORK for a given term may have up to three parts of speech (i.e. only three nodes). RES modules eliminate all but one node for any multiple node SWORK, in effect creating a deterministic parse of the sentence.

RES1 is essentially just preparatory to RES2 where most of thee resolution and macroparse work is done. I will keep commentary on the RES rules to a minimum chiefly because of their complexity but also partly because of documentationm inadequacies and memory lapses.

```
*SWORK RECORDS*
         xx  wc typ fr sbs sps pat stm schg com o2b o3b meaningID| wc typ fr sbs sps pat stm schg com o2b o3b
meaningID| wc typ fr sbs sps pat stm schg com o2b o3b meaningID|
  1 1__  -1  20   1  1   0   1   0   0    0 LOG    0   0        -1|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0          0|  1 bos
  2 1__   2  14   1  3 101   1   0   0    0 LOG    0   0     86452|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0          0|  1 the
  3 1__   3   1  29  1  29   9  18   1    0 LOG    0   0     80811|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0          0|  1 sky
  4 11_   4   2  60  3 886  11  13   4  847 LOG    0   0     52846| 12  60  3 886   1  13   4    0 LOG    0   0
52845|  0   0  0   0   0   0   0    0        0   0          0|  1 is
  5 11_   5   1  40 13 609   6  16   1    0 LOG    0   0     18142|  2  21  1 546   7 233   1    0 LOG    0   0
18143|  0   0  0   0   0   0   0    0        0   0          0|  1 blue
  6 1__   6  20  10  1  10  10   0   0    0 LOG    0   1        0|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0          0|  0 .
```

Let us begin with some observations about Logos rules in general, whether in RES or the TRANs.

Rules have three major components:

1- Comment line (a clea, telegraphic English statement of the source language pattern that the rule is dealing with, and some indication of the action talem. The next to last things on the comment line are the initials of the linguist(s) and the date. The very last thing is the RES or TRAN module to which the rule belongs (not always present)

2- SP line (semantico-syntactic pattern). In effect, a string of SWORKS, up to ten maximum.

- **a. The SWORK of course has three parts: Word Class field, SAL Type field, and Form field.**
  - **i. If the rule is to apply to more than one SAL type, Linguist can specify din the type field a so-called tag set of SAL codes in place of a single code (see d. below).**
- **b. One of the SWORKs can be a so-called stretch element, which functions like a Kleene Star, thus allowing an SP line to encompass many more than ten SWORKS of the input array. The linguist can indiciate that the stretch can only be over certain specified POS classes, expressed by various negative numbers in the wc field of the stretch element.**
  - **i. WC fields that begin with, e.g., 55, indicate that this is a Stretch element.**
  - **ii. An SP line can have up to three Stretch elements.**
- **c. A minus 1 (-1) in the wc field of an SWORK element in the SP line stands for universal set (ie. any wc)**
- **d. A minus 2 (-2) in the type field of an SWORK element in the SP line indicates a tagset, which appears directly below the SP line.**
  - **i. The tag set can be a set of SAL codes, or an instruction, or both.**
  - **ii. Tag sets can be attached to any and all of the SWORKs in the SP line.**
- **e. A -1 in the form field of an SWORK element in the SP line stands for universal set (i.e. any value)**
  - **i. Note that the form field is used in a variety of ways to characterize the element. This is true both for inflectable elements where form has a real morphological purpose, and for uninflectable elements, where the form field has no morphological meaning but can be used to serve other purposes.**

3- VTR (vector transform). This is the action part of the rule, appearing immediately below any tagset lines.

- **a. In RES, the VTR action pertains solely to source analyhsis.**
- **b. In the TRANs, VTR's pertain to both source and target actions.**
- **c. Except where noted, VTR's in both RES and TRANs contain such things as:**

i. Relational pointers (primarily in VTR's of TRAN ruales). Relational pointers point to elements in the SP line. Relational points in the VTR are expressed as a negative number from -1 to -10.

ii. Switches, which are programmed operators or functions (subroutines). Switches are identified by a negative number ranging from -11 to -98 (not all numbers are used, in some cases because the original switch of that number has been discarded.)

1. Some switches pertain to source actions, others to target.
2. All target action is performed exclusively by the-63 switch whicn points to a specific, so-called 30-table. This 30-table is unique to the particular TRAN rule.
   a. The 30-table effects the target actions relative to a given SP line of a given TRAN rule (i.e., the 30-table is essentially an exclusively target VTR within the main VTR).
   b. The 30-table can contain one or more -64 switchs which in turn invokes a 40-table. Where 30-tables are rule specific, 40—tables are shareable target functions. A given 40-table can be invoked by any number of 30-tables.
3. In effect, then all the target actions in the VTR line of a rule are all contained in the 30 table. The 30-table will typically contain one of more -64 switches pointing to a 40-table.
   a. 40-tables themselves can contain -64 switches pointing to another (i.e. nested) 40-table.
      i. There is no limit to this nesting.
4. To summarize, the main VTR of a rule, if there is a target action to be performed, accomplishes that target action in a 30-table unique to that particular rule.
   a. 30-tables are invoked by -63 switches
   b. The 30-table may invoke one or more of the shared, general purpose 40-tables, via a -64 switch.
   c. The -63 and -64 switches and their associated 30- and 40-tables, and any other target-specific swsitches, will be explained in greater detail when we come to the TRANs, below.
5. Switches themselves often contain relational pointers. Within a switch, relational pointers are expressed by negative numbers ranging from -81 to -90.
   a. Reminder: relational points in the VTR (outside of switches) are expressed by negative numbers from -1 to -10.
      i. Relational pointers in VTR's come with a single parameter. This will become apparent when we get into the TRANs.

iii. Slots (in TRAN VTR's only). This will be explained when we deal with actual TRAN rules.

iv. VC's (in TRAN VTR;s only). Ditto.

v. Constants. Constants are numbers pointing to literal target words in a special target dictionary or word list.

1. constants range from 121 to 998. Any number in a VTR in this range, followinged by a single parameter, may be construed as a target constant.

d. 999 marks the end of a VTR.

```
      6300 RULE AT  1
Res1 rule #3775, ID: 3450
* new rule BOS EL (STRETCH 01 795 11) EP199
32  (20 1 1)  (-1 -2 -1)  spec: 8
```

```
{ 6012   81  900 9000 6500   81 8000 8052    0    0    0    0    0    0    0    0    0    0    0 }
 -31  56 -41   2 999
 6500 TAGSET STARTING AT ELEMENT    2
```

1- Comment Line. The comment line always begins with an indication of the length of the SP line (i.e. how many SWORKs are involved).

a. In RES only, however, this initial number is used to prioritize rule application, ranging from 3 to null. In above, the 32 signifies a highest level rule type, of length 2. (I have no recollection what spec:8 signifies. A programmer's notation, not a linguist's.)

2- SP Line. The -1 in the wc field of the second element of the SP line means universal (any) wc.

a. The -2 in type field of the second element of the SP line points to the tagset immediately below. Tagsets in RES beginning with 60nn invoke special functions. Type of function to be performed is indicated by the nn value of the 6000 operation. 9000 marks the end of this function.

3- Constraint line:

a. The Constrain line provides conditions that must be satisfied for the rule to match. This line was originally called a tagset line, and was originally designed to allow for the expansion of a single TYP code in a given SWORK to a set of such codes

b. In the tagset above, 6012 is a function that instructs the system to exclude matches on certain SAL values: The first parameter, 0081 (a relational pointer) specifies the element of the SP line to which this exclusion is to apply (i.e., the first element. The values between the first parameter and 9000 (NOTE: 9000 is end of function marker) specify the SAL codes that are to be excluded in the match. (In this cane only one, viz., 900).

c. The 6500 function specifies certain conditions that must cause this particular RES rule to fail (i.e. be skipped). There are scores of such conditions which will not be gone into here. I have no documentation on the meaning of or function associated with parameters 8000 or 8052.

4- VTR line:

a. -31 56: Instructs system not to match this rule again in the present sentence

b. -41002: Instructs system to backspace the number of slements specified by the parameter (parameter value of 2 in this case) when resuming match function after this rule has been executed. The -41 switch is most important in both RES and TRANs as it allows the system to re-match on the same segment of the input SWORK array, the idea being that if the current rule has changed some SWORK in the SP line, the rematch will march on a different rule. Thjis is a very powerful means for the linguist to direct traffic so to speak (take control of rule matching to whatever extent zhe wishes) (

i. Note that the --31 56 switch above has a related effect without changing any SWORK value.)

```
MATCH AT  1
Res1 rule #3774, ID: 3342
BOS EL = -2 /SET SUBSET OF BOS=900 S389 ER1
32  (20 1 1)  (-1 -2 -1)  spec: 7
  { 6012   81  900  909  385  186  185  186  188  189 9000    0    0    0    0    0    0    0    0    0    0 }
   -46 -81   0 900   0 -31  56 -41   2 999
```

6012 in Constraint line tests that -81 has none of the subset values that follow (9000 is function delimiter)
-46 switch in VTR line alters SWORK values (in RES this effect is temporary and does not affect SWORKs entering TRAN). Switch has four parameters. Fist param is a relational pointer. Params two through four specify SAL changes to wc, type, and form. In the example, type is changed to 900.
-31 56 instructs system not to match this rule again (in the next matching operation)

```
MATCH AT  2
```

```
Res1 rule #3057, ID: 2771
* DET N EL = -3 /TO AVOID PV SET IMP BS385 ERES1
13  (14 -1 -1)  (1 -1 -1)  (-1 -2 -1)  spec: 4
  { 6025   81 6014   82   12   14    6 9000 6012   82  851  849  848  900 9000    0    0    0    0    0    0 }
   -31  56 -13 -82 -41   3 999
```
-13 with its relational pointer locks the SWORK in RES to this POS, such that no changes can be made to it.

```
MATCH AT  2
Res1 rule #2764, ID: 2497
* THE N(SING/PL) BS1184 ERES1
 2  (14 101 3)  (1 -2 57)  spec: 8
  { 6012   82  849  851  900 9000 6014   82   16 9000    0    0    0    0    0    0    0    0    0    0    0 }
   -13   1 999
```
Here the parameter 1 of the -13 switch causes ALL elements in the SP line top be locked to the matched POS

```
MATCH AT  4
Res1 rule #699, ID: 590
IS/ARE N(ADJ) EP284 ERES1
 2  (2 60 -1)  (1 -1 13)  spec: 5
   -13   1 999
```
The VTR here locks the first element to a verb and the seocn eloement to WC one (form 13 indicates this element is itself a N/ADJ homograph, to be later resolved by a TRAN rule).

POSSIBLE MATCH GOING TO RESSEM AT 4 RULE NO. 2334 **(NOTE: This rule has a higher priority (22 + 6) than the preceding rule (which has only priority 2) but will only be executed if the send to SEMTRES is successful. It is not in this case, so the V/AUJX ambiguity of "is" in this case is not resolved to AUX.**

```
Res1 rule #2334, ID: 2108
* +6 AUX V EP1084 ERES
22  (12 -1 -1)  (-9 -1 -1)  spec: 0
   -22 -81 -82   0   0   0 -46 -81   0 848   0 -13   1 999
*SEMWRK VALUES*
      12  60   3   4  12  60   3   4   2  21   1   5
 886   1  60 886   1  60 546   7  21

 NO MATCH IN RESSEM
```
The -22 switch sends the elements specified by the first two parameters (up to five are allowed) to SEMRES (SEMRES is the RES counterpart to SEMTAB in the TRANs). Like SEMTAB, SEMRES allows for more specific, nested tests. In this case, no match is made, and the VTR is not further executed. Had it been executed, the -46 switch would have set lthe TYP field of the AUX to 848 (temporary change for RES only), signifying to RES2 that this element has been resolved to AUX and is no longer ambiguous. As it stands now, RES2 will have to resolve this V/AUX ambiguity of "is".

In point of fact, all amgiguities in this sentence have now been resolved, (There were only two amgiguities, relating to "is" and "blue" and the last RES1 rujle, by virtue of its -13 1 switch, locked these elements to the matched upon POS, namely, V and ADJ/N.

RES2 START

SWORK RECORDS **(Showing the SWORK array after RES1)**

```
        xx  wc typ fr sbs sps pat stm schg com o2b o3b meaningID| wc typ fr sbs sps pat stm schg com o2b o3b
meaningID| wc typ fr sbs sps pat stm schg com o2b o3b meaningID|
  1 1__  -1  20   1  1 900   1   0   0    0 LOG    0   0       -1|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0        0   0         0|  1 bos
  2 1__   2  14   1  3 101   1   0   0    0 LOG    0   0    86452|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0        0   0         0|  1 the
  3 1__   3   1  29  1  29   9  18   1    0 LOG    0   0    80811|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0        0   0         0|  1 sky
  4 1__   4   2  60  3 886  11  13   4  847 LOG    0   0    52846| 12  60  3 886   1  13   4    0 LOG    0   0
52845|  0   0  0   0   0   0   0   0        0   0         0|  1 is
  5 1__   5   1  40 13 609   6  16   1    0 LOG    0   0    18142|  2  21  1 546   7 233   1    0 LOG    0   0
18143|  0   0  0   0   0   0   0   0        0   0         0|  1 blue
  6 1__   6  20  10  1  10  10   0   0    0 LOG    0   1       0|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0        0   0         0|  0 .
```

In the above SWORK array, note the items in blue. This is a three digit field, indicating which of the three possible POS for a given word is the resolved POS. In the present case, the 1 indicates that the first POS shown is the resolved POS. As it happens, all ambiguities have been resolved by RES1. RES2 has nothing to do.

POSSIBLE MATCH GOING TO RESSEM AT 2 RULE NO. 8358 (**Match is only "possible," depending on whether SEMRES match is successful or not)**

```
Res2 rule #8358, ID: 8177
+4 (XMV)(NUM=0) ART (U/PV2RT) N(SG) = -2 EP1085 OERES2
```
(XMV=expect main verb)

```
32  (14 -1 31)  (1 -2 44)  spec: 10
```
32 signifies highest priority (3 is priority code, 2 is length within that code)

```
  { 6050  103 8021    4 8000 9000 6014   82    6 9000 6355   82 8000    0    0    0    0    0    0    0    0 }
   -22 -81 -82   0   0   0 -46 -81   0   0  10 -31  56 -41   2 999
```

6050 in the Constraint line tests for value of 21 in cell 3, or cell 13 (cell 13 is tested only if cell 11=1, indicating a 'minor path' has been entered. Note: The initial '1' in the first parameter effects this double search).
6050 switch contnues to test cell 4 for zero. 9000 delimits the function.
6014 function excludes match on element 2 (1st parameter) if it WC is 6.
6355 looks to right for an unambiguous verb, or a verb marked by RES1 as PVI or PVT (possible verbs)
(See *RES Switch Summary* for fuller explanation of these Constraint line functions). NOTE below that this test 6355 is successful.

```
*SEMWRK VALUES*
      14   1   3   2  14   1   3   2   1  29   1   3
 101   1   1 101   1   1  29   9  29
```

*** **RES22 MATCH** **A Match in SEMRES is found (see below). SEMRES is the RES version of SEMTAB (Semantic Table, extensive use of which is made in the TRANs) SEMRES and SEMTAB are special purpose pattern dictionaries whose chief purpose is to support semantic disambiguation in the TRANs and POS ambiguities in RES. Both these dictionaries are invoked by a -22 switch (see below in TRAN1 for discussion of switches.) In RES, however, the -22 switch has an additional function: the rule sending the -22 switch send to SEMRES will fail if the SEMRES match fails.**

```
Res22 rule #6367, ID: 6313
THE N(TERMINAL) JP584 ERES22
 3  (14 101 1)  (14 101 -1)  (1 -1 42)
   999
 6300 RULE AT  2
```
NOTE: The first 14 101 1 (for "the") is the index to the rule in SEMRES)

```
Res2 rule #8358, ID: 8177
+4 (XMV)(NUM=0) ART (U/PV2RT) N(SG) = -2 EP1085 ORES2
32  (14 -1 31)  (1 -2 44)  spec: 10
  { 6050  103 8021    4 8000 9000 6014   82    6 9000 6355   82 8000    0    0    0    0    0    0    0    0 }
   -22 -81 -82   0   0   0 -46 -81   0   0  10 -31  56 -41   2 999
   6355 TAGSET STARTING AT ELEMENT    4
   UV OR PV FOUND AT   4

MATCH AT  2
Res2 rule #8358, ID: 8177
+4 (XMV)(NUM=0) ART (U/PV2RT) N(SG) = -2 EP1085 ORES2
32  (14 -1 31)  (1 -2 44)  spec: 10
  { 6050  103 8021    4 8000 9000 6014   82    6 9000 6355   82 8000    0    0    0    0    0    0    0    0 }
   -22 -81 -82   0   0   0 -46 -81   0   0  10 -31  56 -41   2 999


CELL VALUES     1   1   1   0   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                                    0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                                    0   0   0   0   0   0   0   0   0   0

POSSIBLE MATCH GOING TO RESSEM AT  2 RULE NO. 7628
Res2 rule #7628, ID: 7463
DET(NOT WC18) N EL(NOT N) = -2 EP889 ER2
 3  (14 -1 -1)  (1 -1 -1)  (-1 -2 -1)  spec: 4
  { 6014   83    1   16 9000 6014   81   18 9000    0    0    0    0    0    0    0    0    0    0    0    0 }
   -22 -81 -82   0   0   0 -41   2 999
*SEMWRK VALUES*
      14   1  10   2  14   1  10   2   1  29   1   3
 101   1   1 101   1   1  29   9  29

  *** RES22 MATCH
Res22 rule #6367, ID: 6313
THE N(TERMINAL) JP584 ERES22
 3  (14 101 1)  (14 101 -1)  (1 -1 42)
   999

MATCH AT  2
Res2 rule #7628, ID: 7463
DET(NOT WC18) N EL(NOT N) = -2 EP889 ER2
 3  (14 -1 -1)  (1 -1 -1)  (-1 -2 -1)  spec: 4
  { 6014   83    1   16 9000 6014   81   18 9000    0    0    0    0    0    0    0    0    0    0    0    0 }
   -22 -81 -82   0   0   0 -41   2 999
```

```
CELL VALUES      1   1   1   0   0   0   0   0   0   0          0   0   0   0   0   0   0   0   0   0
                 0   0   0   0   0   0   0   0   0   0          0   0   0   0   0   0   0   0   0   0
                                     0   0   0   0   0   0   0   0   0   0
                 0   0   0   0   0   0   0   0   0   0          0   0   0   0   0   0   0   0   0   0
                 0   0   0   0   0   0   0   0   0   0          0   0   0   0   0   0   0   0   0   0
                                     0   0   0   0   0   0   0   0   0   0

MATCH AT  3
Res2 rule #214, ID: 209
(NUM=0) (XV) N(NOT WC14/ING) EL = -2 /SET NUM S285 OERES2
12  (1 -1 42)  (-1 -2 -1)  spec: 4
  { 6050 1004 8000 1103 8021 9000 6014   81   14 9000 6012   81   15 9000    0    0    0    0    0    0    0 }
```

6014 tests that -81 is NOT wc 14
6012 tests that -81 does not have superset 15 (which would be an participial ING adjective form)

```
   -18   4   0 -81   2 -42 -81   0   0   0   0 -31  56 -41   2 999
```

-18 switch executes function 2, specified by the 4th parameter, (Function unfortunately is undocumented in the RES Switch Summary. Note that this function also appears to initiate a send to SEMRES).
Notice that 6050 in the Constraint line had already established that cell 4 was zero. The -18 switch has now causes cell 4 to read '1' (see CELL VALUES immediately below)

```
*SEMWRK VALUES*
      1  29   1   3   1  29   1   3
  29   9  29  29   9  29

 NO MATCH IN RESSEM
*SEMWRK VALUES*
      1  29   1   3   1  29   1   3
  29   9  29  29   9  29

 NO MATCH IN RESSEM


CELL VALUES      1   1   1   1   0   0   0   0   0   0          0   0   0   0   0   0   0   0   0   0
                 0   0   0   9  29  29   0   0   0   0          0   0   0   0   0   0   0   0   0   0
                                     0   0   0   0   0   0   0   0   0   0
                 0   0   0   0   0   0   0   0   0   0          0   0   0   0   0   0   0   0   0   0
                 0   0   0   0   0   0   0   0   0   0          0   0   0   0   0   0   0   0   0   0
                                     0   0   0   0   0   0   0   0   0   0

MATCH AT  3
Res2 rule #1726, ID: 1686
(XV)(NUM NOT = 0) N UV = -1 S786 OERES2
22  (1 -1 72)  (-9 -2 41)  spec: 5
  { 6050   11 8000  103 8021  204 8000 9000  844  845  846  847  848    0    0    0    0    0    0    0    0 }
   -17 183 -81   1 -18   4   0 -81   2 -46 -82   3   0   0 999
```

The 844-848 values here are TYP field (subset in this case) elaborations to the second SWORK's TYP field

```
CELL VALUES     1   1   1   1   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                0   0   0   9  29  29   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                                    0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                                    0   0   0   0   0   0   0   0   0   0

MATCH AT  4
Res2 rule #3421, ID: 3321
VI ADJ PUNC = -2 S285 OERES2
23  (3 11 41)  (1 -1 47)  (-6 -2 -1)  spec: 8
  { 6014   82   14 9000  888   20    3   10    7    0    0    0    0    0    0    0    0    0    0    0    0 }
888 etc are elaborations to the TYP field of the third element. These values are OR'd
   -18   3   0 -81   1 -18   5   0 -81   1 -18   6   4 -81   1 -41   2 999



*SEMWRK VALUES*
       3  60   3   4   3  60   3   4
 886  11  60 886  11  60

 NO MATCH IN RESSEM


CELL VALUES     1   1  60   1   3   4   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                0 886  11   9  29  29   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                                    0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                                    0   0   0   0   0   0   0   0   0   0

MATCH AT  5
Res2 rule #213, ID: 208
(XADJ C6=SF29) PRE-ADJ EL(NOT N) = -1 /RESET C6=90 S1091
12  (1 -1 30)  (-1 -2 -1)  spec: 4
  { 6050 1106 8029 9000 6014   82    1 9000    0    0    0    0    0    0    0    0    0    0    0    0    0 }
   -18   6  90 -81   1 999


CELL VALUES     1   1  60   1   3  90   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                0 886  11   9  29  29   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                                    0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0         0   0   0   0   0   0   0   0   0   0
                                    0   0   0   0   0   0   0   0   0   0
```

```
*RES END*

*SWORK RECORDS*
         xx  wc typ fr sbs sps pat stm schg com o2b o3b meaningID| wc typ fr sbs sps pat stm schg com o2b o3b
meaningID| wc typ fr sbs sps pat stm schg com o2b o3b meaningID|
  1 1__  -1  20   1  1   0   1   0   0    0 LOG    0   0        -1|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0         0|  1 bos
  2 1__   2  14   1  3 101   1   0   0    0 LOG    0   0     86452|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0         0|  1 the
  3 1__   3   1  29  1  29   9  18   1    0 LOG    0   0     80811|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0         0|  1 sky
  4 1__   4   2  60  3 886  11  13   4  847 LOG    0   0     52846| 12  60  3 886   1  13   4    0 LOG    0   0
52845|  0   0  0   0   0   0   0    0        0   0         0|  1 is
  5 1__   5   1  40 13 609   6  16   1    0 LOG    0   0     18142|  2  21  1 546   7 233   1    0 LOG    0   0
18143|  0   0  0   0   0   0   0    0        0   0         0|  1 blue
  6 1__   6  20  10  1  10  10   0   0    0 LOG    0   1        0|  0   0  0   0   0   0   0    0        0   0
0|  0   0  0   0   0   0   0   0    0        0   0         0|  0 .


      RESPAS   FOR ELEMENT #

*RESOLVED SWORK RECORDS*
                                                                  **** MAJOR PATH ***** -  *** MINOR  PATH
*****
                                                                  CL UV VB SJ VB VB CL N  PR DC - CL UV VB SJ VB VB CL N  PR
SP
         wc typ fr sbs sps patstm schg com o2b o3b| A     T  F  F  X  B  X     S  -  A     T  F  F  X  B  X
  1 1__  20   1  1   0   1   0  0    0 LOG    0   0|  1   1  1  0  0  0  0  0  0  0 -  0  0  0  0  0  0  0  0  0  1
bos
  2 1__  14   1  3 101   1   0  0    0 LOG    0   0|  1   1  1  0  0  0  0  0  0  0 -  0  0  0  0  0  0  0  0  0  1
the
  3 1__   1  29  1  29   9  18  1    0 LOG    0   0|  1   1  1  1  0  0  0  0  0  0 -  0  0  0  0  0  0  0  0  0  1
sky
  4 1__   2  60  3 886  11  13  4  847 LOG    0   0|  1   1 60  1  3  4  0  0  0  0 -  0  0  0  0  0  0  0  0  0  1
is
  5 1__   1  40 13 609   6  16  1    0 LOG    0   0|  1   1 60  1  3 90  0  0  0  0 -  0  0  0  0  0  0  0  0  0  1
blue
  6 1__  20  10  1  10  10   0  0    0 LOG    0   1|  1   1 60  1  3 90  0  0  0  0 -  0  0  0  0  0  0  0  0  0  0 .
```

NOTE: RES has resolved the syntactic (POS) ambiguity of ‘is’ to the main verb (wc2). Also the ambiguities associated with ‘blue’ have been partially resolved: blue is no longer seen as a verb. It remains ambiguous with respect to N/ADJ, signified by 13 in the form field of a wc1.

NOTE: An important function of RES2 (not illustrated in this simple sentence) is to recognize transitions from the main clause and dependent clauses (MAJOR PATH) to minor clauses (MINOR PATH— which latter are chiefly, but not exclusively, relative clauses.
NOTE: I do not recall at the moment the precise meaning of CL UV SJ VB etc. but I believe CL identifies the type of clause (1=main; 2-dependent) (don't remember at the moment the codes for minor path clauses—perhaps they are always 1); UV=1 identifies verbs that are unambiguously such; VB identifies the SAL set code of the clause's main predicate. (Notice that the verb set code (60) is carried over to all subsequent elements of the clause. This value also gets stored in the SCONs for these elements so that any element in a sentence can test for the clause it appears in, and whether the element occurs after a verb or not, and what the verb set type happens to be. These are sometimes useful data for both source analysis and target synthesis); SJ=1 indicates (I think) that the subject of the clause has occurred. (This would become clearer to me if I had hard copy diagnostics for longer sentences/ The second VB (VB=3)may pertainm to main verb morphology. I reget I cannot be more helpful. But this particular main and minor path diagnostic displays the macro parse effected by the RES modules.

SCON CELL ARRAY
SCON ARRAYS (SCON=Syntacto-morphological control). Associated with each SWORK is a 100-cell SCON CELL ARRAY used to store information leaned about the term as analysis proceeds, e.g,, the SWORK's complete SAL code, top-down parse information about the type of clause the SWSORK appears in, plus other data for both source and target purposes, some of it bult as analysis proceeds.(Documentation on individual cells and their values comprise an small volume and cannot be touched upon here.

1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	847	60	1	3	4	0	0	886	0	0	23	0	0	0	0	0	0	0	12
	0	0	0	0	0	1	60	0	3	0	0	0	0	0	0	0	0	0	0	0
5	1	0	60	1	3	90	0	0	886	0	0	2	0	0	0	0	0	0	0	2
	0	0	0	0	0	7	21	0	1	0	0	0	0	0	0	0	0	0	0	0
6	1	0	60	1	3	90	0	0	886	0	0	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EOS
tran1 START
1 22 bos the sky is blue .
TARG_CODES: ID= 86452 lang=1 MorC=1 CC=LOG ofl2a=0 ofl2b=1 ofl3a=0 ofl3b=1 pat= 94 Gender=1 WC=14
TARG_CODES: ID= 80811 lang=1 MorC=1 CC=LOG ofl2a=0 ofl2b=1 ofl3a=0 ofl3b=1 pat= 52 Gender=1 WC= 1

```
TARG_CODES: ID=  52846 lang=1 MorC=1 CC=LOG ofl2a=0 ofl2b=1 ofl3a=0 ofl3b=2 pat=  1 Gender=3 WC= 2
TARG_CODES: ID=  18142 lang=1 MorC=1 CC=LOG ofl2a=0 ofl2b=3 ofl3a=0 ofl3b=1 pat= 83 Gender=3 WC= 1

*-SWORK RECORDS*                                                                PAT          STEM    LVL                    OFL3I
OFL4I TYPSAV
  1   1 -1 -1  20   1  1   -1     0   0  0    0     0   0  0    0     0  0  0  0  0  0    1 BOS                0     1
0
  2   1 -1 -1  14 101  3    2     0   0  0    0     0   0  0    0     0  0  0  0  0  0    1 the                1     1
0
  3   1 -1 -1   1  29  1    3     0   0  0    0     0   0  0    0    18  0  0  1  0  0    1 sky                1     9
29
  4   1 -1 -1   2 886  3    4    12  60  3    4     0   0  0    0    13 13  0  4  4  0    1 is                 2    11
60
  5   1 -1 -1   1 609 13    5     2  21  1    5     0   0  0    0    16233  0  1  1  0    1 blue               1     6
40
  6   1 -1 -1  20  10  1    6     0   0  0    0     0   0  0    0     0  0  0  0  0  0    0 EOS                0    10
0

*SWORK RECORDS*
         xx  wc typ fr sbs sps patstm com o2b o3b| wc typ fr sbs sps patstm com o2b o3b| wc typ fr sbs sps patstm com
o2b o3b|
  1 1__   -1  20   1  1   0   1   0  0 LOG   0   0| 0   0  0   0   0   0  0       0   0| 0   0  0   0   0   0  0
0   0|  1 BOS
  2 1__    2  14 101  3 101   1   0  0 LOG   0   0| 0   0  0   0   0   0  0       0   0| 0   0  0   0   0   0  0
0   0|  1 the
  3 1__    3   1  29  1  29   9  18  1 LOG   0   0| 0   0  0   0   0   0  0       0   0| 0   0  0   0   0   0  0
0   0|  1 sky
  4 1__    4   2 886  3 886  11  13  4 LOG   0   0| 12  60  3 886   1  13  4 LOG   0   0| 0   0  0   0   0   0  0
0   0|  1 is
  5 1__    5   1 609 13 609   6  16  1 LOG   0   0|  2  21  1 546   7 233  1 LOG   0   0| 0   0  0   0   0   0  0
0   0|  1 blue
  6 1__    6  20  10  1  10  10   0  0 LOG   0   1| 0   0  0   0   0   0  0       0   0| 0   0  0   0   0   0  0
0   0|  0 EOS
```

THE TRATAB **Not certain what this acronym stands for (perhaps Translation Table??), but it appears this all refers to target data. Not sure what O2B amd O3B refers to. TGPN refers to target morphology table that pertains to the particular term. (Incidental Note: The O in the above acronyms should be a letter, not a number, and refers to "Overflow" There is a history behind these acronyms that need not be gone into now)**

```
                        SWKAD    WC  O2A  O2B  O3A  O3B  TGPN  TG25     TARGET WORD
  1  BOS                   -1     1    0    0    0    0     0     0        -unfound/number-
  2  the                86452     1    0    1    0    1    94     1        der
```

```
  3  sky                  80811     1    0    1    0    1    52     1        Himmel
□
  4  is                   52846     1    0    1    0    2     1     3        sein
□
  5  blue                 18142     1    0    3    0    1    83     3        Blau
□
  6  EOS                      0     1    0    0    0    0     0     0        -unfound/number-
□
```

TRAN1

```
***** A MATCH STARTING AT   1 LEVEL   1       ON ELEMENT  1jj              tran1
Tran rule #4161, ID: 4160
**809 BOS = -1 / T90,F02/CK FOR ? (ADD VC108 FOR ADV-S287) ST286 EGSP1
 1  (20 1 1)
   -42  10 809   1   1 -20   0 108   0 -55  19 -81  62 -55  70   1   0 -55  99   1   0 -46 -81   0   900   2 -41   1
999    0   0
```

1. Comment Line: this rule matches on a beginning of sentence rule, checks for interrogative, and backspaces.

2.

1. SP Line: 20 001 01 is unique SAL code for BOS

2. VTR:

1- -42 switch causes system to branch and continue matching on special nested WC 10 rules, looking to match on an WC10 rule whose sp line has its its first element 10 809 1. if match occurs at wc 10 rule, then vtr of that rule rather than that of the calling rule is executed

i. apparently no rule match was made on any WC10 rule so system continues with match on VTR of current rule.

2- -20 000 Switch. Supresses new SWORK formation by this TRAN rule

a. (Note: The -20 swotch has a single parameter which can specify several highly specizlised functions but is used rather infrequently.)

b. IMPORTANT: Unless a TRAN rules's VTR has a -20 switch surpressing this action, each TRAN rule will automatically cause a new SWORK (sometimes referred to as NSWORK) to be formed, based on the wc/type/form values of the last SWORK of the SP line after taking backspacing into consideration (or as otherwise modified by a -46 switch).

i. Of course, as stated, nny such SOWRK formation takes into account the actions of the -41 switch.

1. Thus an SP line with three SWORKS whose VTR has no -20 switch and which has a -41 001, i.e., backspaces one, will cause a new SWPRK to be formed based on the values of the second element in the SP line.

a. A -46 switch however can modify any of the wc/type/form values

ii. This new SWORK will be part of the SWORK output array of the current TRAN. which serves as input the the subsequent level of TRAN.

3- 108 000 This is a VC (variavle constant) that is now added to the output address array (OPADR).

a. Once a VC is loaded into the output address array, it can be addressed by switches for that purpose at any point, and can be loaded with elements, constants, or other VC's. Initially VC's are empty when initially loaded into the OPADR.

4- -55 switch sets values into a clause-level cell array.

a. -55 has three parameters.

b. There are two variants to the -55 switch. Unfortunately I do not have adequate documentation on the first variant of this switch.
   i. 1st variant (cf first instance of -55, above).
      1. 1st parameter specifies a cell in the clause-level cell array that is to be affected by this action.
      2. 2nd param is a relational pointer, some value associated with which is to be placed in specified cell.
      3. 3rd param specifies source of that value. Unfortunately I not have up-to-=date documentation on the meaning of 62
   ii. 2ndst variant (cf second and third instances of -55, above).
      1. 1st param specifies a cell, as in first variant.
      2. 2nd param specifies the value to be placed in the cell.
      3. 3nd param is always null

5- -46 switch re-labels the SAL type field and the form field of the first element of the SP line. It does not, in this case, affect the element's wc field.

6- -41 001. Backspacing switch. Parameter specifies number of SWORK elements to be backspaced when system resumes matching function (in this case a backspace of 1 SWORK of the SP line). As may be seen, -20 switches are frequently used in conjunction with the -41 switch, allowing for iterative processing of the same SWORK input segment.

a. Note: because of the backspace, the original SWORK is being matched on again, but now it has a different form field value and therefore will not match on the same TRAN rule but on a different rule.

REVIEW: Number ranges in VTR's are meaningful. Here is an overview:

1. -1 to -10 (with one param) = relational pointers
2. -11 to -98 (with variable number of param's) = switches
3. 70 to 98 (with one param) = slots
4. 101-120 = VC's
5. 121-998 = target constants

DEFINITIONS:

1. SP = semantico-syntactic pattern
2. Tagsets = supplemental SP lines that allow for expanded specification of SAL types, and also other functions.
3. VTR = vector transform
4. Relational pointer (e.g., -1) = pointer to an SWORK in the SP line (-1 points to first SWORK)
5. Switches = the means by which both source and target actions are taken
6. Slots = recepticals for receiving elements, switches, target constants, and VC's. Slots are loaded by switches and are unloaded when the slot appears in some subsequent VTR. Slots have single parameters.
7. VC = Variable constant. VC's are generally similar to slots and what is true of the loading and unloading of slots is also true of VCs.

8. Target Constants – numeric pointers to target string addresses. Useful for introducing additional target strings not provided by original dictionary look up, or for replacing transfers so provided.

```
***** A MATCH STARTING AT   1 LEVEL   1       ON ELEMENT  1jj              tran1
Tran rule #4075, ID: 4074
PUNC = PUNC ST1184 EGSP1
 1  (20 -1 -1)
    83   0  -1   0  84   0 999     0   0
```

1- Comment Line: The SWORK is matched and, by default action, a new SWORK is being formed with the same wc/type/form values of the old SWORK (done automatically as a by-product of the match, unless inhibited by the -20 switch or modified by the -46 switch).

2- SP line: Here we have a match on a single wc 20 element of any type or form field (wc 20 stands for any sort of punctuation, including parens etc.)

3- VTR:

a. 83 0 (0 is a satalite paramaeter). In VTR's, positive numbers from 70 to 99 followed by a single4 parameter are so-called slots into which relational pointers, constants, VC's and switches may have previously been loaded by means of swsitches, and which now at this point get unloaded. Unloading means that any relational pointers that were placed in 83 will now cause the target address associated with the pointed-to SWORK to be loaded into the OPADR.

i. Any target constants (numbers pointing to target strings) found in the slot are also loaded into the OPADR.

ii. If a slot contains switches, such switches would also be executed at this point.

iii. Thus it is possible to delay the execution of switches by this means. One advantage of this is that it allows delayed switch execution to benefit from any analysis that may have occurred subsequent to the moment when switch was loaded into the slot. Slots are positive numbers ranging from 71 to 98 (but I am not certain of this upper range).

iv. Slots and VC's are the principal tool for effecting target word order.

1. Slots are available for use in any way linguists see fit, in accordance with overall strategies.

b. -1 0 This is a relational pointer (with its parameter), pointing in this case to the first element in the SP line.

i. The appearance of a relational pointer in the VTR cause the target address of the pointed-to SWORK to be loaded into the OPADR.

ii. Target addresses are loaded into the OPADR in the order in which they occur in the VTR.

iii. relational pointers have a single parameter.

a. A positive number in this param from 101-998 identifies a target constant which is to replace the target string initially associated (in the dictionary) with the SWORK.

i. This is one of the main mechanisms for overriding dictionary-level transfers of source words.

b. In the current rule, above, the relational pointer's parameter is 0 (null), meaning that the original dictionary target string associate with the SWORK is loaded into the OPADR (not the string itself, of course, but a pointer to it).

c. Note that target constants are addresses of (pointers to) literal target words stored in a separate target language constant dictionary.

c. Slot 084 is now unloaded into the OPADR.
    i. The OPADR will now have target addresses of either the original dictionary-level transfer for the sosurce word, and/or a target language constant. Such target-language constants may be in addition to, or a replacement for, the target string originally provided for the SWORK (source word) in the main dictionary.
        1. As stated, all target addresses will appear in the OPADR in the order in which they were loaded.
d. 999 signals end of VTR

```
***** A MATCH STARTING AT   2 LEVEL   1       ON ELEMENT  2jj              tran1
Tran rule #3174, ID: 3173
DET = DET E1 ST282 MMT287
 1  (14 -1 -1)
   -20   0 -31  15
   -63   0 458   1 999     0   0
```

1- Comment Line: Comment line here is misleading and should read DET = (DET)
    i. The parens indicates that DET is not to receive its own new SWORK (because of the -20 switch)
    ii. Tnis is part of the process by which NP is formed.
2- SP Line: wc14=Determiner. This rule pertains to any type of Determiner, whether definite or indefinite (e.g., the, a, his, this, no, etc.)
3- VTR:
    i. -20 suppresses new SWORK formation
    ii. -31 is a utility switch that performs the function specified in the single parameter.
        1. param setting 15 causes a definite article flag (ARTDEF) to be set to value of 1.
            a. This ARTDEF setting will be tested later to determine whether to alter the form field of the head noun 'sky' to a 17, signifying a NP introduced with a definite article.
                i. It is interesting that NP will be identified this way, given that the SP line does not test for a definite article SAL code. I cannot say for sure but I believe the action taken here is a default action, taken because no other rule has been matched which would have prevented this action. The Logos system makes extensive use of default rules which infer a condition in the absence of anything else having happened to preclude the default action.
    iii. -63 switch. All target actions in the system are introduced by the -63 switch which points the system to a 30-table, here 30-table 0458.
        1. Each 30 table is linked to a particular TRAN rule
        2. Three parameters of the -63 switch: first two pertain to the 30-table number (e.g. 0458) Unfortunately I not recall purpose of the third param, or even if it has any. (I do not have an up-to-date switch manual at my disposal. However, if I am not mistaken, I believe OpenLogos provides an on-line switch manual. (I do not have the OpenLogos system at my disposal either)
        3. In the present case, the only function of 30-table 0458 is to call a 40-Table via the -64 switch.

```
-Main 30 table #458
   -64   0 115   0 999
```

    iv. -64 switch. This switch calls a 40-table (0115). In other respects it is like a -63 switch.

1. As stated previously, 40-tables are sharable and callable by any 30-table. Because of the efficiencies of this arrangement, most of the target work is done in 40-tables.

```
Main 40 table #115
   -56   1 199 299  42  74
   -57   1  83   0  73   0  71   0
   -57   2 -55  14   1   0 -55  31 -81   2 -55  33 -81  13
   -66 399  56 -81  60 140  60 -36  45 -81 -11  75 107 -81 -31  11
   -57   3 -31 -81
   -57   4 999
```

v. 40-table 0115: Note that both 30- and 40-tables are essentially target VTRs.

1. Some of the switches in these tables are unique to target functions, others are common to both source and target work.

2. -56 switch: This switch tests a cell in the clausal cell array and effects a branch within the VTR depending on results of test (TRUE or FALSE). The number of parameters varies, as follows:
   a. 1st param establishes the number of conditions to be tested, in this case only 1 (i.e. whether cell 42 has the value 74).
      i. Each test condition entails two parameters, one for the cell, one for the value.
      ii. In multiple test conditions, the test can be either OR or AND.
         1. OR is the default.
         2. AND is expressed by inserting 777 777 (signifying AND) between each of the cell/value test pairs.
   b. 2nd param identifies three points in the VTR to which system will branch, as follows
      i. 1st digit is branch point for FALSE (cell has the value being tested for)
      ii. 2nd digit is the end point for processing when true
      iii. 3rd digit is the re-enter point at which to continue processing
   c. 3rd param identifes the three points in the VTR for FALSE match test. The three digits of this parameter have the same significance as for TRUE
   d. the 9's in the second and third parameters stand for the 999's which end the VTR.
   e. Branch points are nomrlally identified by switch -57.
   f. In the example, above, cell 42 is being tested for a value of 74.
   g. See the bold-=faced code below for what happens in the present 40-table:

3. -66 switch: I do not have documenmtation on this switch but it is also a branching switch akin to a -56 switch where rather than testing a cell, an element's SCON is being tested for the specified value. Here, ANDing is expressed by 777 (rather than 777 777, as in -56sw)
4. -36 switch: utility switch. Do not have documentation on function 45 but believe it is related somehow to -31 switch where param is 15, which causes the ARTDEF cell to be set (turned on). Perhaps this switch sets a cell to the .
5. -11 switch: This switch causes the slot in identifies in its one parameter (075) to be loaded with all that follows in the VTR.

a. This slot and all that has been loaded into uit will get unloaded by a later rule (the rule for the head noun).

6. -31 switch is a utility switch (as seen earlier).
   a. I have no documentation on the function specified by param=11. .

SEE AFTER NEXT TRAN RULE FOR A SUMMARY OF -56 AND -66 SWITCHES

```
 CELL 42  =    0
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SCON( 60,-81) =    0
 -66 056 CONDITION AT  32,                              CONTINUE TO THE RIGHT
 BEFORE- SLOTAD,SLOTA2,SLDUMP   5   0   0 SLOT
     0
 AFTER- SLOTAD,SLOTA2,SLDUMP   5   0   0 SLOT
   107-10381   999     1

***** A MATCH STARTING AT   3 LEVEL   2       ON ELEMENT  3jj              tran1
Tran rule #549, ID: 548
N(91) NON-N = -2 / CK S3 STS586 EGSP1
 2  (1 -1 91)  (-3 -1 -1)
   -20   0
   -63   1 352   3 -36  56   0 -41   2 999     0   0
```

1. Comment Line: Match is on a noun plus a non-noun SWORK. Rule backspaces its entire length.
2. SP line:
   a. the 91 in the form field of the 1st element is a superform that includes a small set of form values.
      i. I do not have documentation available on what values this particular superform covers, but can infer that it must refer to values that are singular.
   b. The -3 in the wc field of the 2nd element denotes a small set of wc's that are to be excluded fronm the match.
      i. Here again, I do not have documentation about the exact nature of this set, except to infer that it excludes nouns, and any wc that relates to simple NP's.
3. VTR line:
   a. -20 0 switch suspends automatic new SWORK formation
   b. -63 switch: This switch invokes 30-table # 1352 for target action

```
Main 30 table #1352
   -55   3 -81 351
   -56   3 399  56   3  18   3   8   3  38
   -66 299  56 -81  45   9 777 -81   2 789  60
   -66 299  56 -81  45   9 777 -81   2 123  60
   -66 123  56 -81   3   9  60
   -66 299 399 -81   3   6  60
   -57   1 -54   1 -81   3   1 -48  13   3 -81 -54   1 -81   5   2
   -57   2 -54   1 -81   3   1 -48  13   3 -81 -54   1 -81   5   1
```

```
-57   3 999
```

i. **-55 switch: This switch tests cell 3 with (I believe) the original form field value of -81. (I believe this is the significance of 351)**

ii. **-56 switch: This is the conditional branching switch discussed previously.**
   1. **In the current example above, there are three tests (indicated by the 3 in the 1st param)**
      a. **All three tests check cell 3 for values of 18, 8, and 38.**
      b. **Diagostic below indicates cell 3 has value 1, so test is FALSE**
         i. **I don't recall (and have no documentation) of what 56 means in the FALSE branch field. It appears to mean to continue processing the VTR after the -56 switch.**

iii. **-66 switch: This too is a conditional branching switch, discussed previously.**
   1. **In the above example, scons 45 of the SWORK pointed to be -81 is tested for value 9 AND scon 2 vor value 789. l believe the 60 makes the end of this particular -66 switch.**
      a. **Since the test result is FALSE (as we can see from the diag below), the negative branch 56 is taken, signifying to branch to the end of this switch and continue processing the VTR (i.e. after the 60)**
   2. **The next -66 switch makes a similar test but in this case tests scon 2 for value 123.**
      a. **Test result here is also false. Branch is to point after switch (after 60)**
   3. **The final -66 switch: first param indicates branch points for TRUE, viz. 123, which means branch on TRUE to -57 1, exit at -57 2 and then resume at -57 3.**
      a. **Test result however is FALSE, which means go to end of -66 switch and continue. This happens to be end of VTR (999), so VTR processing stops.**
      b. **System now attempts to match on a new TRAN1 rule.**

iv. **-54 switch (not executed): This switch sets values into a scon cell.**
   1. **1st param indicates number of sconcell/value pairs**
   2. **2nd param points to the SWORIK whose scon is being set**
   3. **3rd param identifies the scon**
   4. **4th param specifies the value to be set**

```
CELL  3  =    1
CELL  3  =    1
CELL  3  =    1
-56 056 CONDITION AT  13,                          CONTINUE TO THE RIGHT
SCON( 45,-81) =    0
-66 056 CONDITION AT  22,                          CONTINUE TO THE RIGHT
SCON( 45,-81) =    0
-66 056 CONDITION AT  33,                          CONTINUE TO THE RIGHT
SCON(  3,-81) =    1
-66 056 CONDITION AT  40,                          CONTINUE TO THE RIGHT
SCON(  3,-81) =    1
-66 SWITCH TEST: CONDITION FALSE
BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99
```

REVIEW OF -56 AND -66 SWITCHES

1. -56 switch: conditional testing of clause-level cells

   a. first param specifies number of cells to be tested.
      i. Testing is OR testing, unless test pairs are separated by 777 777, in which case AND
   b. second param specifies branch points on TRUE: n1n2n3, where n1 is where TRUE processing begins, n2 is where it ends for this switch, and n3 is where VTR processing continues.
      i. Branch points are indicated by -57. E.g. if n1 has value of 2, then branch is to -57 2, etc.
   c. third param specifies branch points on FALSE.
      i. Branching is same as for b.
      ii. If 56, however, signifies that branch is to end of -56 switch where VTR processing resumes.
   d. Tests pairs are cell/value combinations, as many sets of these as specified by first param

2. -66 switch: conditional testing of one or more scons of an SWORK
      i. Testing is OR, unless test pairs are separated by 777, in which case AND
   a. end of -66 sw is indicated by 60
   b. first param specifies branch points on TRUE, as in b. above
   c. second param specifies branch points on FALSE, as in c. above
   d. third param specifies SWORK whose scon is to be tested.
   e. Test pairs are scon/value combinations, as many as appear before 60 (switch-end indicator)

```
***** A MATCH STARTING AT   3 LEVEL   1       ON ELEMENT  3jj              tran1
Tran rule #1, ID: 0
```

NOTE RULE BELOW IS A MOST IMPORTANT HEAD NOUN RULE WHERE ALMOST ALL SIMPLE NOUN PHRASES END AT, NO MATTER HOW MODIFIED. FOR THAT REASON THE TARGET COMPONENT OF THE VTR IS ESPECIALLY COMPLEX.

```
**107 N = N / CK FOR F20,39,PN E1 ST985 MMT287
 1  (1 -1 -1)
   -42  10 107   1   1
   -63   0 802   1 999     0   0
```

1. Comment line:
   a. **107 signifies that system will branch to set of nested rules, all beginning with wc10 in the first element of the SP line (this is merely an address, not a SAL value).
      i. In this case, branch is to 10 107 01
   b. N = N signifies that match is on a N and this causes (=) new SWORK to be formed.
      i. Certain checks are made for forms, 20, 39, and for PN (process nouns)
2. SP line As stated above, this is the head noun rule for any simple NP. The system matches on this rule by default (i.e. in the absence of any other rule pre-empting it (as, e.g., in the case of “blue”, below)).
3. VTR:
   a. -42 switch: this is the switch that causes branch to nested wc10 rules.
      i. Second param specifies the wc10 address (addness not a SAL code value)
      ii. Can’t recakk what 3rd and 4th param signify here (perhaps will become clearer as we proceed)
   b. -63 switch: this invokes target 30-table (#802)

Main 30 table #802

```
   -55   5 -81 351
   -56   1 199 399   5  50
   -57   1 -46 -81  19   0   0
   -22   1 -81   1  -1  57 -46 -81   1   0   0
   -56   1 299 399   2   1
   -57   2 -46 -81   1   0  57
   -57   3 -64   0 138   0 999
```

cell 5 is being set to the original form field value of -81 (believe this is right but not certain)
cell 5 is tested for value 50. Ressult is FALSE and branch is to -57 3
-57 1 (branch point) where -46 sw alters wc of -81, then process goes on to
-22 sw which sends -81 to SEMTAB (discussed later), then
new -56 sw which tests cell 2 for value 1
-57 2 (TRUE branch) executed -46 sw which alters wc and form of SWORK -81
-57 3 (FALSE branch) Process now invokes -64 switch

```
 CELL  5  =    1
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99
```

```
Main 40 table #138
   -64   0  40   0 -64   0 139   0 -64   0  41   0 999
```

c. 40-table 138 executes three -64 switches, which successively call 40-table 40, then 40-table 139, and finally 40-table 41.

```
Main 40 table #40
   -54   1  30  30 -81 -54   1  31  31 -81 -54   1  32  32 -81 -54   1  33  33 -81 -54   1  34  34 -81 -54   1  35  35
-81 -54   1  36  36 -81 -54   1  37  37 -81 -54   1  38  38 -81 -54   1  42  42 -81 -54   1  43  43 -81 999
```

d. -54 switch: This -54 switch is quite different from the -54 switch desribed in a previous TRAN rule.
i. I have no documentation on this variant but my recollection is that this variant -54 sw was designed as a means for placing cell values into scons so that branch switch -66 could test them (e.g., see -66 sw below which tests scon 31 for 0 etc.) How cell 31 got loaded previously is unclear at this point (possibly this was an automatic program function, not sure)
1. The first param identifies the -54 function
a. (don't know if there are other functions for this -54 sw variant)
2. the second param, in the first example above, tells the system to load cell 31 with the value found in scon 31 of SWORK -81, and so on with the subsequent -54 switches in this 40-table.
a. cell 31 contained the SAL subset code of the article "the", which is now loaded into scon 31 of -81 ("sky').
i. See the effect of this in the scon readout at the end of TRAN1.
SWITCH REVIEW: -54 1st parm=function; 2nd parm=cell; 3rd parm=scon 4th parm=SWORK pointed to whose scon is being loaded.

```
Main 40 table #139
   -64   0   7   0 -64   0  39   0 -64   0  45   0
   -66 940   6 -81  20  70 -81  20 171  60 -55   5 -81 351 -55   6 -81  11 -55   7 -81  13 -55   8 -81  14
```

-64 switches call 40-table 7 and then 40-tables 39 and 45
After invoking these 40-tables (see below), 40-table 139 then proceeds with remainder of its code
-66 sw tests scon 20 of -81 for value 70 OR 171. On TRUE branch, not sure what 940 means, perhaps branch is to end of this 40-table.
Not sure what 6 means in FALSE branch, but it appears to be askin to 56, i.e., causes processing to proceed to switches after the 60. -55 sw sets cell 5 to original form field value of -81 (I think)

The remaining series of -55 switches are setting cells (1st param) to values of specified scon (3rd param) of -81. (Note: I have incomplete (not entirely up-to-date) documentation on assigned meanings of scon cells and their values, too extensive to reproduce here.)

```
   -56   7 940  27   5  57 777 777   6  75 777 777   7  12 777 777   8   1
```

-56 sw AND-tests cells 5,6.7,8, Note that the 777 777 pairs are counted in the number of TEST pairs specified in the 1ast param (7). Not sure what the TRUE branch here (940) signifies. On FALSE, branch is to -57 2 and ends at -57 7.

```
   -66  56 199 -81  31   0 777 -81  32   0 777 -81  33   0 777 -81  35   0 777 -81  36   0  60
```

-66 sw: continue to next switch at right on TRUE branch (i.e. after 60). On FALSE, begin at -57 1, continue to 9(99), end at 9(99)

```
   -66 127 199 -81   2 206 777 -81  13   5  60
```

-66 sw: branch to -57 1 on TRUE, end at -57 2, resume at -57 7. On FALSE, branch to -57 1, continue to and end at 9(99) (i.e. end of this 40-table).

```
   -57   1 102   0  75   0 107   0  83   0  73   0 103   0 105   0 108   0 106   0  71   0  77   0 101   0
```

-57 1: The instructions following this branch point now begin to build the target NP. VC 102 is loaded into the OPADR. Slot 75 is emptied and any switches are executed, any relational pointers and any VC's are loaded into OPADR, etc. etc. <u>All of the elements loaded into the OPADR here will stand in front (to the left of) the actual head noun of the target NP, which gets loaded at a subsequent branch point (-57 3, -57 4, or -57 5)</u>

-57 2 Each of the following -56 switches AND-test cell values, cell 8 for 2 AND cell 45 for 57. A FALSE causes continued execution of the next switch to the right. A TRUE result branches to various -57 branch points. In effect, what is being tested is the nature of -81 as indicated by these clause-level cells.

```
   -56   3 348  56   8   2 777 777   5  57 -55   1 -81  20
   -56   3 678  56  19 864 777 777   1  79
   -56   3 458  56   5  57 777 777 408   0
   -56   3 348  56   5  50 777 777   8   0
   -56   3 348  56   5  57 777 777   8   0
   -66 568  56 -81   1  16 777 -81  20  79  60
      -66 tests scon 1 OF -81 for 16 (viz., is -81 wc 16 (arithmate) AND does scon 20 have value 79
   -66 458  56 -81   1  16  60
   -66 678  56 -81  62 851 -81  62 848 -81  62 849 -81   2 848 -81  20 848  60
   -66 568 458 -81  20  79  60
   -57   3 -54   1 -81   8   2 -54   1 -81  13  15 -25   0  -1   2 117   0 -54   1 -81   7   0
   -57   4 -25   0  -1   0 -54   1 -81   7   0
   -57   5 575   0 864   0 865   0 -25   0  -1   0 -54   1 -81   7   0
   -57   6 -64   0  14   2 -81  -1
   -57   7 -64   0   3   0
   -57   8 104   0  74   0 112   0  79   0  96   0 110   0 -55  70   0   0 -54   1 -81  20   1 999
```

```
Main 40 table #7
   -66 124  56 -81  20 101 -81  20 171  60
   -66 234  56 -81  20 315  60
   -56   1  56 499  31 315
   -66 499  56 -81  12   9  60
   -66 499  56 -81  13  15 -81  13   7 -81  13   4  60 -55   5 -81 351
```

```
   -56   1 499  56   5  70
   -66 499  56 -81  46 101  60
   -66 499  56 -81   2 733  60
   -66 499  56 -81  13  15 -81  13  16  60
   -66 399 499 -81   3   7 -81   3   8 -81   3   9  60
   -57   1 -40   0 -11  75 107 131 -31  11 -54   1 -81  46 101
   -57   2 -40   0 -11  75 107 532 -31  11 -54   1 -81  46 315
   -57   3 -36  45   3 -31  16 -48  13   9 -81 -44 -96 107 140   0 -54   1 -81  46 140
   -57   4 999
```

NOTE: all values in the rules involve triplets (dating back to use of IBM cards to write these rules). Thus, value 56 is written as 056. The diagmostic program sometimes expresses expresses it one way (56), sometimes another (056). The significance is the same. E.g. relational pointers in VTRs were written (by linguists) as -01. The diag prog normally displays this as -1.

```
 SCON( 20,-81) =    0
 SCON( 20,-81) =    0
 -66 056 CONDITION AT   7,                              CONTINUE TO THE RIGHT
 SCON( 20,-81) =    0
 -66 056 CONDITION AT  14,                              CONTINUE TO THE RIGHT
 CELL 31  =  101
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99

Main 40 table #39
   -56   3 199  56  37 140 777 777  31 101
   -56   3  56 399  30 140 777 777  31 101
   -57   1
   -66 235  56 -81   3   4 -81   3   5 -81   3   6  60
   -66 399  56 -81   3   7 -81   3   8 -81   3   9  60 -55   5 -81 351
   -56   1 399  56   5  10
   -56   1 235 399 505  71
   -57   2 -36  45   1 -44 -96 107 131   0 -44 -96 107 457 -97
   -57   3
   -56   1 499 599  31 338
   -57   4 -16 -81   2 -81   0 -81
   -57   5 999


 CELL 37  =    0
 CELL 31  =  101
 -56 056 CONDITION AT   9,                              CONTINUE TO THE RIGHT
 CELL 30  =    0
 CELL 31  =  101
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99
 CELL 31  =  101
```

```
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  5   EXECUTE UNTIL -57 99  JUMP -57 99

Main 40 table #45
   -66  56 299 -81  57  29 -81  57 308 -81  57 129 -81  57 307 -81  57 166 777 -81  13   9  60
   -66 199 299 -81  62 864  60
   -57   1 -11  71 878   0 -31  11
   -57   2 999


 SCON( 57,-81) =   18
 SCON( 57,-81) =   18
 SCON( 57,-81) =   18
 SCON( 57,-81) =   18
 SCON( 57,-81) =   18
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SCON( 20,-81) =    0
 SCON( 20,-81) =    0
 -66 CONDITION FALSE, CONTINUE THIS VTR
 CELL  5  =    1
 CELL  6  =   29
 CELL  7  =    9
 CELL  8  =    0
 -56 CONDITION FALSE, CONTINUE THIS VTR
 SCON( 31,-81) =  101
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  1   EXECUTE UNTIL -57 99  JUMP -57 99
 CELL  8  =    0
 CELL  5  =    1
 -56 056 CONDITION AT 127,                           CONTINUE TO THE RIGHT
 CELL 19  =    0
 CELL  1  =    0
 -56 056 CONDITION AT 141,                           CONTINUE TO THE RIGHT
 CELL  5  =    1
 CELL  8  =    0
 -56 056 CONDITION AT 151,                           CONTINUE TO THE RIGHT
 CELL  5  =    1
 CELL  8  =    0
 -56 056 CONDITION AT 161,                           CONTINUE TO THE RIGHT
 CELL  5  =    1
 CELL  8  =    0
 -56 056 CONDITION AT 171,                           CONTINUE TO THE RIGHT
 SCON(  1,-81) =    1
 -66 056 CONDITION AT 180,                           CONTINUE TO THE RIGHT
 SCON(  1,-81) =    1
 -66 056 CONDITION AT 187,                           CONTINUE TO THE RIGHT
```

```
 SCON( 62,-81) =    0
 SCON( 62,-81) =    0
 SCON( 62,-81) =    0
 SCON(  2,-81) =   29
 SCON( 20,-81) =    0
 -66 056 CONDITION AT 206,                              CONTINUE TO THE RIGHT
 SCON( 20,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57  5  JUMP -57  8

Main 40 table #41
   -55  14   0   0 -55  15   0   0 -55  30   0   0 -55  31   0   0 -55  32   0   0 -55  33   0   0 -55  34   0   0 -55
35   0   0 -55  36   0   0 -55  37   0   0 -55  38   0   0 -55  42   0   0 -55  43   0   0 -55  30   0   0 999



***** A MATCH STARTING AT   4 LEVEL   1       ON ELEMENT  4jj              tran1
Tran rule #1119, ID: 1118
**156 V = V E1 ST286 BES1287 T798
 1  (2 -1 -1)
   -42  10 156   1   1 -55  11 -81  11 -55  13 -81  13 -55  48 -81   2
   -66 123 299 -81  31 460 -81    31 461  60
   -57   1 -55  22  17   0
   -57   2 -55  22   0   0
   -57   3
   -63   1  53   1 999     0   0
 SCON( 31,-81) =    0
 SCON( 31,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99

Main 30 table #1053
   -54   4 -81  17   1  18   1  19   1  20   1 -55   5 -81 351
   -56   1 299 129   5  71
   -57   1 113   0  83   0 118   0 111   0 115   0  90   0 120   0 114   0 116   0 -31  21 -21   0  -1   0  76   0 109
0 117   0 110   0
   -57   2 113   0  83   0 118   0 111   0 115   0  90   0 120   0 114   0 116   0 -31  21 -38   0  -1   0  76   0 109
0 117   0 110   0 999


 CELL  5  =    3
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  1   EXECUTE UNTIL -57  2  JUMP -57 99

***** A MATCH STARTING AT   5 LEVEL   2       ON ELEMENT  5jj              tran1
Tran rule #549, ID: 548
N(91) NON-N = -2 / CK S3 STS586 EGSP1
```

```
 2  (1 -1 91)  (-3 -1 -1)
   -20   0
   -63   1 352   3 -36  56   0 -41   2 999     0   0

Main 30 table #1352
   -55   3 -81 351
   -56   3 399  56   3  18   3   8   3  38
   -66 299  56 -81  45   9 777 -81   2 789  60
   -66 299  56 -81  45   9 777 -81   2 123  60
   -66 123  56 -81   3   9  60
   -66 299 399 -81   3   6  60
   -57   1 -54   1 -81   3   1 -48  13   3 -81 -54   1 -81   5   2
   -57   2 -54   1 -81   3   1 -48  13   3 -81 -54   1 -81   5   1
   -57   3 999


 CELL  3  =   13
 CELL  3  =   13
 CELL  3  =   13
 -56 056 CONDITION AT  13,                              CONTINUE TO THE RIGHT
 SCON( 45,-81) =    0
 -66 056 CONDITION AT  22,                              CONTINUE TO THE RIGHT
 SCON( 45,-81) =    0
 -66 056 CONDITION AT  33,                              CONTINUE TO THE RIGHT
 SCON(  3,-81) =    3
 -66 056 CONDITION AT  40,                              CONTINUE TO THE RIGHT
 SCON(  3,-81) =    3
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99

***** A MATCH STARTING AT   5 LEVEL   1       ON ELEMENT  5jj              tran1
Tran rule #1, ID: 0
**107 N = N / CK FOR F20,39,PN E1 ST985 MMT287
 1  (1 -1 -1)
   -42  10 107   1   1
   -63   0 802   1 999     0   0

Main 30 table #802
   -55   5 -81 351
   -56   1 199 399   5  50
   -57   1 -46 -81  19   0   0
   -22   1 -81   1  -1  57 -46 -81   1   0   0
   -56   1 299 399   2   1
   -57   2 -46 -81   1   0  57
   -57   3 -64   0 138   0 999
```

```
 CELL  5  =   13
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99

Main 40 table #138
   -64   0  40   0 -64   0 139   0 -64   0  41   0 999



Main 40 table #40
   -54   1  30  30 -81 -54   1  31  31 -81 -54   1  32  32 -81 -54   1  33  33 -81 -54   1  34  34 -81 -54   1  35  35
-81 -54   1  36  36 -81 -54   1  37  37 -81 -54   1  38  38 -81 -54   1  42  42 -81 -54   1  43  43 -81 999



Main 40 table #139
   -64   0   7   0 -64   0  39   0 -64   0  45   0
   -66 940   6 -81  20  70 -81  20 171  60 -55   5 -81 351 -55   6 -81  11 -55   7 -81  13 -55   8 -81  14
   -56   7 940  27   5  57 777 777   6  75 777 777   7  12 777 777   8   1
   -66  56 199 -81  31   0 777 -81  32   0 777 -81  33   0 777 -81  35   0 777 -81  36   0  60
   -66 127 199 -81   2 206 777 -81  13   5  60
   -57   1 102   0  75   0 107   0  83   0  73   0 103   0 105   0 108   0 106   0  71   0  77   0 101   0
   -57   2
   -56   3 348  56   8   2 777 777   5  57 -55   1 -81  20
   -56   3 678  56  19 864 777 777   1  79
   -56   3 458  56   5  57 777 777 408   0
   -56   3 348  56   5  50 777 777   8   0
   -56   3 348  56   5  57 777 777   8   0
   -66 568  56 -81   1  16 777 -81  20  79  60
   -66 458  56 -81   1  16  60
   -66 678  56 -81  62 851 -81  62 848 -81  62 849 -81   2 848 -81  20 848  60
   -66 568 458 -81  20  79  60
   -57   3 -54   1 -81   8   2 -54   1 -81  13  15 -25   0  -1   2 117   0 -54   1 -81   7   0
   -57   4 -25   0  -1   0 -54   1 -81   7   0
   -57   5 575   0 864   0 865   0 -25   0  -1   0 -54   1 -81   7   0
   -57   6 -64   0  14   2 -81  -1
   -57   7 -64   0   3   0
   -57   8 104   0  74   0 112   0  79   0  96   0 110   0 -55  70   0   0 -54   1 -81  20   1 999



Main 40 table #7
   -66 124  56 -81  20 101 -81  20 171  60
   -66 234  56 -81  20 315  60
   -56   1  56 499  31 315
   -66 499  56 -81  12   9  60
   -66 499  56 -81  13  15 -81  13   7 -81  13   4  60 -55   5 -81 351
```

```
   -56   1 499  56   5  70
   -66 499  56 -81  46 101  60
   -66 499  56 -81   2 733  60
   -66 499  56 -81  13  15 -81  13  16  60
   -66 399 499 -81   3   7 -81   3   8 -81   3   9  60
   -57   1 -40   0 -11  75 107 131 -31  11 -54   1 -81  46 101
   -57   2 -40   0 -11  75 107 532 -31  11 -54   1 -81  46 315
   -57   3 -36  45   3 -31  16 -48  13   9 -81 -44 -96 107 140   0 -54   1 -81  46 140
   -57   4 999


 SCON( 20,-81) =    0
 SCON( 20,-81) =    0
 -66 056 CONDITION AT   7,                              CONTINUE TO THE RIGHT
 SCON( 20,-81) =    0
 -66 056 CONDITION AT  14,                              CONTINUE TO THE RIGHT
 CELL 31  =    0
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99

Main 40 table #39
   -56   3 199  56  37 140 777 777  31 101
   -56   3  56 399  30 140 777 777  31 101
   -57   1
   -66 235  56 -81   3   4 -81   3   5 -81   3   6  60
   -66 399  56 -81   3   7 -81   3   8 -81   3   9  60 -55   5 -81 351
   -56   1 399  56   5  10
   -56   1 235 399 505  71
   -57   2 -36  45   1 -44 -96 107 131   0 -44 -96 107 457 -97
   -57   3
   -56   1 499 599  31 338
   -57   4 -16 -81   2 -81   0 -81
   -57   5 999


 CELL 37  =    0
 CELL 31  =    0
 -56 056 CONDITION AT   9,                              CONTINUE TO THE RIGHT
 CELL 30  =    0
 CELL 31  =    0
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99
 CELL 31  =    0
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  5   EXECUTE UNTIL -57 99  JUMP -57 99

Main 40 table #45
```

```
  -66  56 299 -81  57  29 -81  57 308 -81  57 129 -81  57 307 -81  57 166 777 -81  13   9  60
  -66 199 299 -81  62 864  60
  -57   1 -11  71 878   0 -31  11
  -57   2 999


SCON( 57,-81) =   16
SCON( 57,-81) =   16
SCON( 57,-81) =   16
SCON( 57,-81) =   16
SCON( 57,-81) =   16
-66 SWITCH TEST: CONDITION FALSE
BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
SCON( 20,-81) =    0
SCON( 20,-81) =    0
-66 CONDITION FALSE, CONTINUE THIS VTR
CELL  5  =   13
CELL  6  =   40
CELL  7  =    6
CELL  8  =    0
-56 CONDITION FALSE, CONTINUE THIS VTR
SCON( 31,-81) =    0
SCON( 32,-81) =    0
SCON( 33,-81) =    0
SCON( 35,-81) =    0
SCON( 36,-81) =    0
-66 056 CONDITION AT  76,                            CONTINUE TO THE RIGHT
SCON(  2,-81) =  609
-66 SWITCH TEST: CONDITION FALSE
BRANCH TO -57  1   EXECUTE UNTIL -57 99  JUMP -57 99
CELL  8  =    0
CELL  5  =   13
-56 056 CONDITION AT 127,                            CONTINUE TO THE RIGHT
CELL 19  =    0
CELL  1  =    0
-56 056 CONDITION AT 141,                            CONTINUE TO THE RIGHT
CELL  5  =   13
CELL  8  =    0
-56 056 CONDITION AT 151,                            CONTINUE TO THE RIGHT
CELL  5  =   13
CELL  8  =    0
-56 056 CONDITION AT 161,                            CONTINUE TO THE RIGHT
CELL  5  =   13
CELL  8  =    0
-56 056 CONDITION AT 171,                            CONTINUE TO THE RIGHT
SCON(  1,-81) =    1
-66 056 CONDITION AT 180,                            CONTINUE TO THE RIGHT
```

```
 SCON(  1,-81) =    1
 -66 056 CONDITION AT 187,                              CONTINUE TO THE RIGHT
 SCON( 62,-81) =    0
 SCON( 62,-81) =    0
 SCON( 62,-81) =    0
 SCON(  2,-81) =  609
 SCON( 20,-81) =    0
 -66 056 CONDITION AT 206,                              CONTINUE TO THE RIGHT
 SCON( 20,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57  5  JUMP -57  8

Main 40 table #41
   -55  14   0   0 -55  15   0   0 -55  30   0   0 -55  31   0   0 -55  32   0   0 -55  33   0   0 -55  34   0   0 -55
35   0   0 -55  36   0   0 -55  37   0   0 -55  38   0   0 -55  42   0   0 -55  43   0   0 -55  30   0   0 999


***** A MATCH STARTING AT   6 LEVEL   1       ON ELEMENT  6jj              tran1
Tran rule #4190, ID: 4189
EOS = EOS /ADDED 091 SEPT/99 PTGTRG STS
 1  (20 10 1)
    83   0  84   0  91   0  -1   0 -55  64   0   0 -55  66   0   0 999     0   0


 -----------     tran1  PROCESSING COMPLETE      ----------
 THE SCON FOR tran1
```

1. **Each SWORK has a SCON. SCONs supplement the information about each element. Every element in the output address array for the target (OPADR) will have a SCON. (See Archives website for SCON meanings and values.)**

2. **The SCON array immediately below shows current SCON settings after TRAN1 processing. Note that the SCON ARRAY has 100 cells.**

3. **Note that SCON 5 for 'blue' contains the SAL codes for the main verb of the clause ('is'). This information in SCON 5 indicates that the term 'blue' appears <u>after</u> the main verb, a fact sometimes needed for analysis.**

    a. **Beginning with SCON 27 you will see a series of SCON arrays with numbers like 108 and 102 etc., appearing in the first cell. These particular SCONS and Numbers relate to something called "variable constants" or VC's. VC'se created during analysis, chiefly but not exclusively in TRAN1. VC's are extremely critical to both the target synthesis function, and will be explained in TRAN commentary, below.**

```
 1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
```

1	**20**	**1**	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
2	**14**	**101**	1	1	0	3	1	0	0	2	0	1	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	94	0
	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
3	1	29	1	1	0	3	1	0	0	3	29	1	9	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	**101**	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	18	0	52	0
	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
4	**2**	**886**	1	0	0	3	0	0	0	4	60	2	11	0	0	0	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	13	0	1	0
	1	847	60	1	3	4	0	0	886	0	0	23	0	0	0	0	0	0	0	12
	0	0	0	0	0	1	60	0	3	0	0	0	0	0	0	0	0	0	0	0
5	**1**	**609**	9	3	0	3	1	0	0	5	40	1	6	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	16	0	83	0
	1	0	60	1	3	90	0	0	886	0	0	2	0	0	0	0	0	0	0	2
	0	0	0	0	0	7	21	0	1	0	0	0	0	0	0	0	0	0	0	0
6	**20**	**10**	0	0	0	0	0	0	0	6	0	0	10	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	60	1	3	90	0	0	886	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	**108**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
    28    102    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    29    107    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    30    103    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    31    105    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    32    108    0    1    1    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    33    106    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    34    101    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    35    104    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    36    112    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    37    110    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    38    113    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    39    118    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    40    111    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    41    115    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    42    120    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    43    114    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    44    116    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    45    109    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    46    117    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    47    110    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    48    102    0    9    3    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    49    107    0    9    3    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    50    103    0    9    3    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    51    105    0    9    3    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    52    108    0    9    3    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
    53    106    0    9    3    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    54    101    0    9    3    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    55    104    0    9    3    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    56    112    0    9    3    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0
    57    110    0    9    3    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0

          ***** OUTPUT TARGET ARRAYS IN tran1  *****


( 1) SWORKO =    20   900     2     1   BOS                  1     2
     OPADRO    -108    -1
     SCONPO      27     1
     HFDOPO       0     0

( 2) SWORKO =     1    29    17     3   sky                  3    14
     OPADRO    -102  -107  -107  -103  -105  -108  -106  -101     3  -104  -112  -110
     SCONPO      28     2    29    30    31    32    33    34     3    35    36    37
     HFDOPO       0     2     0     0     0     0     0     0     0     0     0     0

( 3) SWORKO =     2   886     3     4   is                  15    25
     OPADRO    -113  -118  -111  -115  -120  -114  -116     4  -109  -117  -110
     SCONPO      38    39    40    41    42    43    44     4    45    46    47
```

```
      HFDOPO       0     0     0     0     0     0     0     0     0     0     0

 ( 4) SWORKO =     1   609    13     5   blue                26    36
      OPADRO    -102  -107  -103  -105  -108  -106  -101     5  -104  -112  -110
      SCONPO      48    49    50    51    52    53    54     5    55    56    57
      HFDOPO       0     0     0     0     0     0     0     0     0     0     0

 ( 5) SWORKO =    20    10     1     6   EOS                 37    37
      OPADRO       6
      SCONPO       6
      HFDOPO       0
*EOS*
*tran2   START*
    1  22 bos the sky is blue .

               ***** THE SWORK TABLE IN tran2  *****
  20  900   2  1          1   29  17  3          2  886   3  4          1  609  13  5         20   10   1  6
    BOS                    sky                    is                    blue                   EOS
CLSNFO ARRAYS - NUMBER OF CLAUSES IDENTIFIED   (INCLUDING MAIN CLAUSE) = 1
                NUMBER OF CLAUSES MOVED        (EXCLUDING MAIN CLAUSE) = 0
                NUMBER OF CLAUSES STILL TO BE MOVED                    =  0
             BEGIN  ENDING BEGIN  ENDING
       CLAUSE INPUT  INPUT  OUTPUT OUTPUT PARENT CLMRKR ANTCDN ANTCDN ANTCDN ANTCDN RELPRO
         ID   SWORK  SWORK  SWORK  SWORK  CLAUSE SCONS  SWORK  SCONPT OPIBEG OPIEND SCON
            1      1      5      1      0      0      0      0      0      0      0      0
       CLAUSE  PARENT
         ID    CELLS    ( TRAILING ZEROES ARE NOT PRINTED )

CURRENT CLAUSE ID =  1
CLSCON ARRAYS   (CLSID IS INITIALIZED TO 1. ENTRY NOT PRINTED IF CLSID=1 AND BOTH CMCHLD AND ACHILD = 0

***** A MATCH STARTING AT   1 LEVEL   3         ON ELEMENT  1jj           tran2
Tran rule #2226, ID: 2225
BOS .1S. V(95) = -A*0 / FORM FIELD IN C15 STS586 EGSP2
 3  (20 900 -1)  (51 -1 -1)  (2 -1 95)
   -20   0
   -63   1 376   1 -36  56   0 -41 100 999   10326 18499

 STR1CHG:     0        STR2CHG:     0        STR3CHG:     0

Main 30 table #1376
   -55  15 -83 351 999


 SW55 - LOADED CELL: 15 WITH VALUE:     3, VBRELP =  3

***** A MATCH STARTING AT   1 LEVEL   1         ON ELEMENT  1jj           tran2
```

```
Tran rule #2221, ID: 2220
BOS = -1 / SET C11=1 ST585 EGSP2
 1  (20 900 -1)
   -20   0 -55  11   1   0 -36  56   0 -41   1 999     0 3822

 SW55 - LOADED CELL: 11 WITH VALUE:   1

***** A MATCH STARTING AT   1 LEVEL   1         ON ELEMENT  1jj           tran2
Tran rule #1983, ID: 1982
PUNC = PUNC E2 GS1181
 1  (20 -1 -1)
   -63   0 447   1 999   32767 3504


Main 30 table #447
   -66  56 199 -81   2 942  60
   -56   3  56 199  31  22 777 777  28 791 -36 184 -81
   -57   1
   -66 299 234 -81   2 877  60
   -57   2  73   0  -1   0
   -57   3  72   0  91   0  81   0  85   0  83   0  88   0  89   0  96   0  84   0  86   0  90   0  87   0  92   0  82
0  97   0  98   0  93   0
   -57   4
   -66 699 599 -81  20 909 -81  20 719 -81  20 436  60
   -57   5 -54   1 -81  20   0
   -57   6 999


 SCON(  2,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  1   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  21
 SCON(  2,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57  3  JUMP -57  4
 SW57 - VTR BREAK POINT, K3:  30
 SW57 - VTR BREAK POINT, K3:  36
 SW57 - CONDITIONAL EXECUTION COMPLETED, BRANCH TO:   72
 SW57 - VTR BREAK POINT, K3:  72
 SCON( 20,-81) =    0
 SCON( 20,-81) =    0
 SCON( 20,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  5   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  87
 SW57 - VTR BREAK POINT, K3:  94
```

```
***** A MATCH STARTING AT   2 LEVEL   1         ON ELEMENT  3jj           tran2
Tran rule #1, ID: 0
**237 N = N / CK FOR CONNOM PREP-OBJ,GEN; IF SC40=15 CHAIN(S287);ST584 EGSP2
 1  (1 -1 -1)
   -42  10 237   1   1
   -66 199 299 -81  40  15  60
   -57   1 -20   0
   -57   2
   -63   0   2   1 999     0   0

 SCON( 40,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  17

Main 30 table #2
   -64   0 133   1 -81
   -66 940 126 -81   2 103 777 -81  59  94  60
   -66 128  56 -81  18  84  60
   -66 348  56 -81  46  13  60 -55   3 -81 351
   -56   1 799  56   3 175
   -56   1  56 599   3  65
   -66 238  56 -81  20  79  60
   -56   1  56 348   3  90
   -66 458 348 -81  36   0 777 -81  31   0  60
   -57   1 -38   1  -1   0 -13 -81
   -57   2  83   0 575   0 864   0 -38 -99  -1   0 -13 -81  89   0
   -57   3  83   0 -38 -99  -1   0 -13 -81  89   0
   -57   4  83   0 -25   5  -1   0 -13 -81  89   0
   -57   5
   -66 678 799 -81  20  79  60
   -57   6  83   0 575   0 864   0 -25   0  -1   0  89   0
   -57   7  83   0 -25   0  -1   0  89   0
   -57   8 999



Main 40 table #133
   -66 123  56 -81  28 800 777 -81  57  29 777 -81  13   1  60
   -66 299 399 -81  31 338  60
   -57   1 -16   3 -81   3 -81 -81
   -57   2 -54   1 -81   5   2
   -57   3
   -66 499 599 -81 235   0  60
   -57   4 -64   0  35   1 -81
   -57   5
   -66 699 799 -81 231   0 777 -81  46 140  60
```

```
  -57   6 -54   1 -81   3   9 -16 -81 -81 -81   0 -81
  -57   7 999


 SCON( 28,-81) =    0
 -66 056 CONDITION AT  12,                              CONTINUE TO THE RIGHT
 SCON( 31,-81) =  101
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  38
 SCON( 35,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  5   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  54
 SCON( 31,-81) =  101
 SCON( 46,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  7   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  80
 SCON(  2,-81) =   29
 -66 CONDITION FALSE, CONTINUE THIS VTR
 SCON( 18,-81) =    0
 -66 056 CONDITION AT  20,                              CONTINUE TO THE RIGHT
 SCON( 46,-81) =    0
 -66 056 CONDITION AT  27,                              CONTINUE TO THE RIGHT
 SW55 - LOADED CELL:  3 WITH VALUE:    17, VBRELP =  2
 CELL  3  =   17
 -56 056 CONDITION AT  39,                              CONTINUE TO THE RIGHT
 CELL  3  =   17
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  5   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 119
 SCON( 20,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  7   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 142
 SW57 - VTR BREAK POINT, K3: 152

***** A MATCH STARTING AT   3 LEVEL   3         ON ELEMENT  4jj           tran2
Tran rule #944, ID: 943
V(INTRANS) .5S. N(ADJ) = -A*0/ADJ=F90/ALTWC ST585 EGSP2
 3  (2 11 -1)  (55 -1 -1)  (1 -1 13)
   -20   0 -46 -83   0   0  90
   -63   0 948   1 -36  56   0 -41 100 999    14   1

 STR1CHG:    -1        STR2CHG:     0        STR3CHG:     0
```

```
Main 30 table #948
   -33 -83 999


***** A MATCH STARTING AT   3 LEVEL   2         ON ELEMENT  4jj           tran2
Tran rule #930, ID: 929
V(INTRANS) PREDADJ = V -1 ST1084 EGSP2
 2  (2 11 -1)  (1 -1 90)
   -63   0 504   1 -41   1 999   8261 20308


Main 30 table #504
   -55  11 -81  11  85   0
   -66 199 299 -81  10   0  60
   -57   1 111   0 115   0 120   0 114   0 116   0 109   0 117   0 110   0
   -57   2  -1   0
   -57   3 999


 SW55 - LOADED CELL: 11 WITH VALUE:    60, VBRELP =  3
 SCON( 10,-81) =    4
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  32
 SW57 - VTR BREAK POINT, K3:  36

***** A MATCH STARTING AT   4 LEVEL   1         ON ELEMENT  5jj           tran2
Tran rule #1, ID: 0
**237 N = N / CK FOR CONNOM PREP-OBJ,GEN; IF SC40=15 CHAIN(S287);ST584 EGSP2
 1  (1 -1 -1)
   -42  10 237   1   1
   -66 199 299 -81  40  15  60
   -57   1 -20   0
   -57   2
   -63   0   2   1 999     0   0


***** A MATCH STARTING AT   4 LEVEL   2         ON ELEMENT  5jj           tran2
Tran rule #1605, ID: 1604
01 ***237 N(PRED ADJ) = (ADJ) / E2 CMG1/92
 2  (10 237 1)  (1 -2 90)
  { 6081    8 8002    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0 }
   -63   4 233   1 999   -46


Main 30 table #4233
```

```
   -55   5 -81 352
   -56   1 123 299   5  13
   -57   1  83   0  -1   0  89   0
   -57   2  83   0 -38   0  -1   0  89   0
   -57   3 999


 SW55 - LOADED CELL:  5 WITH VALUE:    13, VBRELP =  4
 CELL  5  =   13
 -56 SWITCH TEST: CONDITION TRUE AT   9
 BRANCH TO -57  1   EXECUTE UNTIL -57  2  JUMP -57  3
 SW57 - VTR BREAK POINT, K3:  11
 SW57 - VTR BREAK POINT, K3:  19
 SW57 - CONDITIONAL EXECUTION COMPLETED, BRANCH TO:   29
 SW57 - VTR BREAK POINT, K3:  29

***** A MATCH STARTING AT   5 LEVEL   1         ON ELEMENT  6jj           tran2
Tran rule #2070, ID: 2069
EOS=EOS /BACKSTOP FOR REL SLOTS L143 KB1085 EGSP2
 1  (20 10 -1)
   -63   1 243   1 999   7031 8680


Main 30 table #1243
   -66 123 299 -81   2 877  60
   -57   1  85   0  83   0  88   0  89   0  96   0  84   0  86   0  90   0  87   0  92   0  82   0  97   0  98   0  93
0
   -57   2  95   0  91   0  94   0  81   0  85   0  83   0  88   0  89   0  96   0  84   0  86   0  90   0  87   0  92
0  82   0  97   0  98   0  93   0
   -57   3  -1   0 999


 SCON(  2,-81) =   10
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  38
 SW57 - VTR BREAK POINT, K3:  76


 -----------     tran2  PROCESSING COMPLETE      ----------
 THE SCON FOR tran2
              1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
       1     20    1    0    0    0    0    0    0    0    1    0    0    1    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0
              1    0    1    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
       2     14  101    1    1    0    3    1    0    0    2    0    1    1    0    0    0    0    0    0    0
```

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	94	0
	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
3	1	29	1	1	0	3	1	0	0	3	29	1	9	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	101	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	18	0	52	0
	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
4	2	886	1	0	0	3	0	0	0	4	60	2	11	0	0	0	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	13	0	1	0
	1	847	60	1	3	4	0	0	886	0	0	23	0	0	0	0	0	0	0	12
	0	0	0	0	0	1	60	0	3	0	0	0	0	0	0	0	0	0	0	0
5	1	609	9	3	0	3	1	2	0	5	40	1	6	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	16	0	83	0
	1	0	60	1	3	90	0	0	886	0	0	2	0	0	0	0	0	0	0	2
	0	0	0	0	0	7	21	0	1	0	0	0	0	0	0	0	0	0	0	0
6	20	10	0	0	0	0	0	0	0	6	0	0	10	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	60	1	3	90	0	0	886	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	108	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	102	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
29	107	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
30	103	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
31	105	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
32	108	0	1	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
      33    106    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0    0
      34    101    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0    0
      35    104    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0    0
      36    112    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0    0
      37    110    0    1    1    0    3    1    0    0    0    0    0    0    0    0    0    0    0    0    0
      38    113    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      39    118    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      40    111    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      41    115    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      42    120    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      43    114    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      44    116    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      45    109    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      46    117    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      47    110    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      48    102    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      49    107    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      50    103    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      51    105    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      52    108    0    9    3    0    3    0    2    0    0    0    0    0    0    0    0    0    0    0    0
      53    106    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      54    101    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      55    104    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      56    112    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      57    110    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0

              ***** THE SWORKO TABLE IN tran2  *****
  20  900   2  1          1   29  17  3          2  886   3  4          1  609  90  5          20   10   1  6
    BOS                    sky                    is                    blue                   EOS
CLSNFO ARRAYS - NUMBER OF CLAUSES IDENTIFIED   (INCLUDING MAIN CLAUSE) = 1
                NUMBER OF CLAUSES MOVED        (EXCLUDING MAIN CLAUSE) = 0
                NUMBER OF CLAUSES STILL TO BE MOVED                    =  0
             BEGIN  ENDING BEGIN  ENDING
       CLAUSE INPUT  INPUT  OUTPUT OUTPUT PARENT CLMRKR ANTCDN ANTCDN ANTCDN ANTCDN RELPRO
         ID   SWORK  SWORK  SWORK  SWORK  CLAUSE SCONS  SWORK  SCONPT OPIBEG OPIEND SCON
            1      1      5      1      5      0      0      0      0      0      0      0
       CLAUSE  PARENT
         ID    CELLS    ( TRAILING ZEROES ARE NOT PRINTED )

CURRENT CLAUSE ID =  1
CLSCON ARRAYS   (CLSID IS INITIALIZED TO 1. ENTRY NOT PRINTED IF CLSID=1 AND BOTH CMCHLD AND ACHILD = 0

             ***** OUTPUT TARGET ARRAYS IN tran2  *****


 ( 1) SWORKO =    20   900     2     1   BOS                 1     2
      OPADRO    -108    -1
```

```
      SCONPO      27     1
      HFDOPO       0     0

 ( 2) SWORKO =     1    29    17     3   sky                  3    14
      OPADRO    -102  -107  -107  -103  -105  -108  -106  -101     3  -104  -112  -110
      SCONPO      28     2    29    30    31    32    33    34     3    35    36    37
      HFDOPO       0     2     0     0     0     0     0     0     0     0     0     0

 ( 3) SWORKO =     2   886     3     4   is                  15    25
      OPADRO    -113  -118  -111  -115  -120  -114  -116     4  -109  -117  -110
      SCONPO      38    39    40    41    42    43    44     4    45    46    47
      HFDOPO       0     0     0     0     0     0     0     0     0     0     0

 ( 4) SWORKO =     1   609    90     5   blue                26    36
      OPADRO    -102  -107  -103  -105  -108  -106  -101     5  -104  -112  -110
      SCONPO      48    49    50    51    52    53    54     5    55    56    57
      HFDOPO       0     0     0     0     0     0     0     0     0     0     0

 ( 5) SWORKO =    20    10     1     6   EOS                 37    37
      OPADRO       6
      SCONPO       6
      HFDOPO       0
*EOS*
*tran3   START*
    1  22 bos the sky is blue .

               ***** THE SWORK TABLE IN tran3  *****
  20  900   2  1          1   29  17  3          2  886   3  4          1  609  90  5         20   10   1  6
    BOS                    sky                    is                    blue                   EOS
CLSNFO ARRAYS - NUMBER OF CLAUSES IDENTIFIED   (INCLUDING MAIN CLAUSE) = 1
                NUMBER OF CLAUSES MOVED        (EXCLUDING MAIN CLAUSE) = 0
                NUMBER OF CLAUSES STILL TO BE MOVED                    =  0
              BEGIN  ENDING BEGIN  ENDING
       CLAUSE INPUT  INPUT  OUTPUT OUTPUT PARENT CLMRKR ANTCDN ANTCDN ANTCDN ANTCDN RELPRO
         ID   SWORK  SWORK  SWORK  SWORK  CLAUSE SCONS  SWORK  SCONPT OPIBEG OPIEND SCON
           1      1      5      1      0      0      0      0      0      0      0      0
       CLAUSE  PARENT
         ID    CELLS    ( TRAILING ZEROES ARE NOT PRINTED )

CURRENT CLAUSE ID =  1
CLSCON ARRAYS   (CLSID IS INITIALIZED TO 1. ENTRY NOT PRINTED IF CLSID=1 AND BOTH CMCHLD AND ACHILD = 0

***** A MATCH STARTING AT   1 LEVEL   2    ON ELEMENT  1jj            tran3
Tran rule #1208, ID: 1207
BOS N = BOS -1 / N=WC07 E3 STS884 ESM1189
 2  (20 1 -1)  (1 -1 -1)
   -42  10 106   1   2
```

```
   -63   0 228   1 -24 -81 -20   0 -26 -82   1 -82 -82 -36  33 -82 -46 -82   7     0   0 -41   1 999   -31 888


Main 30 table #228
   -66 124  56 -81  10   1 777 -82  20 122 777 -82   2 175  60
   -66 235 499 -81   2 390 777 -82  20 390 777 -81  46  13  60
   -57   1 -54   1 -82  10   2
   -57   2 -38 -99  -1   0 -13 -81
   -57   3 -54   1 -81  46 140  -1 489 -16   2   0   0   0 -81 122   0 309   0 -13 -81
   -57   4  -1   0
   -57   5 -55   5 -81 350
   -56   1  56 799   5 900
   -66  56 799 -82   2 175 777 -82  13   0  60
   -66 699  56 -82  20  35 777 -82  19   0  60
   -66 699 799 -82  19 102 -82  19 104 -82  19  37  60
   -57   6 -54   1 -82  46 122 -54   1 -82  47 122
   -57   7 999


 SCON( 10,-81) =    1
 SCON( 20,-82) =    1
 -66 056 CONDITION AT  12,                              CONTINUE TO THE RIGHT
 SCON(  2,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  67
 SW57 - VTR BREAK POINT, K3:  71
 SW55 - LOADED CELL:  5 WITH VALUE:   900, VBRELP =  1
 CELL  5  =  900
 -56 056 CONDITION AT  81,                              CONTINUE TO THE RIGHT
 SCON(  2,-82) =   29
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  7   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 130
           tran3  OPADRO      0  -108    -1
          SWORKO   1      0  0  0  0 20900  2  1
          SWORKO   2     20900  2  1

***** A MATCH STARTING AT   2 LEVEL   1    ON ELEMENT  3jj            tran3
Tran rule #674, ID: 673
N (94) = -1/CK C30=1/DEL 136 IF POS
 1  (7 -1 94)
   -20   0
   -63   0 496   1 -36  56   0 -41   1 999   308 -18578


Main 30 table #496
```

```
   -66 199 299 -81  46 293 777 -81 220 140  60
   -57   1 -36 293 -81
   -57   2
   -66 399 499 -81   2 865  60
   -57   3 -54   1 -81   5   2
   -57   4 999


 SCON( 46,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  17
 SCON(  2,-81) =   29
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  33

***** A MATCH STARTING AT   2 LEVEL   3    ON ELEMENT  3jj            tran3
Tran rule #564, ID: 563
N(B4V) V(COP) N = -3/CK N BEFORE LOAD E3 CMG6/90
 3  (7 -1 -1)  (2 886 -1)  (1 -1 -1)
   -20   0
   -63   3 469   1 -36  56   0 -41   3 999   -81  17


Main 30 table #3469
   -66 123  56 -81   2 123 777 -81  12   9 777 -83 213   5  60
   -66  56 399 -81   2 303 777 -81 213   4 777 -81 213   7  60
   -66 299 399 -83   2 420 -83   2 433  60
   -57   1 -54   1 -81  46   9
   -57   2 -54   1 -81  46   3 -54   1 -81   4   3
   -57   3
   -66  56 499 -81  46 293  60 -54   1 -82  46 293
   -57   4 999


 SCON(  2,-81) =   29
 -66 056 CONDITION AT  12,                              CONTINUE TO THE RIGHT
 SCON(  2,-81) =   29
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  60
 SCON( 46,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  74
```

```
***** A MATCH STARTING AT   2 LEVEL   2    ON ELEMENT  3jj            tran3
Tran rule #554, ID: 553
N(B4V) V = N -1 / CK N BEFORE LOAD E3 CMG11/89 STS884 BMO890
 2  (7 -1 -1)  (-9 -1 -1)
   -63   1 929   1 -34   1 -81 -81 -81 -41   1 999     0   0


Main 30 table #1929
   -66 348  56 -81  20  35 777 -81   2 123 777 -81   1 -16  60
   -66 126  56 -81   2 144 777 -81  11  53 777 -81  31   0 777 -82   2 886  60
   -66 348  56 -81  46 293 777 -81  20 140  60
   -66 236  56 -81  33  46 777 -81  46  13  60
   -66 236  56 -81  46  13  60 -67  55   5  20   3 -55   3 -81  46
   -56   3 348  56   5 309 777 777   3 293
   -66 599  56 -81  46 293  60
   -66  56 346 -81   2 123 777 -81  11  21 777 -81 240  15  60
   -66 456 346 -81  31   0 777 -81 246   9 777 -81 209  23  60
   -57   1 -54   1 -81  46 101 -54   1 -81   8   1 297   0  -1   0 -13 -81
   -57   2 -38 -99  -1   0 -13 -81
   -57   3  -1   0
   -57   4 -54   1 -81  13   5 -36 488 -81  -1 488
   -57   5 -36 293 -81  -1   0
   -57   6
   -66  56 899 -81   2 303 777 -81 213   4 777 -81 213   7  60
   -56   6  56 899  16 392  16 103  16 440  16 122  16 866  17   1 -67  55   4  97   1 -67  55   5  98   1
   -56   3 799 899 404   0 777 777 405   0
   -57   7 -54   1   4   4 -81 -54   1 -81  98 303
   -57   8 999


 SCON( 20,-81) =    1
 -66 056 CONDITION AT  12,                          CONTINUE TO THE RIGHT
 SCON(  2,-81) =   29
 -66 056 CONDITION AT  31,                          CONTINUE TO THE RIGHT
 SCON( 46,-81) =    0
 -66 056 CONDITION AT  42,                          CONTINUE TO THE RIGHT
 SCON( 33,-81) =    1
 -66 056 CONDITION AT  53,                          CONTINUE TO THE RIGHT
 SCON( 46,-81) =    0
 -66 056 CONDITION AT  60,                          CONTINUE TO THE RIGHT
SW67 055:, SETTING CELL   5 EQUAL TO    0   FOR FUNCTION   3
 SW55 - LOADED CELL:  3 WITH VALUE:     0, VBRELP =  2
 CELL  5  =    0
 CELL  3  =    0
 -56 056 CONDITION AT  81,                          CONTINUE TO THE RIGHT
 SCON( 46,-81) =    0
 -66 056 CONDITION AT  86,                          CONTINUE TO THE RIGHT
```

```
 SCON(  2,-81) =   29
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57  4  JUMP -57  6
 SW57 - VTR BREAK POINT, K3: 146
 SW57 - VTR BREAK POINT, K3: 150
 SW57 - CONDITIONAL EXECUTION COMPLETED, BRANCH TO:  169
 SW57 - VTR BREAK POINT, K3: 169
 SCON(  2,-81) =   29
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  8   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 234
           tran3  OPADRO   -102  -107  -107  -103  -105  -108  -106  -101     3  -104  -112  -110
          SWORKO   2      7 29 17  3
          SWORKO   2      1 29 17  3

***** A MATCH STARTING AT   3 LEVEL   3    ON ELEMENT  4jj            tran3
Tran rule #13, ID: 12
*28 V .S. PUNC = -A*0(SICÜ) / SMTB STS586 EGSP3
 3  (2 -1 -1)  (52 -1 -1)  (20 -1 -1)
   -20   0
   -22   4 -81   1  -1   0 -99  91  -2   0  -3   0 -46 -81  17   0   0 -41 100 999   23696 2092

 STR1CHG:     0        STR2CHG:     0        STR3CHG:     0


      ***SEMWRK VALUES
          2  886   60    0          2  886    3    4          1   29   91    3          1  609   90    5          20
10    1    6

          SEMTAB MATCHING PARAMETERS HAVE BEEN LOADED AS FOLLOWS:
          LOGUSR =   1 USRUSR =   2EXTENDED SEARCH =   1 LUDIFF =   1
          EL1LVL =   1 CMPEL1 =   2 CMPELX =   2
          company codes [ 1] LOG
          SEMTAB: NO MATCH FOUND

  SEMWRKS =   1  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  1  3  2  4  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  3  5  3  1  0  0  0
          SWORKO   3      7 29 17  3

***** A MATCH STARTING AT   3 LEVEL   2    ON ELEMENT  4jj            tran3
Tran rule #1068, ID: 1067
**438 V(AFT SMTB) N = V -1/CK FOR PN E3 ST585 MMT987
 2  (17 -1 -1)  (1 -1 -1)
   -42  10 438   1   1  85   0  -1   0 -34   2 -81 -81 -81
   -63   2 118   3 -46 -82   5   0   0 -41     1 999   888 888
```

```
Main 30 table #2118
   -66 124  56 -81  19  93 -81  19  94  60
   -66 234  56 -81  19  35  60
   -66 399 499 -82  51  92 -82  51  93 -82  51  94  60
   -57   1 -54   1 -81  20  35 -67  54   1   2  19  35
   -57   2 -67  54   1   2  19  35
   -57   3 -54   1 -82  46  53
   -57   4 999


 SCON( 19,-81) =    1
 SCON( 19,-81) =    1
 -66 056 CONDITION AT   7,                              CONTINUE TO THE RIGHT
 SCON( 19,-81) =    1
 -66 056 CONDITION AT  14,                              CONTINUE TO THE RIGHT
 SCON( 51,-82) =    0
 SCON( 51,-82) =    0
 SCON( 51,-82) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  59
           tran3  OPADRO   -113  -118  -111  -115  -120  -114  -116     4  -109  -117  -110
          SWORKO   3      2886  3  4

***** A MATCH STARTING AT   4 LEVEL   1    ON ELEMENT  5jj            tran3
Tran rule #185, ID: 184
**101 N(AFTV) = N / CK FOR -ING ST1085 EGSP3 PS0787
 1  (5 -1 -1)
   -42  10 101   1   1
   -63   2 472   1 -34   1 -81 -81 -81 999   -66 126


Main 30 table #2472
   -66 299  56 -81   2 175 777 -81 213   4 777 -81 213   7  60
   -66 125  56 -81  46  13  60 -55   3 -81 351
   -56   1 125 299 503  85
   -57   1  73   0 -38 -99  -1   0 -13 -81
   -57   2
   -66 345  56 -81   1  19 777 -81   2 893 777 -81  62 850  60
   -66 345 499 -81   1  19 777 -81   2 895 777 -81  62 850  60
   -57   3 297   0  -1   0
   -57   4  73   0  -1   0
   -57   5 999


 SCON(  2,-81) =  609
```

```
 -66 056 CONDITION AT  12,                              CONTINUE TO THE RIGHT
 SCON( 46,-81) =    0
 -66 056 CONDITION AT  19,                              CONTINUE TO THE RIGHT
 SW55 - LOADED CELL:  3 WITH VALUE:    90, VBRELP =  4
 CELL  3  =   90
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  43
 SCON(  1,-81) =    1
 -66 056 CONDITION AT  56,                              CONTINUE TO THE RIGHT
 SCON(  1,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  81
 SW57 - VTR BREAK POINT, K3:  87
          tran3  OPADRO   -102  -107  -103  -105  -108  -106  -101     5  -104  -112  -110
         SWORKO   4      5609 90  5
         SWORKO   4      1609 90  5

***** A MATCH STARTING AT   5 LEVEL   1    ON ELEMENT  6jj            tran3
Tran rule #1289, ID: 1288
EOS = EOS S1079 BMO1090
 1  (20 10 -1)
   -63   0 294   1 -55  30   0   0 999   1018   0


Main 30 table #294
   -55  16   0   0 -55  17   0   0  -1   0 999


 SW55 - LOADED CELL: 16 WITH VALUE:   0
 SW55 - LOADED CELL: 17 WITH VALUE:   0
          tran3  OPADRO      6
         SWORKO   5     20 10  1  6
 SW55 - LOADED CELL: 30 WITH VALUE:   0
 THE SCON FOR tran3
             1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
      1     20    1    0    0    0    0    0    0    0    1    0    0    1    0    0    0    0    0    0    0
             0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
             0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0
             1    0    1    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
      2     14  101    1    1    0    3    1    0    0    2    0    1    1    0    0    0    0    0    0    0
             0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
             0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0   94    0
             1    0    1    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
      3      1   29    1    1    0    3    1    0    0    3   29    1    9    0    0    0    0    0    0    1
             0    0    0    0    0    0    0    0    0    0  101    0    1    0    0    0    0    0    0    0
```

	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	18	0	52	0
	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
4	2	886	1	0	0	3	0	0	0	4	60	2	11	0	0	0	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	13	0	1	0
	1	847	60	1	3	4	0	0	886	0	0	23	0	0	0	0	0	0	0	12
	0	0	0	0	0	1	60	0	3	0	0	0	0	0	0	0	0	0	0	0
5	1	609	9	3	0	3	1	2	0	5	40	1	6	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	16	0	83	0
	1	0	60	1	3	90	0	0	886	0	0	2	0	0	0	0	0	0	0	2
	0	0	0	0	0	7	21	0	1	0	0	0	0	0	0	0	0	0	0	0
6	20	10	0	0	0	0	0	0	0	6	0	0	10	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	60	1	3	90	0	0	886	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	108	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	102	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
29	107	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
30	103	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
31	105	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
32	108	0	1	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	106	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
34	101	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
35	104	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
36	112	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
37	110	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0

```
      38    113    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      39    118    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      40    111    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      41    115    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      42    120    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      43    114    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      44    116    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      45    109    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      46    117    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      47    110    0    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0    0
      48    102    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      49    107    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      50    103    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      51    105    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      52    108    0    9    3    0    3    0    2    0    0    0    0    0    0    0    0    0    0    0    0
      53    106    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      54    101    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      55    104    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      56    112    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
      57    110    0    9    3    0    3    1    2    0    0    0    0    0    0    0    0    0    0    0    0
CLSNFO ARRAYS - NUMBER OF CLAUSES IDENTIFIED   (INCLUDING MAIN CLAUSE) = 1
                NUMBER OF CLAUSES MOVED        (EXCLUDING MAIN CLAUSE) = 0
                NUMBER OF CLAUSES STILL TO BE MOVED                    =  0
              BEGIN  ENDING BEGIN  ENDING
       CLAUSE INPUT  INPUT  OUTPUT OUTPUT PARENT CLMRKR ANTCDN ANTCDN ANTCDN ANTCDN RELPRO
         ID   SWORK  SWORK  SWORK  SWORK  CLAUSE SCONS  SWORK  SCONPT OPIBEG OPIEND SCON
           1      1      5      1      5      0      0      0      0      0      0      0
       CLAUSE  PARENT
         ID    CELLS    ( TRAILING ZEROES ARE NOT PRINTED )

CURRENT CLAUSE ID =  1
CLSCON ARRAYS   (CLSID IS INITIALIZED TO 1. ENTRY NOT PRINTED IF CLSID=1 AND BOTH CMCHLD AND ACHILD = 0

               ***** THE SWORKO TABLE IN tran3  *****
  20  900   2  1          1   29  17  3          2  886   3  4          1  609  90  5         20   10   1  6
    BOS                    sky                    is                    blue                   EOS

             ***** OUTPUT TARGET ARRAYS IN tran3  *****


 ( 1) SWORKO =    20   900     2     1   BOS                 1     2
      OPADRO    -108    -1
      SCONPO      27     1
      HFDOPO       0     0

 ( 2) SWORKO =     1    29    17     3   sky                 3    14
      OPADRO    -102  -107  -107  -103  -105  -108  -106  -101     3  -104  -112  -110
```

```
      SCONPO      28     2    29    30    31    32    33    34     3    35    36    37
      HFDOPO       0     2     0     0     0     0     0     0     0     0     0     0

 ( 3) SWORKO =     2   886     3     4   is                  15    25
      OPADRO    -113  -118  -111  -115  -120  -114  -116     4  -109  -117  -110
      SCONPO      38    39    40    41    42    43    44     4    45    46    47
      HFDOPO       0     0     0     0     0     0     0     0     0     0     0

 ( 4) SWORKO =     1   609    90     5   blue                26    36
      OPADRO    -102  -107  -103  -105  -108  -106  -101     5  -104  -112  -110
      SCONPO      48    49    50    51    52    53    54     5    55    56    57
      HFDOPO       0     0     0     0     0     0     0     0     0     0     0

 ( 5) SWORKO =    20    10     1     6   EOS                 37    37
      OPADRO       6
      SCONPO       6
      HFDOPO       0
*EOS*
*tran4   START*
    1  22 bos the sky is blue .

               ***** THE SWORK TABLE IN tran4  *****
  20  900   2  1          1   29  17  3          2  886   3  4          1  609  90  5         20   10   1  6
    BOS                   sky                    is                    blue                   EOS
CLSNFO ARRAYS - NUMBER OF CLAUSES IDENTIFIED   (INCLUDING MAIN CLAUSE) = 1
                NUMBER OF CLAUSES MOVED        (EXCLUDING MAIN CLAUSE) = 0
                NUMBER OF CLAUSES STILL TO BE MOVED                    =  0
              BEGIN  ENDING BEGIN  ENDING
       CLAUSE INPUT  INPUT  OUTPUT OUTPUT PARENT CLMRKR ANTCDN ANTCDN ANTCDN ANTCDN RELPRO
         ID   SWORK  SWORK  SWORK  SWORK  CLAUSE SCONS  SWORK  SCONPT OPIBEG OPIEND SCON
           1      1      5      1      0      0      0      0      0      0      0      0
       CLAUSE  PARENT
         ID    CELLS    ( TRAILING ZEROES ARE NOT PRINTED )

CURRENT CLAUSE ID =  1
CLSCON ARRAYS   (CLSID IS INITIALIZED TO 1. ENTRY NOT PRINTED IF CLSID=1 AND BOTH CMCHLD AND ACHILD = 0

***** A MATCH STARTING AT   1 LEVEL   1       ON ELEMENT  1jj       tran4
Tran rule #2313, ID: 2312
PURE BOS = -1 / CELL 20 = 1 ST286 EGSP4
 1  (20 900 2)
   -63   1 346   1 -36  56   0 -41   1 999     0   0


Main 30 table #1346
   -55  20   1   0 999
```

```
 SW55 - LOADED CELL: 20 WITH VALUE:   1

***** A MATCH STARTING AT   1 LEVEL   3       ON ELEMENT  1jj       tran4
Tran rule #2255, ID: 2254
BOS .STR. V = -A*0 / SCON 21 OF VERB IN CELL 21 E4 STS1087
 3  (20 900 -1)  (52 -1 -1)  (2 -1 95)
   -63   1 867   1 -36  56   0 -41 100 999     0   0

 STR1CHG:     0        STR2CHG:     0        STR3CHG:     0

Main 30 table #1867
   -55  11 -83  11 -55  28 -83 350 -55  18 -83 351 -55  19 -83  19 -55  21 -83  31 -55  43 -83   3 999


 SW55 - LOADED CELL: 11 WITH VALUE:    60, VBRELP =  3
 SW55 - LOADED CELL: 28 WITH VALUE:   886, VBRELP =  3
 SW55 - LOADED CELL: 18 WITH VALUE:     3, VBRELP =  3
 SW55 - LOADED CELL: 19 WITH VALUE:     1, VBRELP =  3
 SW55 - LOADED CELL: 21 WITH VALUE:     0, VBRELP =  3
 SW55 - LOADED CELL: 43 WITH VALUE:     1, VBRELP =  3

***** A MATCH STARTING AT   1 LEVEL   2       ON ELEMENT  1jj       tran4
Tran rule #2243, ID: 2242
**272 BOS N = BOS -1 / N=WC08 / CK FOR GENUINE BOS ST585 EGSP4
 2  (20 900 -1)  (1 -1 -1)
   -42  10 272   1   1 -46 -82   8   0   0
   -63   0 707   1 -41   1 999     0   0


***** A MATCH STARTING AT   1 LEVEL   2       ON ELEMENT  1jj       tran4
Tran rule #1360, ID: 1359
02 ***272 BOS N = BOS -1 / GENUINE BOS; N=WC18 ST585 EGSP4
 2  (10 272 1)  (-1 -1 2)
   -46 -82  18   0   0
   -63   0 696   1 -41   1 999     0   0


Main 30 table #696
   -55  20   1   0  -1   0
   -66  56 399 -82   2 175 777 -82  13   0  60
   -56   1 235 125  15   0
   -57   1 -55  29  81   0 -54   1 -82  46 122
   -57   2 -55  29  91   0 -54   1 -82  46 122
   -57   3
   -66 499 599 -82  31 115 777 -82   5   2  60
   -57   4 -44 -82 107 456   0
```

```
   -57   5
   -66 699 799 -81   2 888  60
   -57   6 -46 -82   8   0   0 -55  29   0   0
   -57   7 999


 SW55 - LOADED CELL: 20 WITH VALUE:   1
 SCON(  2,-82) =   29
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  46
 SCON( 31,-82) =  101
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  5   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  66
 SCON(  2,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  7   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  86

***** A MATCH STARTING AT   2 LEVEL   2       ON ELEMENT  3jj       tran4
Tran rule #1714, ID: 1713
**53 (MC) N(B4V) V = -2 / CK F91,F66 E4 ST1284 MMT487
 2  (18 -1 -1)  (2 -1 -1)
   -42  10  53   1   1 -46 -81   0 851   0 -46 -82   3   0   0
   -63   2 132   1 -41   2 999     0   0


***** A MATCH STARTING AT   2 LEVEL   2       ON ELEMENT  3jj       tran4
Tran rule #1227, ID: 1226
02 ***53 (MC) N(NOM) V = -2/AGREEMENT/CK PASSACT/CK AGENT ST785 EGSP4
 2  (10 53 1)  (18 -1 91)
   -46 -81   0 851   0 -46 -82   3   0   0
   -63   0 187   1 -41   2 999     0   0


Main 30 table #187
   -66 899  56 -81   2 144 777 -81  11  94 777 -82   2 886 777 -81  60 291  60
   -66 348  56 -81  20 140  60
   -66 568  56 -82   2 571 777 -81  19  93  60
   -66 568  56 -82   2 930 777 -81  19  93  60 -55   5 -82 351 -55   9 -81  19
   -56   3  56 199 605  65 777 777   9  93
   -66 678 199 -81  19  93 777 -82 202 571 777 -82 202 930  60
   -57   1 -55   2 -82 351 -55   3 -81   2 -55   4 -81  11
   -56   5 799  56 602  85 777 777   3 140 777 777   4  30
   -56   3 238  56  19   4 777 777  17   1
   -56   3 238  56  19   3 777 777  17   1
```

```
  -56   3 348  56  19   4 777 777 618  65
  -56   3 348 458  19   3 777 777 618  65
  -57   2 -54   1 -82 456  24
  -57   3 -16   3   1   3   0 -82
  -57   4 -16 -81 -81 -81   0 -82
  -57   5 -54   1 -81  20  93
  -57   6 -54   1 -81  19   1
  -57   7 -54   1 -81   5   1 -54   1 -81   4   3 -16   3   1   3   0 -82
  -57   8 999


SCON(  2,-81) =   29
-66 056 CONDITION AT  16,                              CONTINUE TO THE RIGHT
SCON( 20,-81) =    1
-66 056 CONDITION AT  23,                              CONTINUE TO THE RIGHT
SCON(  2,-82) =  886
-66 056 CONDITION AT  34,                              CONTINUE TO THE RIGHT
SCON(  2,-82) =  886
-66 056 CONDITION AT  45,                              CONTINUE TO THE RIGHT
SW55 - LOADED CELL:  5 WITH VALUE:     3, VBRELP =  3
SW55 - LOADED CELL:  9 WITH VALUE:     0, VBRELP =  2
CELL  5  =    3
CELL  9  =    0
-56 SWITCH TEST: CONDITION FALSE
BRANCH TO -57  1   EXECUTE UNTIL -57 99  JUMP -57 99
SW57 - VTR BREAK POINT, K3:  82
SW55 - LOADED CELL:  2 WITH VALUE:     3, VBRELP =  3
SW55 - LOADED CELL:  3 WITH VALUE:    29, VBRELP =  2
SW55 - LOADED CELL:  4 WITH VALUE:    29, VBRELP =  2
CELL  2  =    3
CELL  3  =   29
CELL  4  =   29
-56 056 CONDITION AT 108,                              CONTINUE TO THE RIGHT
CELL 19  =    1
CELL 17  =    0
-56 056 CONDITION AT 118,                              CONTINUE TO THE RIGHT
CELL 19  =    1
CELL 17  =    0
-56 056 CONDITION AT 128,                              CONTINUE TO THE RIGHT
CELL 19  =    1
CELL 18  =    3
-56 056 CONDITION AT 138,                              CONTINUE TO THE RIGHT
CELL 19  =    1
CELL 18  =    3
-56 SWITCH TEST: CONDITION FALSE
BRANCH TO -57  4   EXECUTE UNTIL -57  5  JUMP -57  8
SW57 - VTR BREAK POINT, K3: 165
```

```
 SW57 - VTR BREAK POINT, K3: 173
 SW57 - CONDITIONAL EXECUTION COMPLETED, BRANCH TO:  205
 SW57 - VTR BREAK POINT, K3: 205

***** A MATCH STARTING AT   2 LEVEL   1       ON ELEMENT  3jj       tran4
Tran rule #1809, ID: 1808
(MC) N(B4V-NOM) = N)/S91/SC13 IN C27(SUPERSET)CMG7/87/ST585 EGSP4
 1  (18 851 91)
   -63   0 471   1 999     0   0


Main 30 table #471
   -66 950 902 -81   2 175 777 -81  13   0  60
   -56   1 199  56  17   1 -65   0 953   1 -81 -64   0 252   1 -81 -55  30  18   0
   -57   1
   -66 237  56 -81  20 122 777 -81  10   2  60
   -66 567  56 -81  19 291 777 -81 220 291  60
   -66 699  56 -81  20 140  60
   -66 347 457 -81  19 302 -81  20 291  60
   -57   2  -1   0
   -57   3 -31 -81 -11  91 291   0
   -57   4 -64   0 176   1 -81 -55  29  91   0 -65   0 297   0
   -57   5 -11  91 291   0 -11  93 122   0 149   0  -1   0 -16 -81 -81 -81   1 -81 -31  11
   -57   6 -11  99  -1   0 -31  11
   -57   7 999


 SCON(  2,-81) =   29
 -66 CONDITION FALSE, CONTINUE THIS VTR
 CELL 17  =    0
 -56 056 CONDITION AT  16,                              CONTINUE TO THE RIGHT

Main 50 table #953
   -66 299  56 -81   2 175 777 -81  13   0  60 -55   5 -81 352
   -56   1  56 299 505  71
   -66 299  56 -81   4   0  60
   -66 299  56 -81   2 350 777 -81  11   0  60
   -56   1 299  56  37 117
   -66 299  56 -81   5   2 -81   2 865  60 -55   5 -81   2 -55   6 -81   1 -55   7 -81  13
   -56   5 299  56   5 303 777 777   6   5 777 777  13   0
   -56   3 299  56   7   5 777 777   6   5
   -56   3 299  56   5 102 777 777   6   5
   -66 299 199 -81  13   5 777 -81 211  94  60
   -57   1 -55  13 -81   4
   -57   2 999
```

```
 SCON(  2,-81) =   29
 -66 056 CONDITION AT   8,                              CONTINUE TO THE RIGHT
 SW55 - LOADED CELL:  5 WITH VALUE:     1, VBRELP =  2
 CELL  5  =    1
 -56 056 CONDITION AT  20,                              CONTINUE TO THE RIGHT
 SCON(  4,-81) =    1
 -66 056 CONDITION AT  25,                              CONTINUE TO THE RIGHT
 SCON(  2,-81) =   29
 -66 056 CONDITION AT  36,                              CONTINUE TO THE RIGHT
 CELL 37  =    0
 -56 056 CONDITION AT  44,                              CONTINUE TO THE RIGHT
 SCON(  5,-81) =    0
 SCON(  2,-81) =   29
 -66 056 CONDITION AT  52,                              CONTINUE TO THE RIGHT
 SW55 - LOADED CELL:  5 WITH VALUE:    29, VBRELP =  2
 SW55 - LOADED CELL:  6 WITH VALUE:     1, VBRELP =  2
 SW55 - LOADED CELL:  7 WITH VALUE:     9, VBRELP =  2
 CELL  5  =   29
 CELL  6  =    1
 CELL 13  =    0
 -56 056 CONDITION AT  80,                              CONTINUE TO THE RIGHT
 CELL  7  =    9
 CELL  6  =    1
 -56 056 CONDITION AT  90,                              CONTINUE TO THE RIGHT
 CELL  5  =   29
 CELL  6  =    1
 -56 056 CONDITION AT 100,                              CONTINUE TO THE RIGHT
 SCON( 13,-81) =    9
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  1   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 113
 SW55 - LOADED CELL: 13 WITH VALUE:     1, VBRELP =  2
 SW57 - VTR BREAK POINT, K3: 119

Main 40 table #252
   -55   5 -81 351
   -56   1 399  56   5  96
   -66 399  56 -81   1   3 -81   1   6  60
   -66 399  56 -81   2 175 777 -81  13   0  60 -55   5 -81  13 -55   4 -81  62
   -56   3 399  56 424   0 777 777   5   6
   -56   3 399  56 424   0 777 777   5   4
   -56   3 399  56 424   0 777 777   5   7
   -56   3 399  56 424   0 777 777   4 864
   -56   3 399  56 444   0 777 777   5   6
   -56   3 399  56 444   0 777 777   5   4
   -56   3 399  56 444   0 777 777   5   7
   -56   3 399  56 444   0 777 777   4 864
```

```
  -66 399  56 -81   1   5 777 -81 202 303  60
  -66 399  56 -81   1  11 -81   1  13  60 -55   1 -81 351
  -56   1 399  56   1  66 -55   5 -81 352
  -56   1  56 399 505  71
  -66 399  56 -81   4   0  60
  -66 399  56 -81   2 350 777 -81  11   0  60
  -56   1 399  56  37 117
  -66 399  56 -81   5   2 -81   2 865  60
  -66 399  56 -81  46 293 -81  60 293  60
  -66 399 199 -81  13   5 777 -81 211  94  60
  -57   1 -55  24 -81 456
  -56   3 299 399  97   0 777 777  98   0
  -57   2 -55  97 -81   4 -55  98 -81   5 -67   6  97  98 -96 -81
  -57   3 999


SW55 - LOADED CELL:  5 WITH VALUE:    17, VBRELP =  2
CELL  5  =   17
-56 056 CONDITION AT   9,                              CONTINUE TO THE RIGHT
SCON(  1,-81) =    1
SCON(  1,-81) =    1
-66 056 CONDITION AT  17,                              CONTINUE TO THE RIGHT
SCON(  2,-81) =   29
-66 056 CONDITION AT  28,                              CONTINUE TO THE RIGHT
SW55 - LOADED CELL:  5 WITH VALUE:     9, VBRELP =  2
SW55 - LOADED CELL:  4 WITH VALUE:     0, VBRELP =  2
CELL 24  =    0
CELL  5  =    9
-56 056 CONDITION AT  48,                              CONTINUE TO THE RIGHT
CELL 24  =    0
CELL  5  =    9
-56 056 CONDITION AT  58,                              CONTINUE TO THE RIGHT
CELL 24  =    0
CELL  5  =    9
-56 056 CONDITION AT  68,                              CONTINUE TO THE RIGHT
CELL 24  =    0
CELL  4  =    0
-56 056 CONDITION AT  78,                              CONTINUE TO THE RIGHT
CELL 44  =    0
CELL  5  =    9
-56 056 CONDITION AT  88,                              CONTINUE TO THE RIGHT
CELL 44  =    0
CELL  5  =    9
-56 056 CONDITION AT  98,                              CONTINUE TO THE RIGHT
CELL 44  =    0
CELL  5  =    9
-56 056 CONDITION AT 108,                              CONTINUE TO THE RIGHT
```

```
CELL 44  =    0
CELL  4  =    0
-56 056 CONDITION AT 118,                              CONTINUE TO THE RIGHT
SCON(  1,-81) =    1
-66 056 CONDITION AT 127,                              CONTINUE TO THE RIGHT
SCON(  1,-81) =    1
SCON(  1,-81) =    1
-66 056 CONDITION AT 137,                              CONTINUE TO THE RIGHT
SW55 - LOADED CELL:  1 WITH VALUE:    17, VBRELP =  2
CELL  1  =   17
-56 056 CONDITION AT 149,                              CONTINUE TO THE RIGHT
SW55 - LOADED CELL:  5 WITH VALUE:     1, VBRELP =  2
CELL  5  =    1
-56 056 CONDITION AT 159,                              CONTINUE TO THE RIGHT
SCON(  4,-81) =    1
-66 056 CONDITION AT 164,                              CONTINUE TO THE RIGHT
SCON(  2,-81) =   29
-66 056 CONDITION AT 175,                              CONTINUE TO THE RIGHT
CELL 37  =    0
-56 056 CONDITION AT 183,                              CONTINUE TO THE RIGHT
SCON(  5,-81) =    0
SCON(  2,-81) =   29
-66 056 CONDITION AT 191,                              CONTINUE TO THE RIGHT
SCON( 46,-81) =    0
SCON( 60,-81) =    0
-66 056 CONDITION AT 201,                              CONTINUE TO THE RIGHT
SCON( 13,-81) =    9
-66 SWITCH TEST: CONDITION FALSE
BRANCH TO -57  1   EXECUTE UNTIL -57 99  JUMP -57 99
SW57 - VTR BREAK POINT, K3: 216
SW55 - LOADED CELL: 24 WITH VALUE:   113, VBRELP =  2
CELL 97  =    0
CELL 98  =    0
-56 SWITCH TEST: CONDITION TRUE AT 230
BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
SW57 - VTR BREAK POINT, K3: 232
SW55 - LOADED CELL: 97 WITH VALUE:     1, VBRELP =  2
SW55 - LOADED CELL: 98 WITH VALUE:     0, VBRELP =  2
SW57 - VTR BREAK POINT, K3: 248
SW55 - LOADED CELL: 30 WITH VALUE:  18
SW57 - VTR BREAK POINT, K3:  32
SCON( 20,-81) =    1
-66 056 CONDITION AT  41,                              CONTINUE TO THE RIGHT
SCON( 19,-81) =    0
-66 056 CONDITION AT  52,                              CONTINUE TO THE RIGHT
SCON( 20,-81) =    1
-66 056 CONDITION AT  59,                              CONTINUE TO THE RIGHT
```

```
 SCON( 19,-81) =    0
 SCON( 20,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57  5  JUMP -57  7
 SW57 - VTR BREAK POINT, K3:  85

Main 40 table #176
   -56   1  93  10  99  99  56  50 122 -55   5 -81 351
   -56   1  93  10  99  99  56   5  96
   -66  93  10  99  99  56 -81   1   3 -81   1   6  60
   -66 399  56 -81  13   5 -81  11  35 -81  11  94  60
   -56   1  56 699  50   0
   -56   1 126 299   5  66
   -57   1 -55  50 123   0 -67   6  50  50 -96 -81
   -57   2
   -57   3 -55   6 -81 352
   -56   2  93  10  99  99  56   6  50   6   5
   -56   1 456 599   5  66
   -57   4 -55  50 123   0 -67   6  50  50 -96 -81
   -57   5 -55  50 -81 456 -67   6  50  50 -96 -81
   -57   6
   -56   1  56  93  10  99  99  27   0
   -66  93   7   8  99 899 -81  13   3 777 -81  11  35  60
   -57   7 -55  27  35   0
   -57   8 -55  27 -81  13
   -57  10 999


 CELL 50  =    0
 -56 056 CONDITION AT   8,                              CONTINUE TO THE RIGHT
 SW55 - LOADED CELL:  5 WITH VALUE:    17, VBRELP =  2
 CELL  5  =   17
 -56 056 CONDITION AT  21,                              CONTINUE TO THE RIGHT
 SCON(  1,-81) =    1
 SCON(  1,-81) =    1
 -66 056 CONDITION AT  32,                              CONTINUE TO THE RIGHT
 SCON( 13,-81) =    9
 SCON( 11,-81) =   29
 SCON( 11,-81) =   29
 -66 056 CONDITION AT  45,                              CONTINUE TO THE RIGHT
 CELL 50  =    0
 -56 056 CONDITION AT  53,                              CONTINUE TO THE RIGHT
 CELL  5  =   17
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  73
 SW57 - VTR BREAK POINT, K3:  75
```

```
 SW55 - LOADED CELL:  6 WITH VALUE:     1, VBRELP =  2
 CELL  6  =    1
 CELL  6  =    1
 -56 056 CONDITION AT  90,                              CONTINUE TO THE RIGHT
 CELL  5  =   17
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  5   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 110
 SW55 - LOADED CELL: 50 WITH VALUE:   113, VBRELP =  2
 SW57 - VTR BREAK POINT, K3: 122
 CELL 27  =    0
 -56 056 CONDITION AT 131,                              CONTINUE TO THE RIGHT
 SCON( 13,-81) =    9
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  8   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 153
 SW55 - LOADED CELL: 27 WITH VALUE:     9, VBRELP =  2
 SW57 - VTR BREAK POINT, K3: 159
 SW55 - LOADED CELL: 29 WITH VALUE:  91

Main 50 table #297
   -66 950 902 -81   2 175 777 -81 213   4 777 -81 213   7  60
   -66 950 902 -81   2 392 777 -81  13   1  60
   -66 950 902 -81   2 401 777 -81  13   1  60
   -66 950 902 -81   2 440 777 -81  13   2  60
   -66 950 711 -81  20  92 -81  20  93 -81  20  94  60
   -66 199  56 -81  62 859 -81  62 858 -81  62 865 -81  62 989  60 -65   0 627   1 -81 -64   0 179   1 -81
   -57   1
   -66 239  56 -81  20  54  60
   -56   3 239  56  19   4 777 777  17   1
   -56   3 349  56  19   3 777 777  17   1
   -56   3 459  56 618  65 777 777  19   4
   -56   3 569  56 618  65 777 777  19   3
   -56   3  93  10  11  99  56 618  65 777 777  19   2
   -66 789  56 -81  19  93 -81  20  93  60
   -56   5 789  56  43   3 777 777 618  65 777 777  17   0
   -56   5 789  56  43   7 777 777 618  65 777 777  17   0
   -56   5 789  56  43   2 777 777 618  65 777 777  17   0
   -56   5 789  56  43   6 777 777 618  65 777 777  17   0
   -66 349  56 -81  20  53  60
   -66  93   8  10  99  56 -81  20 488  60
   -66  93   8  10  99  56 -81  19  94  60
   -66 789  56 -81  19  93  60
   -66 459  56 -81  20  84  60
   -66 569  56 -81  20  83  60
   -56   1  93   8  10  99 679  17   1
   -57   2 -55  29  83   0 -11  83 117   0 -38   4  -1   0
```

```
   -57   3 -55  29  83   0 -11  83 117   0 -38   3  -1   0
   -57   4 -55  29  91   0 -11  91 117   0 -38   4  -1   0
   -57   5 -55  29  91   0 -11  91 117   0 -38   3  -1   0
   -57   6 -55  29  91   0 -11  91 -38   1  -1   0
   -57   7 -55  29  91   0 -11  91 -38   3  -1   0
   -57   8 -55  29  88   0 -11  88 -38   4  -1   0 -31  11
   -57  10 -55  29  88   0 -11  91 117   0 -38   2  -1   0 -31  11
   -57  11 999


 SCON(  2,-81) =   29
 -66 CONDITION FALSE, CONTINUE THIS VTR
 SCON(  2,-81) =   29
 -66 CONDITION FALSE, CONTINUE THIS VTR
 SCON(  2,-81) =   29
 -66 CONDITION FALSE, CONTINUE THIS VTR
 SCON(  2,-81) =   29
 -66 CONDITION FALSE, CONTINUE THIS VTR
 SCON( 20,-81) =    1
 SCON( 20,-81) =    1
 SCON( 20,-81) =    1
 -66 CONDITION FALSE, CONTINUE THIS VTR
 SCON( 62,-81) =    0
 SCON( 62,-81) =    0
 SCON( 62,-81) =    0
 SCON( 62,-81) =    0
 -66 056 CONDITION AT  74,                            CONTINUE TO THE RIGHT

Main 50 table #627
   -66 699  56 -81  31 101 -81  46 101 -81  62 849 -81  62 864 -81  62 848 -81  62 851 -81  46 621 -81  46 341 -81   9
11 -81   9  21 -81   9  31 -81   9  19 -81   9  29 -81   9  39  60
   -66 699  56 -81   2 303 777 -81 213   4 777 -81 213   7  60 -55   3 -81 350
   -56   1 699  56   3 175
   -66 699  56 -81  46 319 -81  46 101 -81  19 140  60
   -66 699  56 -81  43 140  60 -55   6 -81  31 -55   5 -81 351
   -56   3 699  56   6   0 777 777   5  90 -55   1 -81   2 -55   2 -81  11 -55   8 -81  13 -55   9 -81  46
   -56   7 456  56   6   0 777 777   9   0 777 777   2  21 777 777  43   2
   -56   9 456  56   5  91 777 777   1 855 777 777   2  94 777 777   8   5 777 777   6   0
   -66 699  56 -81  60 140 -81  20 140  60
   -56   2 699  56   5  43   5  55
   -56   1  56 399  15 942
   -66 126  56 -81  31 115 777 -81   5   2  60
   -66 236 399 -81  31 115 777 -81   5   1  60
   -57   1 -44 -81 107 140   0 -54   1 -81  46 140 -48  43 -81   9
   -57   2 -44 -81 107 532   0 -54   1 -81  46 315 -48  43 -81   6
   -57   3 -55   3 -81 352
   -56   1 699  56   3  23
```

```
   -66 699  56 -81  31 115 777 -81 246   0  60
   -66 699  56 -81 239   0 -81  42 575  60
   -66 699  56 -81  62 850 -81  62 864  60
   -66 699  56 -81   1   5  60
   -66  56 699 -81  31   0  60
   -66  56 699 -81   5   1 -81   5   0  60
   -66  56 699 -81  32   0 777 -81  33   0 777 -81  35   0 777 -81  36   0 777 -81  41   0 777 -81  46   0  60
   -66 699  56 -81  11  21 -81  11  52  60
   -66 456  56 -81  13   4 -81  13   7  60
   -66 456  56 -81   2  23 -81   2 582 -81   2 327 -81   2 655 -81   2 602 -81   2  78 -81   2 702 -81   2  50 -81   2
173 -81   2 708 -81   2 749 -81   2 574 -81   2 450 -81   2 716 -81   2 297  60
   -66  56 699 -81   2  46 -81   2  49 -81   2 609  60
   -66 456 699 -81  17  58 -81  42   0  60
   -57   4 -48  43 -81   3 -44 -81 107 131   0 -54   1 -81  46 101
   -57   5 -65   0 722   1 -81
   -57   6 999


 SCON( 31,-81) =  101
 -66 SWITCH TEST: CONDITION TRUE AT   4
 BRANCH TO -57  6   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 453

Main 40 table #179
   -66 899 199 -81 246   0 -81   1  -5 -81   1   5  60
   -57   1
   -66 899  56 -81  46 319 -81  46 101 -81  19 140  60 -55   5 -81  31 -55   6 -81   5
   -66 348  56 -81  31 315 777 -81   5   2 777 -81 246 101  60
   -66 348  56 -81  13  11 777 -81  31 115  60
   -56   5 678  56   5 115 777 777  67 909 777 777   6   2
   -56   3 238  56  67 909 777 777   5 115
   -56   7  56 499 467   0 457   0  15 968  15 977  15 967  15 976  15 942
   -66  56 499 -81  46   0 777 -81  47   0  60
   -66  56 499 -81  31 315  60
   -66 348 899 -81   5   2  60
   -57   2 -44 -81 107 532   0 -48  43 -81   6 -54   1 -81  46 115
   -57   3 -44 -81 107 140   0 -54   1 -81  46 140 -48  43 -81   9
   -57   4
   -66 568  56 -81  31 115 777 -81   5   2  60
   -66 799  56 -81  31 115 777 -81 240   0  60
   -66 799 899 -81  31 115  60
   -57   5 -44 -81 107 456   0 -48  43 -81   6
   -57   6 -44 -81 107 341   0 -48  43 -81   6 -54   1 -81  46 341
   -57   7 -44 -81 107 621   0 -48  43 -81   3 -54   1 -81  46 621
   -57   8 999
```

```
SCON( 46,-81) =    0
SCON(  1,-81) =    1
SCON(  1,-81) =    1
-66 SWITCH TEST: CONDITION FALSE
BRANCH TO -57  1   EXECUTE UNTIL -57 99  JUMP -57 99
SW57 - VTR BREAK POINT, K3:  14
SCON( 46,-81) =    0
SCON( 46,-81) =    0
SCON( 19,-81) =    0
-66 056 CONDITION AT  25,                              CONTINUE TO THE RIGHT
SW55 - LOADED CELL:  5 WITH VALUE:   101, VBRELP =  2
SW55 - LOADED CELL:  6 WITH VALUE:     0, VBRELP =  2
SCON( 31,-81) =  101
-66 056 CONDITION AT  48,                              CONTINUE TO THE RIGHT
SCON( 13,-81) =    9
-66 056 CONDITION AT  59,                              CONTINUE TO THE RIGHT
CELL  5  =  101
CELL 67  =    0
CELL  6  =    0
-56 056 CONDITION AT  75,                              CONTINUE TO THE RIGHT
CELL 67  =    0
CELL  5  =  101
-56 056 CONDITION AT  85,                              CONTINUE TO THE RIGHT
CELL 67  =    0
CELL 57  =    0
CELL 15  =    0
CELL 15  =    0
CELL 15  =    0
CELL 15  =    0
CELL 15  =    0
-56 SWITCH TEST: CONDITION FALSE
BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99
SW57 - VTR BREAK POINT, K3: 162
SCON( 31,-81) =  101
-66 056 CONDITION AT 171,                              CONTINUE TO THE RIGHT
SCON( 31,-81) =  101
-66 056 CONDITION AT 182,                              CONTINUE TO THE RIGHT
SCON( 31,-81) =  101
-66 SWITCH TEST: CONDITION FALSE
BRANCH TO -57  8   EXECUTE UNTIL -57 99  JUMP -57 99
SW57 - VTR BREAK POINT, K3: 236
SW57 - VTR BREAK POINT, K3:  88
SCON( 20,-81) =    1
-66 056 CONDITION AT  93,                              CONTINUE TO THE RIGHT
CELL 19  =    1
CELL 17  =    0
-56 056 CONDITION AT 105,                              CONTINUE TO THE RIGHT
```

```
CELL 19  =    1
CELL 17  =    0
-56 056 CONDITION AT 115,                              CONTINUE TO THE RIGHT
CELL 18  =    3
CELL 19  =    1
-56 056 CONDITION AT 125,                              CONTINUE TO THE RIGHT
CELL 18  =    3
CELL 19  =    1
-56 056 CONDITION AT 135,                              CONTINUE TO THE RIGHT
CELL 18  =    3
CELL 19  =    1
-56 056 CONDITION AT 148,                              CONTINUE TO THE RIGHT
SCON( 19,-81) =    0
SCON( 20,-81) =    1
-66 056 CONDITION AT 156,                              CONTINUE TO THE RIGHT
CELL 43  =    1
CELL 18  =    3
CELL 17  =    0
-56 056 CONDITION AT 172,                              CONTINUE TO THE RIGHT
CELL 43  =    1
CELL 18  =    3
CELL 17  =    0
-56 056 CONDITION AT 186,                              CONTINUE TO THE RIGHT
CELL 43  =    1
CELL 18  =    3
CELL 17  =    0
-56 056 CONDITION AT 200,                              CONTINUE TO THE RIGHT
CELL 43  =    1
CELL 18  =    3
CELL 17  =    0
-56 056 CONDITION AT 214,                              CONTINUE TO THE RIGHT
SCON( 20,-81) =    1
-66 056 CONDITION AT 219,                              CONTINUE TO THE RIGHT
SCON( 20,-81) =    1
-66 056 CONDITION AT 229,                              CONTINUE TO THE RIGHT
SCON( 19,-81) =    0
-66 056 CONDITION AT 239,                              CONTINUE TO THE RIGHT
SCON( 19,-81) =    0
-66 056 CONDITION AT 246,                              CONTINUE TO THE RIGHT
SCON( 20,-81) =    1
-66 056 CONDITION AT 253,                              CONTINUE TO THE RIGHT
SCON( 20,-81) =    1
-66 056 CONDITION AT 260,                              CONTINUE TO THE RIGHT
CELL 17  =    0
-56 SWITCH TEST: CONDITION FALSE
BRANCH TO -57  6   EXECUTE UNTIL -57  7  JUMP -57 99
SW57 - VTR BREAK POINT, K3: 329
```

```
 SW55 - LOADED CELL: 29 WITH VALUE:  91
 SW57 - VTR BREAK POINT, K3: 341
 SW57 - CONDITIONAL EXECUTION COMPLETED, BRANCH TO:  385
 SW57 - VTR BREAK POINT, K3: 100
 SW57 - CONDITIONAL EXECUTION COMPLETED, BRANCH TO:  130
 SW57 - VTR BREAK POINT, K3: 130

***** A MATCH STARTING AT   3 LEVEL   2       ON ELEMENT  4jj       tran4
Tran rule #423, ID: 422
BE PREDADJ = -2 S487 E4
 2  (3 886 -1)  (1 -1 90)
   -63   1 703   1 -36  56   0 -41   2 999     0   0


Main 30 table #1703
   -66 699  56 -81  19 140  60 -64   0 386   2 -81 -82
   -66 199  56 -81  31 460 777 -81 217  15  60
   -66 199 299 -81  31 461 777 -81 217  15  60
   -57   1 -54   1 -81  46  87
   -57   2 -16 -81 -81 -81   0 -82
   -66 399 499 -81  20 896  60
   -57   3 -55  21   0   0 -54   1 -81  28 482 -11  83 482   0 -31  11
   -57   4
   -66 567 799 -81  19  30 777 -81  53 735 777 -82   2 428  60
   -57   5 -54   1 -82  20 140
   -57   6 -36 140 -81
   -57   7
   -66  56 899 -81   1  12  60 -55  21 -81   2
   -57   8 999


 SCON( 19,-81) =    1
 -66 056 CONDITION AT   4,                              CONTINUE TO THE RIGHT

Main 40 table #386
   -66 199 299 -81  19 191  60
   -57   1 -44   2 -81   1 117 102 -48 102 -81 -82
   -57   2 999


 SCON( 19,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  20
 SCON( 31,-81) =    0
 -66 056 CONDITION AT  21,                              CONTINUE TO THE RIGHT
 SCON( 31,-81) =    0
```

```
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  43
 SCON( 20,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  75
 SCON( 19,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  7   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 104
 SCON(  1,-81) =    2
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  8   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 117

***** A MATCH STARTING AT   3 LEVEL   1       ON ELEMENT  4jj       tran4
Tran rule #380, ID: 379
V(ACT) = -1/ STS286 E4
 1  (3 -1 96)
   -63   1 288   1 -36  56   0 -41   1 999     0   0


Main 30 table #1288
   -55  11 -81  11 -55  28 -81   2
   -56   1 199 499  16 591
   -57   1
   -56   1 399 299  20   1
   -57   2 -55  16 592   0 -11  81 291   0
   -57   3 -11  93 122   0 -31  11
   -57   4 999


 SW55 - LOADED CELL: 11 WITH VALUE:    60, VBRELP =  3
 SW55 - LOADED CELL: 28 WITH VALUE:   886, VBRELP =  3
 CELL 16  =    0
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  41

***** A MATCH STARTING AT   3 LEVEL   2       ON ELEMENT  4jj       tran4
Tran rule #229, ID: 228
**31 (MC) V N = -2(SICÜ) / MC CHAIN/ E4 ST286 MMT0289
 2  (3 -1 -1)  (1 -1 -1)
   -42  10  31   1   2 -46 -82   6   0   0 -46 -81   2 851   0
   -63   1 283   1 -41   2 999    -1  -1
```

```
***** A MATCH STARTING AT   3 LEVEL   3       ON ELEMENT  4jj       tran4
Tran rule #1218, ID: 1217
02***31(MC) V N (ACC) = -2/CK FOR BE-TYPE VERBS E4 MMT1288
 3  (10 31 1)  (3 11 -1)  (1 -2 -1)
  { 6010   81   60   61    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0 }
   -63   2 168   1 -55   2 -82 351
   -56   7 123 299   2  90   2  53   2  43   2  86   2  96   2  88     2  79
   -57   1 -46 -82   6   0   0
   -57   2 -46 -82   6   0  81
   -57   3 -46 -81   2 851   0 -41     2 999     0


Main 30 table #2168
   -64   0 386   2 -81 -82 -64   0 275   0
   -66 699  56 -81  19 140 -82  20 140  60
   -66 699  56 -82   2 175 777 -82  13   0  60
   -66 199  56 -81  31 461 777 -81   2 710 777 -82  31   0  60 -65   0 627   1 -82
   -66 199 299 -81  31 461 -81  31 460  60
   -57   1 -64   0  71   2 -81 -82
   -57   2
   -66  56 399 -82  20 101  60 -36  53 -82 -54   1 -82   7   3 -44 -82 102 134   0
   -57   3
   -56   1  56 699  16 591
   -56   1 599 499  20   1
   -57   4 -11  81 291   0
   -57   5 -55  16 592   0
   -57   6 999



Main 40 table #386
   -66 199 299 -81  19 191  60
   -57   1 -44   2 -81   1 117 102 -48 102 -81 -82
   -57   2 999


 SCON( 19,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  20

Main 40 table #275
   -64   0 258   0 -55   5 -81 351
   -56   1 299 199   5  39
   -57   1 -55  29   0   0 -55  19 -81  19 -55  43 -81   3
   -57   2
```

```
   -66  56 499 -82   2 175 777 -82  13   0  60
   -66 399 499 -82  19 401 -82  19 402 -82  19 392  60
   -57   3 -54   1 -81  48  16
   -57   4 999



Main 40 table #258
   -56   1  56 299  28 705 -55   5 -82 351
   -56   1  56 299   5  90
   -66  56 299 -81  20  31  60
   -66 199 299 -82   2 180 -82   2 181 -82   2 183  60
   -57   1 -44 -81 120 496   0
   -57   2 999


 CELL 28  =  886
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  44
 SW55 - LOADED CELL:  5 WITH VALUE:     3, VBRELP =  3
 CELL  5  =    3
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  1   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  15
 SW55 - LOADED CELL: 29 WITH VALUE:   0
 SW55 - LOADED CELL: 19 WITH VALUE:     1, VBRELP =  3
 SW55 - LOADED CELL: 43 WITH VALUE:     1, VBRELP =  3
 SW57 - VTR BREAK POINT, K3:  29
 SCON(  2,-82) =  609
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  62
 SCON( 19,-81) =    1
 SCON( 20,-82) =    1
 -66 056 CONDITION AT  17,                              CONTINUE TO THE RIGHT
 SCON(  2,-82) =  609
 -66 056 CONDITION AT  28,                              CONTINUE TO THE RIGHT
 SCON( 31,-81) =    0
 -66 056 CONDITION AT  43,                              CONTINUE TO THE RIGHT

Main 50 table #627
   -66 699  56 -82  31 101 -82  46 101 -82  62 849 -82  62 864 -82  62 848 -82  62 851 -82  46 621 -82  46 341 -82   9
11 -82   9  21 -82   9  31 -82   9  19 -82   9  29 -82   9  39  60
   -66 699  56 -82   2 303 777 -82 213   4 777 -82 213   7  60 -55   3 -82 350
   -56   1 699  56   3 175
   -66 699  56 -82  46 319 -82  46 101 -82  19 140  60
```

```
   -66 699  56 -82  43 140  60 -55   6 -82  31 -55   5 -82 351
   -56   3 699  56   6   0 777 777   5  90 -55   1 -82   2 -55   2 -82  11 -55   8 -82  13 -55   9 -82  46
   -56   7 456  56   6   0 777 777   9   0 777 777   2  21 777 777  43   2
   -56   9 456  56   5  91 777 777   1 855 777 777   2  94 777 777   8   5 777 777   6   0
   -66 699  56 -82  60 140 -82  20 140  60
   -56   2 699  56   5  43   5  55
   -56   1  56 399  15 942
   -66 126  56 -82  31 115 777 -82   5   2  60
   -66 236 399 -82  31 115 777 -82   5   1  60
   -57   1 -44 -82 107 140   0 -54   1 -82  46 140 -48  43 -82   9
   -57   2 -44 -82 107 532   0 -54   1 -82  46 315 -48  43 -82   6
   -57   3 -55   3 -82 352
   -56   1 699  56   3  23
   -66 699  56 -82  31 115 777 -82 246   0  60
   -66 699  56 -82 239   0 -82  42 575  60
   -66 699  56 -82  62 850 -82  62 864  60
   -66 699  56 -82   1   5  60
   -66  56 699 -82  31   0  60
   -66  56 699 -82   5   1 -82   5   0  60
   -66  56 699 -82  32   0 777 -82  33   0 777 -82  35   0 777 -82  36   0 777 -82  41   0 777 -82  46   0  60
   -66 699  56 -82  11  21 -82  11  52  60
   -66 456  56 -82  13   4 -82  13   7  60
   -66 456  56 -82   2  23 -82   2 582 -82   2 327 -82   2 655 -82   2 602 -82   2  78 -82   2 702 -82   2  50 -82   2
173 -82   2 708 -82   2 749 -82   2 574 -82   2 450 -82   2 716 -82   2 297  60
   -66  56 699 -82   2  46 -82   2  49 -82   2 609  60
   -66 456 699 -82  17  58 -82  42   0  60
   -57   4 -48  43 -82   3 -44 -82 107 131   0 -54   1 -82  46 101
   -57   5 -65   0 722   1 -82
   -57   6 999


 SCON( 31,-82) =    0
 SCON( 46,-82) =    0
 SCON( 62,-82) =    0
 SCON( 62,-82) =    0
 SCON( 62,-82) =    0
 SCON( 62,-82) =    0
 SCON( 46,-82) =    0
 SCON( 46,-82) =    0
 SCON(  9,-82) =    0
 SCON(  9,-82) =    0
 SCON(  9,-82) =    0
 SCON(  9,-82) =    0
 SCON(  9,-82) =    0
 SCON(  9,-82) =    0
 -66 056 CONDITION AT  43,                          CONTINUE TO THE RIGHT
 SCON(  2,-82) =  609
```

```
 -66 056 CONDITION AT  58,                                CONTINUE TO THE RIGHT
 SW55 - LOADED CELL:  3 WITH VALUE:   609, VBRELP =  4
 CELL  3  =  609
 -56 056 CONDITION AT  70,                                CONTINUE TO THE RIGHT
 SCON( 46,-82) =    0
 SCON( 46,-82) =    0
 SCON( 19,-82) =    0
 -66 056 CONDITION AT  81,                                CONTINUE TO THE RIGHT
 SCON( 43,-82) =    0
 -66 056 CONDITION AT  88,                                CONTINUE TO THE RIGHT
 SW55 - LOADED CELL:  6 WITH VALUE:     0, VBRELP =  4
 SW55 - LOADED CELL:  5 WITH VALUE:    90, VBRELP =  4
 CELL  6  =    0
 CELL  5  =   90
 -56 SWITCH TEST: CONDITION TRUE AT 108
 BRANCH TO -57  6   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 453
 SCON( 31,-81) =    0
 SCON( 31,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  70
 SCON( 20,-82) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  92
 CELL 16  =    0
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  6   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 118
 SW55 - LOADED CELL:  2 WITH VALUE:    90, VBRELP =  4
 CELL  2  =   90
 -56 SWITCH TEST: CONDITION TRUE AT  13
 BRANCH TO -57  1   EXECUTE UNTIL -57  2  JUMP -57  3
 SW57 - VTR BREAK POINT, K3:  27
 SW57 - VTR BREAK POINT, K3:  34
 SW57 - CONDITIONAL EXECUTION COMPLETED, BRANCH TO:   41
 SW57 - VTR BREAK POINT, K3:  41

***** A MATCH STARTING AT   3 LEVEL   1       ON ELEMENT  4jj       tran4
Tran rule #110, ID: 109
(MC) V(PRES-PUNCT-ACT) = (V) / CK PASSACT ST386 EGSP4
 1  (2 851 51)
   -63   0  40   1 999    -1  -1


Main 30 table #40
```

```
   -66 123 299 -81  19 140  60
   -57   1 -27 -81  -1 140
   -57   2 -65   0 253   0
   -57   3 999


 SCON( 19,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  14

Main 50 table #253
   -55   3 -81  20
   -56   1 950 839   3 140 -55   5 -81  19
   -56   1 950 575   5  65
   -56   1 950 576   5  66
   -56   1 950 518   5  54
   -66 950 576 -81  20 273  60 -65 100 501   0  -1   0 999


 SW55 - LOADED CELL:  3 WITH VALUE:     1, VBRELP =  3
 CELL  3  =    1
 -56 CONDITION FALSE, CONTINUE THIS VTR
 SW55 - LOADED CELL:  5 WITH VALUE:     1, VBRELP =  3
 CELL  5  =    1
 -56 CONDITION FALSE, CONTINUE THIS VTR
 CELL  5  =    1
 -56 CONDITION FALSE, CONTINUE THIS VTR
 CELL  5  =    1
 -56 CONDITION FALSE, CONTINUE THIS VTR
 SCON( 20,-81) =    1
 -66 CONDITION FALSE, CONTINUE THIS VTR

Main 50 table #501
   -64   0  80   1 -81
   -66 950 840 -81  59 336  60 -55   3 -81  12
   -56   2 126  56   3   3   3   5
   -56   1 236  56   3   4
   -56   1 347  56   3   7
   -56   1 457  56   3   8
   -56   1 567 699   3   9
   -57   1 -11  81 271   0 -16   1 -81 -81   3 271
   -57   2 -11  81 271   0 -16   1 -81 -81   4 271
   -57   3 -11  92 -38   4  -1   0 -11  82 330   0 -16 -81 -81 -81   1 330
   -57   4 -11  92 -38   4  -1   0 -11  82 330   0 -16 -81 -81 -81   1 330 -11  81 271   0 -16   1 -81 -81   3 271
   -57   5 -11  92 -38   4  -1   0 -11  82 330   0 -16 -81 -81 -81   1 330 -11  81 271   0 -16   1 -81 -81   4 271
   -57   6 -11  82 -38   1  -1   0
```

```
   -57   7 -11  91 -27 -81
   -57   8
   -66 950 514 -81  48  16  60
   -66 950 517 -81  48  19  60 999



Main 40 table #80
   -66 399  56 -81  60 140 -81  59   0  60
   -66 123  56 -81  15   1 -81  15   2  60
   -66 123  56 -81  59   9 -81  59  13 -81  59  14 -81  59  15 -81  59  16 -81  59  32  60
   -66 123  56 -81  59  33 -81  59  34 -81  59  35 -81  59  36 -81  59  37 -81  59  38  60
   -66 123  56 -81  59  39 -81  59  40 -81  59  41 -81  59  42 -81  59  43 -81  59  44  60
   -66 123  56 -81  59  45 -81  59  46 -81  59  48 -81  59  49 -81  59  58 -81  59  66  60
   -66 123  56 -81  59  80 -81  59 100 -81  59 121 -81  59 130 -81  59 133 -81  59 136  60
   -66 123  56 -81  59 150 -81  59 151 -81  59 152 -81  59 153 -81  59 154 -81  59 155  60
   -66 123  56 -81  59 156 -81  59 157 -81  59 169 -81  59 171 -81  59 181 -81  59 182  60
   -66 123  56 -81  59 267 -81  59 269 -81  59 271 -81  59 273 -81  59 278 -81  59 281  60
   -66 123  56 -81  59 283 -81  59 287 -81  59 289 -81  59 290 -81  59 291 -81  59 293  60
   -66 123  56 -81  59 296 -81  59 301 -81  59 303 -81  59 306 -81  59 308 -81  59 309  60
   -66 123  56 -81  59 310 -81  59 311 -81  59 316 -81  59 319 -81  59 325 -81  59 328  60
   -66 123 299 -81  59 510 -81  59 530 -81  59 543 -81  59 544 -81  59 545  60
   -57   1 -54   1 -81  59 336 -55  59 336   0
   -57   2 -64   0 393   1 -81
   -57   3 999


 SCON( 60,-81) =    0
 SCON( 59,-81) =    1
 -66 056 CONDITION AT   7,                          CONTINUE TO THE RIGHT
 SCON( 15,-81) =    0
 SCON( 15,-81) =    0
 -66 056 CONDITION AT  17,                          CONTINUE TO THE RIGHT
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 -66 056 CONDITION AT  39,                          CONTINUE TO THE RIGHT
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 -66 056 CONDITION AT  61,                          CONTINUE TO THE RIGHT
```

```
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT  83,                              CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 105,                              CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 127,                              CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 149,                              CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 171,                              CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 193,                              CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
```

```
 SCON( 59,-81) =    1
 -66 056 CONDITION AT 215,                              CONTINUE TO THE RIGHT
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 -66 056 CONDITION AT 237,                              CONTINUE TO THE RIGHT
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 -66 056 CONDITION AT 259,                              CONTINUE TO THE RIGHT
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 293

Main 40 table #393
   -66 199  56 -81  59 188 -81  59 189 -81  59 196 -81  59 203 -81  59 205 -81  59 206  60
   -66 199  56 -81  59 207 -81  59 209 -81  59 210 -81  59 229 -81  59 231 -81  59 237  60
   -66 199  56 -81  59 240 -81  59 246 -81  59 248 -81  59 250 -81  59 253 -81  59 256  60
   -66 199  56 -81  59 330 -81  59 331 -81  59 333 -81  59 338 -81  59 340 -81  59 343  60
   -66 199  56 -81  59 346 -81  59 348 -81  59 350 -81  59 352 -81  59 355 -81  59 358  60
   -66 199  56 -81  59 360 -81  59 363 -81  59 365 -81  59 368 -81  59 371 -81  59 375  60
   -66 199  56 -81  59 378 -81  59 384 -81  59 387 -81  59 388 -81  59 389 -81  59 391  60
   -66 199  56 -81  59 394 -81  59 396 -81  59 397 -81  59 398 -81  59 399 -81  59 400  60
   -66 199  56 -81  59 401 -81  59 402 -81  59 403 -81  59 405 -81  59 406 -81  59 407  60
   -66 199  56 -81  59 408 -81  59 409 -81  59 410 -81  59 413 -81  59 438 -81  59 430  60
   -66 199  56 -81  59 435 -81  59 436 -81  59 439 -81  59 441 -81  59 442 -81  59 444  60
   -66 199  56 -81  59 445 -81  59 447 -81  59 448 -81  59 450 -81  59 451 -81  59 452  60
   -66 199  56 -81  59 454 -81  59 456 -81  59 459 -81  59 462 -81  59 480 -81  59 481  60
   -66 199  56 -81  59 139 -81  59 199 -81  59 234 -81  59 243 -81  59 259 -81  59 299  60
   -66 199 299 -81  59 313 -81  59 322 -81  59 596 -81  59 602 -81  59 643  60
   -57   1 -54   1 -81  59 336 -55  59 336   0
   -57   2 999


 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
```

```
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT  19,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT  41,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT  63,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT  85,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 107,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 129,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 151,                          CONTINUE TO THE RIGHT
```

```
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 173,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 195,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 217,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 239,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 261,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
-66 056 CONDITION AT 283,                          CONTINUE TO THE RIGHT
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
SCON( 59,-81) =    1
```

```
 SCON( 59,-81) =    1
 -66 056 CONDITION AT 305,                                CONTINUE TO THE RIGHT
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 SCON( 59,-81) =    1
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 339
 SW57 - VTR BREAK POINT, K3: 300
 SCON( 59,-81) =    1
 -66 CONDITION FALSE, CONTINUE THIS VTR
 SW55 - LOADED CELL:  3 WITH VALUE:     2, VBRELP =  3
 CELL  3  =    2
 CELL  3  =    2
 -56 056 CONDITION AT  23,                                CONTINUE TO THE RIGHT
 CELL  3  =    2
 -56 056 CONDITION AT  29,                                CONTINUE TO THE RIGHT
 CELL  3  =    2
 -56 056 CONDITION AT  35,                                CONTINUE TO THE RIGHT
 CELL  3  =    2
 -56 056 CONDITION AT  41,                                CONTINUE TO THE RIGHT
 CELL  3  =    2
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  6   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 147
 SW57 - VTR BREAK POINT, K3: 155
 SW57 - VTR BREAK POINT, K3: 161
 SCON( 48,-81) =    0
 -66 CONDITION FALSE, CONTINUE THIS VTR
 SCON( 48,-81) =    0
 -66 CONDITION FALSE, CONTINUE THIS VTR

***** A MATCH STARTING AT   4 LEVEL   2       ON ELEMENT  5jj       tran4
Tran rule #689, ID: 688
(MC) N PUNC(CB) = -2(SICÜ) / (MC)CHAIN JV582 EGSP4
 2  (6 -1 -1)  (20 -1 -1)
   -63   1 948   3 -46 -82  16   0   0 -46 -81   1 851   0 -41   2 999     0   0


Main 30 table #1948
   -64   0 357   2 -81 -82 -64   0 382   2 -81 -82
   -66 127  56 -81   2 733 777 -81  31 102 777 -82   2 887 777 -81 205   2  60
   -66 457  56 -82   2 947  60 -55   5 -81 351
   -56   1  56 399   5  50
   -66 235  56 -81   2 596 777 -81  11  90 777 -82   2 968  60
```

```
   -66 456 399 -82  11  20 -82   1  20  60
   -57   1 -44 -81 107 140   0 -36 750 -81 -13 -81
   -57   2 -32 -99   8 -36 933 -81 -13 -81 -64   0  83   0 122   0
   -57   3
   -66 599  56 -82   2 410  60
   -66 567  56 -81   2 873  60
   -66 699  56 -82 246   0  60
   -66 457  56 -82  11  20  60
   -66 499  56 -82   2   1 -82   2 909 -82   2 186 -82   2 185 -82   2 189 -82   2 887  60
   -66 499  56 -82   2 886 -82   2 884 -82   2   5 -82   2 866 -82   2 386 -82   2 877  60
   -66 499  56 -82   2 885 -82   2 387 -82   2 777 -82   2 130 -82   2 826  60
   -66 499  56 -82   2 829 -82   2 830  60
   -66  56 799 -82   2 850  60
   -66 457 799 -81  45   2 -81  12   2  60
   -57   4 -54   1 -82  46 140
   -57   5 -54   1 -81  48  16
   -57   6 -54   1 -82   4 -81 -54   1 -82   5 -81
   -57   7
   -66  56 899 -82   2 888  60 -55   5 -81 351
   -56   1 899  56   5  53
   -56   1  56 899 505  85
   -56   2  56 899  18   5  18  35 -54   1 -82  82   0 -54   1 -82  83  21
   -57   8 999


Main 40 table #357
   -55   5 -81 351
   -56   1  56 299   5  43
   -66 126 299 -81   2 274 777 -81   1   6 777 -82   1  19 777 -82   2 911  60
   -57   1 -36 673 -81
   -57   2
   -66 346 499 -81  13  12 777 -81   2 945 777 -82   2 915 777 -82  60 309  60
   -57   3 -54   1 -82  46 309 -33 -81
   -57   4
   -66  56 699 -82   2 887  60
   -66  56 699 -81   1  11 -81   1  13 -81   1   3  60
   -66 599 699 -81   2 481 -81   2 945 -81   2 703 -81   2 132 -81   2 159 -81   2 522  60
   -57   5 -54   1 -81  20  93
   -57   6 999


 SW55 - LOADED CELL:  5 WITH VALUE:    90, VBRELP =  4
 CELL  5  =   90
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  35
```

```
 SCON( 13,-81) =    6
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  4   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  65
 SCON(  2,-82) =   10
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  6   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 116

Main 40 table #382
   -66 199 299 -81  46 144  60
   -57   1 -54   1 -82  46 144
   -57   2 999


 SCON( 46,-81) =    0
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  15
 SCON(  2,-81) =  609
 -66 056 CONDITION AT  28,                              CONTINUE TO THE RIGHT
 SCON(  2,-82) =   10
 -66 056 CONDITION AT  35,                              CONTINUE TO THE RIGHT
 SW55 - LOADED CELL:  5 WITH VALUE:    90, VBRELP =  4
 CELL  5  =   90
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 102
 SCON(  2,-82) =   10
 -66 056 CONDITION AT 107,                              CONTINUE TO THE RIGHT
 SCON(  2,-81) =  609
 -66 056 CONDITION AT 114,                              CONTINUE TO THE RIGHT
 SCON( 46,-82) =    0
 -66 056 CONDITION AT 121,                              CONTINUE TO THE RIGHT
 SCON( 11,-82) =    0
 -66 056 CONDITION AT 128,                              CONTINUE TO THE RIGHT
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 -66 056 CONDITION AT 150,                              CONTINUE TO THE RIGHT
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
```

```
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 -66 056 CONDITION AT 172,                              CONTINUE TO THE RIGHT
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 -66 056 CONDITION AT 191,                              CONTINUE TO THE RIGHT
 SCON(  2,-82) =   10
 SCON(  2,-82) =   10
 -66 056 CONDITION AT 201,                              CONTINUE TO THE RIGHT
 SCON(  2,-82) =   10
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  7   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 248
 SCON(  2,-82) =   10
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  8   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 291

***** A MATCH STARTING AT   4 LEVEL   1       ON ELEMENT  5jj       tran4
Tran rule #26, ID: 25
(MC) N(PRED ADJ) = (N) / S87 ST386 EGSP4
 1  (1 851 90)
   -63   0   5   1 999    -1  -1


Main 30 table #5
   -66 199 299 -81  13  13 777 -81  31 148  60
   -57   1 -44 -81 107 140   0 -54   1 -81  12   4 -54   1 -81  31   0
   -57   2 -65   0 223   0 999


 SCON( 13,-81) =    6
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  29

Main 50 table #223
   -66 950 971 -81  47 123  60
   -66 950 917 -81  19  51  60
   -66 950 548 -81 231   0  60
   -66 950 649 -81  20 273  60 -55   1 -81   2
   -66 799  56 -81   1   2 777 -81   7   6  60 -55   5 -81  19 -55   3 -81  20
   -56   1 128  56   5 140
   -56   1 237  56   3 140
```

```
  -56   3 568  56   1 206 777 777  27   5
  -56   1 458  56   5  51
  -66 568  56 -81  20 187  60
  -66 348  56 -81  20   5  60
  -66 568  56 -81  20  91  60
  -66 678 348 -81  12   5 777 -81  36   3  60
  -57   1 -36  53 -81 -36 140 -81 -11  87  -1   0 -31  11
  -57   2 -11  99  -1   0
  -57   3 -55  29  87   0 -11  87 -38   5  -1   0 -31  11
  -57   4 -55  29  87   0 -11  87 -38   4  -1   0 -13 -81 864   0 607   0 -16   0   0   0   5 607
  -57   5 -55  29  87   0 -11  87 -38   1  -1   0 -31  11
  -57   6 -44 -81 107 140   0 -11  87 455   0 -38   5  -1   0 -31  11
  -57   7 -55  29  87   0 -11  87 -38   6  -1   0 -31  11
  -57   8
  -66 950 514 -81  48  16  60
  -66 950 517 -81  48  19  60 999


SCON( 47,-81) =    0
-66 CONDITION FALSE, CONTINUE THIS VTR
SCON( 19,-81) =    0
-66 CONDITION FALSE, CONTINUE THIS VTR
SCON( 31,-81) =    0
-66 CONDITION FALSE, CONTINUE THIS VTR
SCON( 20,-81) =    1
-66 CONDITION FALSE, CONTINUE THIS VTR
SW55 - LOADED CELL:  1 WITH VALUE:   609, VBRELP =  4
SCON(  1,-81) =    1
-66 056 CONDITION AT  40,                              CONTINUE TO THE RIGHT
SW55 - LOADED CELL:  5 WITH VALUE:     0, VBRELP =  4
SW55 - LOADED CELL:  3 WITH VALUE:     1, VBRELP =  4
CELL  5  =    0
-56 056 CONDITION AT  56,                              CONTINUE TO THE RIGHT
CELL  3  =    1
-56 056 CONDITION AT  62,                              CONTINUE TO THE RIGHT
CELL  1  =  609
CELL 27  =    9
-56 056 CONDITION AT  72,                              CONTINUE TO THE RIGHT
CELL  5  =    0
-56 056 CONDITION AT  78,                              CONTINUE TO THE RIGHT
SCON( 20,-81) =    1
-66 056 CONDITION AT  83,                              CONTINUE TO THE RIGHT
SCON( 20,-81) =    1
-66 056 CONDITION AT  90,                              CONTINUE TO THE RIGHT
SCON( 20,-81) =    1
-66 056 CONDITION AT  97,                              CONTINUE TO THE RIGHT
SCON( 12,-81) =    1
```

```
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57  4  JUMP -57  8
 SW57 - VTR BREAK POINT, K3: 132
 SW55 - LOADED CELL: 29 WITH VALUE:  87
 SW57 - VTR BREAK POINT, K3: 146
 SW57 - CONDITIONAL EXECUTION COMPLETED, BRANCH TO:  215
 SW57 - VTR BREAK POINT, K3: 215
 SCON( 48,-81) =    0
 -66 CONDITION FALSE, CONTINUE THIS VTR
 SCON( 48,-81) =    0
 -66 CONDITION FALSE, CONTINUE THIS VTR

***** A MATCH STARTING AT   5 LEVEL   1       ON ELEMENT  6jj       tran4
Tran rule #1598, ID: 1597
(MC) CB = -1 (SICÜ) / UNLOAD SLOTS STS1085 EGSP4
 1  (16 -1 -1)
   -46 -81  20   0   0
   -63   0 460   1 -41   1 999     0   0


Main 30 table #460
   -55   5 -81 350 -55   6 -81 351
   -56   1 199  56   6  16
   -66 199  56 -81  20 961  60
   -66 199  56 -81  20  93 777 -81   1  13  60
   -66 199  56 -81  20  93 777 -81   1  11  60
   -66 299  56 -81  60 140 777 -81  46 140  60
   -66 299  56 -81  46 144  60
   -66  56 199 -81   2 820 -81   2 828  60
   -66 299 199 -81  20  88 -81  20  92 -81  20  53  60
   -57   1  91   0  82   0  81   0  85   0 113   0  83   0 118   0 111   0 115   0  88   0 119   0  96   0  89   0 114
0  84   0 116   0  86   0 120   0 -27   1  90   0  87   0  92   0  97   0  98   0  93   0 -55  29   0   0
   -57   2 -55  22   0   0 -55  42   0   0 -55  16   0   0 -55   5 -81 350
   -56   2 499 399   5 888   5 900
   -57   3 -55  15 -81 350
   -57   4 999


 SW55 - LOADED CELL:  5 WITH VALUE:    10, VBRELP =  5
 SW55 - LOADED CELL:  6 WITH VALUE:     1, VBRELP =  5
 CELL  6  =    1
 -56 056 CONDITION AT  13,                              CONTINUE TO THE RIGHT
 SCON( 20,-81) =    0
 -66 056 CONDITION AT  18,                              CONTINUE TO THE RIGHT
 SCON( 20,-81) =    0
 -66 056 CONDITION AT  29,                              CONTINUE TO THE RIGHT
 SCON( 20,-81) =    0
```

```
 -66 056 CONDITION AT  40,                              CONTINUE TO THE RIGHT
 SCON( 60,-81) =    0
 -66 056 CONDITION AT  51,                              CONTINUE TO THE RIGHT
 SCON( 46,-81) =    0
 -66 056 CONDITION AT  58,                              CONTINUE TO THE RIGHT
 SCON(  2,-81) =   10
 SCON(  2,-81) =   10
 -66 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  1   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  85
 SW38     3   14    1    1    3    3
 SW38    15   25    1    1    4    4
 SW38    26   36    1    1    5    5
 SW55 - LOADED CELL: 29 WITH VALUE:   0
 SW57 - VTR BREAK POINT, K3: 141
 SW55 - LOADED CELL: 22 WITH VALUE:   0
 SW55 - LOADED CELL: 42 WITH VALUE:   0
 SW55 - LOADED CELL: 16 WITH VALUE:   0
 SW55 - LOADED CELL:  5 WITH VALUE:    10, VBRELP =  5
 CELL  5  =   10
 CELL  5  =   10
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  3   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 167
 SW55 - LOADED CELL: 15 WITH VALUE:    10, VBRELP =  5
 SW57 - VTR BREAK POINT, K3: 173

***** A MATCH STARTING AT   5 LEVEL   1       ON ELEMENT  6jj       tran4
Tran rule #1975, ID: 1974
EOS = EOS / EMPTY SLOTS ST286 EGSP4
 1  (20 10 -1)
   -63   0 533   1 999     0   0


Main 30 table #533
   -64   0 199   0
   -56   1 123 299  67 909
   -57   1  72   0  91   0  82   0  81   0  85   0  83   0  88   0  96   0  89   0  84   0  86   0  90   0  87   0  92
0  97   0  98   0  93   0  -1   0
   -57   2  72   0  91   0  81   0  85   0  83   0  88   0  89   0  96   0  84   0  86   0  90   0  87   0  92   0  82
0  97   0  98   0  93   0  -1   0
   -57   3 999



Main 40 table #199
   -56   9 299  56 418   0 777 777 418  35 777 777 418  38 777 777 418  28 777 777 418   5
```

```
   -56   9 299 199 438   0 777 777 438  35 777 777 438  38 777 777 438  28 777 777 438   5
   -57   1 -67  55  18  18   1 -67  55  38  38   1
   -56   9 299  56 418   0 777 777 418  35 777 777 418  38 777 777 418  28 777 777 418   5
   -56   9 299 399 438   0 777 777 438  35 777 777 438  38 777 777 438  28 777 777 438   5
   -57   2 -67   6  18  38 -96 -81
   -57   3 999


 CELL 18  =    3
 CELL 18  =    3
 CELL 18  =    3
 CELL 18  =    3
 CELL 18  =    3
 -56 SWITCH TEST: CONDITION TRUE AT  21
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3: 101
 SW57 - VTR BREAK POINT, K3: 109
 CELL 67  =    0
 -56 SWITCH TEST: CONDITION FALSE
 BRANCH TO -57  2   EXECUTE UNTIL -57 99  JUMP -57 99
 SW57 - VTR BREAK POINT, K3:  49
 SW57 - VTR BREAK POINT, K3:  87


          ------------   tran4  PARSING COMPLETE   -------------

     ----   BEFORE CLEANING UP THE OUTPUT   ----
 TOTAL NUMBER OF CLAUSES IN SENTENCE =   1
 TOTAL NUMBER OF PHRASES (PHCTO)      =   1
 TOTAL NUMBER OF OPADR2 ELEMENTS (OPO)=  45
CLSNFO ARRAYS - NUMBER OF CLAUSES IDENTIFIED   (INCLUDING MAIN CLAUSE) = 1
                NUMBER OF CLAUSES MOVED        (EXCLUDING MAIN CLAUSE) = 0
                NUMBER OF CLAUSES STILL TO BE MOVED                    =  0
              BEGIN  ENDING BEGIN  ENDING
       CLAUSE INPUT  INPUT  OUTPUT OUTPUT PARENT CLMRKR ANTCDN ANTCDN ANTCDN ANTCDN RELPRO
         ID   SWORK  SWORK  SWORK  SWORK  CLAUSE SCONS  SWORK  SCONPT OPIBEG OPIEND SCON
            1      1      5      1      1      0      0      0      0      0      0      0
       CLAUSE  PARENT
         ID    CELLS    ( TRAILING ZEROES ARE NOT PRINTED )

CURRENT CLAUSE ID =  1
CLSCON ARRAYS   (CLSID IS INITIALIZED TO 1. ENTRY NOT PRINTED IF CLSID=1 AND BOTH CMCHLD AND ACHILD = 0

PHRBEG: STARTING OPADRO POSITION OF EACH PHRASE
     1

PHREND: ENDING OPADRO POSITION OF EACH PHRASE
```

```
    45

OPADRO
  -108    -1  -102  -107  -107  -103  -105  -108  -106  -101     3  -104  -112  -110  -113  -118  -111  -115  -120  -
114
  -116     4  -109  -117  -110  -113  -118  -111  -115  -119  -114  -116  -120  -102  -107  -103  -105  -108  -106  -
101
     5  -104  -112  -110     6

SCONPO
    27     1    28     2    29    30    31    32    33    34     3    35    36    37    38    39    40    41    42
43
    44     4    45    46    47    58    59    60    61    62    63    64    65    48    49    50    51    52    53
54
     5    55    56    57     6

HFDOPO
     0     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0
     0     0     0     0     0

     --- T4CLRTRN:  ONLY 1 CLAUSE.  NO REPOSITIONING NEEDED  ---

 ***********     tran4  PROCESSING COMPLETE      **********

 THE SCON FOR tran4
              1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
       1     20    1    0    0    0    0    0    0    0    1    0    0    1    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0
              1    0    1    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
       2     14  101    1    1    0    3    1    0    0    2    0    1    1    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0   94    0
              1    0    1    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
       3      1   29    1    1    0    3    1    0    0    3   29    1    9    0    0    0    0    0    0    1
              0    0    0    0    0    0    0    0    0    0  101    0    1    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0   18    0   52    0
              1    0    1    1    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
       4      2  886    1    1    0    3    1    0    0    4   60    2   11    0    0    0    1    1    1    1
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0   13    0    1    0
```

	1	847	60	1	3	4	0	0	886	0	0	23	0	0	0	0	0	0	0	12
	0	0	0	0	0	1	60	0	3	0	0	0	0	0	0	0	0	0	0	0
5	1	609	9	3	0	3	5	2	0	5	40	1	6	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	16	0	83	0
	1	0	60	1	3	90	0	0	886	0	0	2	0	0	0	0	0	0	0	2
	0	0	0	0	0	7	21	0	1	0	0	0	0	0	0	0	0	0	0	0
6	20	10	0	0	0	0	0	0	0	6	0	0	10	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	60	1	3	90	0	0	886	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	108	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	102	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
29	107	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
30	103	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
31	105	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
32	108	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
33	106	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
34	101	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
35	104	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
36	112	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
37	110	0	1	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
38	113	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	118	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	111	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	115	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	120	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
      43    114    0    0    1    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
      44    116    0    0    1    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
      45    109    0    0    1    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
      46    117    0    0    1    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
      47    110    0    0    1    0    3    0    0    0    0    0    0    0    0    0    0    0    0    0
      48    102    0    9    3    0    3    5    2    0    0    0    0    0    0    0    0    0    0    0
      49    107    0    9    3    0    3    5    2    0    0    0    0    0    0    0    0    0    0    0
      50    103    0    9    3    0    3    5    2    0    0    0    0    0    0    0    0    0    0    0
      51    105    0    9    3    0    3    5    2    0    0    0    0    0    0    0    0    0    0    0
      52    108    0    9    3    0    3    5    2    0    0    0    0    0    0    0    0    0    0    0
      53    106    0    9    3    0    3    5    2    0    0    0    0    0    0    0    0    0    0    0
      54    101    0    9    3    0    3    5    2    0    0    0    0    0    0    0    0    0    0    0
      55    104    0    9    3    0    3    5    2    0    0    0    0    0    0    0    0    0    0    0
      56    112    0    9    3    0    3    5    2    0    0    0    0    0    0    0    0    0    0    0
      57    110    0    9    3    0    3    5    2    0    0    0    0    0    0    0    0    0    0    0

PHRBEG: STARTING OPADRO POSITION OF EACH PHRASE
     1

PHREND: ENDING OPADRO POSITION OF EACH PHRASE
    45

OPADRO
    -1     2     3     4     5     6

SCONPO
     1     2     3     4     5     6

HFDOPO
     0     0     0     2     0     0

*EOS*
*TRANSL OUT*

*Scon Information*
OPADR CASE DECL DEGR GEND NUMB PERS TENS  PAT   WC SC08  SSU CC  SMC
   -1    0    0    0    0    0    0    0    0   20    0    0 LOG
    2    1    1    1    1    1    3    1   94   14    0    0 LOG 001            der
    3    1    1    1    1    1    3    1   52    1    0    0 LOG 001            Himmel
    4    1    1    2    1    1    3    1    1    2    0    0 LOG 001            sein
    5    5    9    1    3    1    3    5  443    4    2    0 LOG 001            blau
    6    0    0    0    0    0    0    0    0   20    0    0 LOG 001            .


*From Dictionary*
 der Himmel sein blau.
```

```
*After Stemgen*
 der Himmel ist blau.

*After Black Hole*
 der Himmel ist blau.

*After Finish Rules*
 der Himmel ist blau.

*After deleteEmptyUnits*
 der Himmel ist blau.

*After Capitalization*
 Der Himmel ist blau.

*After adjustFinalSpaces*
Der Himmel ist blau.

*After Pattern Matcher*

Pattern matcher has no rules

*After Complete Generate*
Der Himmel ist blau.

*Output*
Der Himmel ist blau.

*EOS*
```

APPENDIX B
GLOSSARY OF TERMS

OpenLogos Glossary of Terms
(v. 3, October, 2010)

Term	Meaning
Alternate Word Class	Certain word classes have secondary target transfers, called its "alternate word class." In the dictionary, certain word classes (adjectives, adverbs, process nouns, verbs) have not one but two target transfers. This is to allow rules the ability, e.g., to transform certain adjectives into adverb, or process nouns into verbs, etc. ▪ Alternate word class of verbs is a process noun ▪ Alternative word class of a process noun is a verb. ▪ Alternative word class of a participial adjective is a verb ▪ Alternative word class of certain classes of adjectives are adverbs and vice versa. ▪ Thus, e.g., an English source construct like "specific indication", if desired, can be rendered in the target as "indicate specifically."
Cells	Communication areas used by RES and TRAN rules to record and/or test information about a constituent or sentence during analysis. • Cells 1-10 are local to any given rule and are initialized at the end of each rule processing. • Cells 11-98 are arbitrarily defined by each language pair (in TRANs only) • Cells are employed for both intra- and inter-

	TRAN purposes. • RES cells are not directly accessible to the TRANs, but the values in certain RES Cell are stored higher positions of the Scons associated with each SWORK in the TRANs.
CSA	Clause Status Array. This is the array that appears in the diagnostics at the end of RES2. It summarizes clausal information for both major and minor paths of the given sentence.
Constants	Constants are addresses to target words stored in a special target dictionary separate from the main dictionary. ▪ When a rule-writer wishes to introduce a target word not provided for by in the initial dictionary look-up, he or she may draw upon terms stored in separate target dictionaries ▪ Pointers to words in these target dictionaries are via numbers. o Constants numbered 121-998 are considered high-frequency constants. o Constants numbered 1000 to 9998 are low-frequency constants. ▪ When a constant is loaded into the OPADR (q.v.) it receives its own Scon (q.v.)
Constraint Line	Constrain lines are a component of a TRAN rule, appearing below the SP line, and specifying constraints that must be satisfied for the rule to fire. Thus, even though the SP line may have matched on a rule, the constraints associated with the rule must also be satisfied. Constrain line will test either cells or Scon positions for the presence or absence of some value or set of values. The constrain line may also contain tagsets (q.v.)
Multi-Target	OpenLogos is a multi-target system, meaning that one

	source analysis may feed into any number of target languages for which appropriate data bases are available.
Negative Word Classes	If an SWORK has a negative value in its WC field, this value stands for a set of word classes. E.g., WC -8 stands for WC 1 (nouns), WC 14 (definite articles), WC15 (indefinite articles), and WC 16 (arithmates). Negative word classes only appear in rules. I.e., a rule with an SWORK with -8 in its WC will match on any of the word classes cited above. (See Tagsets, below)
OPADR	OPADR is the target language 'output address array' that is built by the linguist during analysis of a source constituent. • The OPADR consists of pointers to target-word equivalents for the words of a source constituent. • In sum, OPADRs are target word equivalents of a source constituent, and get built at the point where analysis of a source constituent is completed. o Thus, OPADRs, one for each target parse tree constituent, constitute the target sentence in formation, prior to actual generation.
Overflow Fields	These terms refers to fields in the dictionary record of an early, initial (1970) version of the Logos Model, before the SAL representation language, when only a single type code was assigned for the semantics of a word. ▪ When SAL was introduced, the original type field was used to store the SAL set code, and the subset code (if any) was placed in a field called Overflow1, and the superset code was placed in Overflow4. Overflow 2 and 3 were used to record target word features, e.g., reflexives, etc. ▪ Though no longer meaningful in the OpenLogos

	relational database, these terms still appear in diagnostics and early documentation.
Relational Pointer	Relational pointers occur in the VTR line and serve to point to elements in the SP line. There are two kinds of relationaol pointers: (1) relational pointers in the parameters of switches (e.g., -81, -82, etc). tell the switch which element in the SP line the switch function pertains to. E.g., -81 signifies the first element (SWORK)in the SP line. In switches, the relational pointer always has the form -8n. Their range is from -81 to -89 (-90 is a possibility too I believe but SP lines rarely have 10 elements) (2) relational pointers in the VTR line (outside of a switch) have the form -1 0 , -2 0, etc.. They instruct the system to load the address pointer of the designated element in the SP line into the OPADR. The relational pointer her always has a parameter which is usually zero. If it has the value 2, it tells the system to load the address pointer of the so-called “alternate word class” instead. (See above.)
RES RES rules	RES1 and RES2 are major program modules executed immediately following dictionary look-up. Their function is to match the SAL SWORK input stream against associated RES rules (a large body of rules for each of the RES modules), and that, upon a match, to then execute the instructions provided by the action part of the rule. • The RES rules may be understood as accomplishing a macro-parse of the input sentence. • Marco parse is limited to: (1) part-of-speech homograph resolution and (2) clause identification.

	• RES1 resolves parts of speech where the context makes POS unambiguous, e.g., the term 'house' (noun or verb) is resolved to noun when preceded by a determiner. • RES2 makes final decisions about all ambiguous parts of speech and identifies the nature of each clause, including embedded clauses • Top-down parse information is supplied to the TRAN modules which then perform a bottom-up a micro parse of the sentence, using information from RES analysis as necessary.
RESPAS Cells	These are the cells that RES passes on to the TRANs They are displayed in diagnostics as MAJOR PATH and MINOR PATH, appearing at the end of RES2 processing. Cell values are: Cell 1 – CL/A: Clause type for principal clauses: 1 – MC; 2 = DC; 3=Undiffrerentiated clause Cell 2 – UV Unambiguous verb (not used) Cell 3 – VB/T: Clauses's verb Set Code Cell 4 – SJ/F: Form code for Clause's Noun Subject Cell 5 – VB/F: Form code of verb Cell 6 – VB/X: Expectations associated with main verb Cell 7 – CL/B: Secondary Clause characteristics Cell 8 – N/X: Expectations associated with N or ADJ Cell 9 – PR: When > 0, signifies presence of a PREP Cell 10 – DC/S: Secondary characteristics of DC or other special use Cells 11 -19 mirror their counterpart above for minor clauses (e.g., relative clauses, embedded parentheticals, etc.)
SAL	Semantico-syntactic abstraction language, the symbolic language by which natural language is represented internally in OpenLogos.

	• SAL is considered a second-order language (NL being a first-order language) • NL strings easily map to SAL strings • SAL is a taxonomy with supersets, sets and subsets (though not all sets have subsets) • SAL has about 1000 entities covering both closed and open classes. As such it occupies a mid-point between the sparseness of syntax and the richness of natural language. • Both the input stream and the pattern portion of rules are expressed as SAL patterns. This fact is what allows TRAN and RES rulebases to be accessed like a dictionary and accounts for the very large number of rules in these databases. • SAL seeks to characterize language at the point where syntax and semantic intersect, i.e. at the point where syntax and semantic influence each other, as, e.g., in the sentence "keep the book on war on the table," where the meaning (and parse) of 'on' in both instances is a function of other words in the sentence, and where the meaning of the verb 'keep' itself is a function of the second preposition. • SAL is expressed numerically internally to the system, e.g., the SAL code for 'book' is: o superset: 12 (information) o set: 76 (recorded data) o subset: none for this word.
Scon	Semantico-syntactic control record. ▪ Scons supplement and embelish the information associated with an SWORK. ▪ Each SWORK has associated with it a 100 word record into which is stored such details as

	○ gender, person, number, case ○ SAL superset, set, subset ○ other pertinent data regarding the element, or its context ▪ Information from RES about the nature of the clause, etc. is stored in higher positions of a Scon. ▪ Scon positions are accessed by switches in a VTR
Semtab	Semantic Table containing a very large number of semantically focused rules whose purpose is to resolve polysemous words to the meaning appropriate to a given context. • Semtab rules are nested rules invoked by TRAN rules at any point in source analysis. TRAN sends some portion of the input stream it is dealing with to Semtab for a nested match. • Semtab rules look and function very much like a TRAN rules, i.e., they consist of an SP line (semantico-syntactic line, or SAL Pattern, q.v.) .and an action portion which is executed when a match occurs. ○ Semtab rules however are considerably more restricted in what they can do. • Like the TRAN and RES rulebases, Semtab in effect is a highly indexed pattern dictionary, where the patterns are in the representation language known as SAL. (q.v.)
Slot	Slots appear in the Action line (VTR) of a rule. They are receptors into which a portion of the VTR can be loaded for delayed executed by a subsequent rule. Slots range in number from 70 to 98 and are loaded via a -11 switch (e.g., -11 073 -1 0 loads the relational pointer -1 0 into

	slot 73. Leading zeros are an old convention and have no meaning). In effect, Slots are mechanism for specifying a delayed action. A subsequent rule whose VTR contains a Slot will automatically execute any such instructions that may have been previously loaded in the Slot.
SP Line	This is the semantico-syntactic pattern of a TRAN rule that must match on a corresponding pattern in the input stream for the rule to become active. SP line can have up to ten elements, three of which can be Kleene Starts (Note: Before the rule can actually fire it must also satisfy any of its rule constraints.)
SUBCHG	Subset Change (actually Subset supplementation). The SAL Subset of an SWORK may be supplemented, during dictionary look-up or by RES, with an additional value designed to signal some property of the SWORK needed in subsequent processing. (This supplemental Subset does not actually change the original Subset, which can still be accessed.) RES1 will supplement the SAL Subset field of an SWORK to 862 to indicate that the element is a possible transitive verb (PVT), thus allowing a subsequent RES2 rule to resolve the SWORK accordingly. This sort of SUBCHG does not survive into the TRANs. Other SUBCHG values do survive into the TRANs, in a Scon, viz., those having to do orthography.E.g., an initially-capped unfound word will receive a SUBCHG value of 859. (NOTE: Unfound words are assumed to be nouns.)
Superform Values	A superform value is form field value that will match on the set of primitive form values associated with it. The SP line in rules will commonly have a superform value in the form field of one or more of its SWORKs to

	enable to rule to match on a relevant variety of primitive form field values in the SWORK input stream.
Switch(es)	Programmed functions occurring in the action portion (VTR) of a TRAN rule. • Switches are introduced by a negative number and have parameters the number of which depends upon the particular switch. • Switch numbers range from -11 to -68. (Some numbers are not used)
SWORK	Semantico-syntactic work record. Each word of an input sentence is converted into an SWORK at end of dictionary look-up. • A NL sentence thus becomes a string of SWORKs prior to analysis. • All operations within RES and TRAN are performed on SWORKs. o SWORKs in effect are entities (objects) in the same sense that NL words are entities or objects. o SWORKs are synonymous with SAL words. • An SWORK has three fields: o Word class (expressed as a number, e.g., wc for nouns = 1. for verbs = 2, etc. o Semantico-syntactic (SAL) Type (three numbers, one each for superset, set and subset). (See SAL in this glossary.) ▪ Only one of these fields as actually displayed (usually the subset) but all are available for matching purposes. ▪ The Scon record associated with each SWORK will have the full set of SAL values, among other

	information supplemental to the SWORK. o Morphology. E.g., a 1 in the form field of a noun signifies singular, 2 signifies plural ▪ For uninflected words, this field may be used to store other kinds of intelligence. ▪ For example, prepositional phrases that are seen to complement a verb (i.e., are converbal) will be assigned a form field of 53 by the TRANs.. • A forth value associated with SWORKS is the SWORK number, corresponding to the word number in the input sentence. o This 4th number allows the linguist to relate SWORKs to their origins. • In the course of the parse, SWORKs become concatenated and thus, e.g., the multiple SWORKS of a noun phrase will be represented by a single SWORK for the head noun. o Once concatenated, elements of the noun phrase such as determiners, adjectives, and modifying nouns are no longer displayed in the higher TRANs. o Thus, the SWORK string becomes more and more abstract as analysis proceeds across the TRANs, much in the fashion of a parse tree.
Tagsets	Tagsets are a set of SAL supersets, sets, and/or subsets that replace the single value in the Type field of an SWOPRK. Tagsets appear on the Constrain line of a TRAN rule. When composing a rule, if the linguist places a -02 in the Type field of an SWORK, he or she

	will insert a line immediately beneath the SP line containing a set of SAL codes. The rule generator will replace the -02 with a pointer to an address where the set of SAL codes are stored. This set of codes is a targset, and it enables rules to match on multiple SAL codes. (See Negative Word Classes, above).
TRAN(s)	Program modules that, along with the RES modules, perform analysis of the source system and create its parse. • There are four TRAN modules, numbered from 1 to 4, which are executed sequentially. • Associated with each TRAN module is a large body of TRAN rules (also called SP rules). • The output of TRAN1 serves as input to TRAN2, etc.
TRAN Rules	Rules associated with the TRAN modules, one body of such rules for each of the TRANs. • TRAN rulebases are databases organized like a dictionary, indexed on the SP line. • During analysis of a sentence, the input stream (sentence), itself now a string of SAL words or elements, is matched against the TRAN rulebases much in the way that words in a NL text are matched against a dictionary. • TRAN rules have four data components: o Comment line (for linguist'se only) ▪ The line typically summarizes in telegraphic style the rule action, shown after "=". ▪ E.g., = N would mean that the noun is (1) being added as NP to the output SWORK array which

	will serve as input to the next TRAN; and (2) that all the target words associated with this NP are being loaded into its associated OPADR (q.v.) ▪ E.g., = (adj) would mean that the adjective which the rule has matched on is being concatenated with an element yet to be processed, (which may or may not be the head noun). I.e., no SWORK is being formed and no target addresses are being loaded into the OPADR at this point.. ▪ -1, -2, etc., signifies the number of SWORK elements the rule is backspacing on, i.e., where the next match will begin. It's absence means no backspace. ▪ -A*1, a backspace indicator for rules that have stretch elements (Kleene stars) in their SP line. It means 'backspace all but one element.' ▪ -A*0 means 'backspace all the way.' o SP line, the semantico-syntactic pattern line (SAL Pattern) to which the rule pertains and by which the rule is indexed. o Constraint line, the conditions which must be satisfied. When the SP line of a rule is found to match some portion of the input stream, any and all constraints in the Constraint line must be satisfied otherwise

	the rule will not fire. ▪ Constraints can pertain to almost anything previously understood and recorded about the sentence by RES or some earlier TRAN rule. ▪ Constrain lines test inter- or intra-TRAN communication Cells or Scons, q.v.) o Action line, or VTR (q.v.), where the action takes place once a rule is fired. Actions pertain to source parse analysis and, optionally, to target work parse tree work. ▪ Optional target work is accomplished by links to 30-, 40- and 50-tables wherein target work on the SP line of this rule is specified.
TYP	A term used for the second field in the triple code (WC/TYP/FORM) for SWORKS. TYP may contain SAL codes for Set, Subset, or Superset. .
VC	Variable constant, so-called because the label is constant but the contents will vary. VC's serve as labeled receptors into which target words are inserted during execution of the action portion of a TRAN rule. • VC's were designed to allow the linguist to specifically address and alter a target word that has been previously dealt with when handling a source constituent, provided it was inserted into a VC. • VC labels are numbers ranging from 101 to 119. • VCs (along with target words not previously inserted into VC's) are loaded into the target

	OPADR at the point where source constituent analysis is completed and its target equivalencies are established.. o OPADR is the 'output address array' containing pointers to target-word equivalents to the words of a source constituent. In effect, OPADRs are target equivalent of a source constituent, and gets built at the point where analysis of a source constituent is completed. o Thus, OPADRs, one for each target parse tree constituent, constitute the target sentence in formation. • As an example of VC usage, determiners that have been concatenated with their head noun in TRAN1 sometimes need to be altered or removed at a higher TRAN level, because of some context. Because the transfer for this determiner was inserted into a specific VC before it was loaded into the OPADR, the linguist can now address that VC and alter its contents.
VTR	VTR means vector transform. It is the action part of a RES or TRAN rule. • The VTR deals with source actions (analysis and source tree parse), and *optionally* with target issues, e.g, making notations for target equivalents for a given source constituent. • Target actions within a VTR are accomplished in so-called 30-, 40-, and 50-Tables (q.v.). • Any number of targets may feed off a given source rule, making OpenLogos a so-called multi-target system. • Virtually all source and target actions are effected by switches (q.v.).

	• At the end of the VTR, the -41 switch indicates how many elements the system is to backspace on for the next match.
	  ○ -41 101 means 'backspace all the way but one' (used for rules whose SP line has stretch elements, i.e., where the actual number of elements in the rule is unspecifiable
	  ○ -41 100 means 'backspace all the way.' .

30-, 40-, 50-Tables	These tables store the target actions. They are linked to during execution of the VTR of a TRAN rule.
	  ▪ Like the VTR of a TRAN rule, the rules in these Tables consist of switches, some of which make condition tests and have branch points..
	  ▪ 30-tables are always uniquely (albeit optionally) linked to a single source rule. I.e., the VTR of that rule optionally links to a 30-table specific to that particular rule, in order to accomplish some desired target action on the SP pattern which the TRAN rule is dealing with. .
	  ▪ 40-tables have sharable functions and can be linked to by any VTR or 30-table, or by other 40-table.
	  ▪ 50-tables are a special inter-lingual form of 40-table. It's actions are designed to be language neutral.

- - end