**APPENDIX B**

**GLOSSARY OF TERMS**

OpenLogos Glossary of Terms

(v. 2, September, 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 pattern-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 pattern-rules to record and/or test information about a constituent or sentence during analysis.   * Cells 1-10 are local to any given pattern-rule and are initialized at the end of each pattern-rule processing. * Cells 11-98 are arbitrarily defined by each language pair * 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. |
| Constants | Constants are addresses to target words stored in a special target dictionary separate from the main dictionary.   * When a pattern-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.   + Constants numbered 121-998 are considered high-frequency constants.   + 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 pattern-rule, appearing below the SP line, and specifying constraints that must be satisfied for the pattern-rule to fire. Thus, even though the SP line may have matched on a pattern-rule, the constraints associated with the pattern-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 pattern-rules. I.e., a pattern-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.   + 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 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 pattern-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 pattern-rules (a large body of pattern-rules for each of the RES modules), and that, upon a match, to then execute the instructions provided by the action part of the pattern-rule.   * The RES pattern-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. |
| 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 pattern-rules are expressed as SAL patterns. This fact is what allows TRAN and RES pattern-rulebases to be accessed like a dictionary and accounts for the very large number of pattern-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:   + superset: 12 (information)   + set: 76 (recorded data)   + 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 pattern-rules whose purpose is to resolve polysemous words to the meaning appropriate to a given context.   * Semtab pattern-rules are nested pattern-rules invoked by TRAN pattern-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 pattern-rules look and function very much like a TRAN pattern-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 pattern-rules however are considerably more restricted in what they can do. * Like the TRAN and RES pattern-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 pattern-rule. They are receptors into which a portion of the VTR can be loaded for delayed executed by a subsequent pattern-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 pattern-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 pattern-rule that must match on a corresponding pattern in the input stream for the pattern-rule to become active. SP line can have up to ten elements, three of which can be Kleene Starts (Note: Before the pattern-rule can actually fire it must also satisfy any of its pattern-rule constraints.) |
| SUBCHG | Subset Change. SAL Subset in an SWORK may be temporarily or permanently changed, during dictionary look up or by RES, to signal some property of the SWORK needed for processing later on. For example, RES1 will change the SAL Subset field of an SWORK (its sometimes-called TYP field) to 862 to indicate that the element is a possible transitive verb (PVT), allowing a RES2 pattern-rule to effect that resolution. This sort of SUBCHG does not survive into the TRANs. Other SUBCHG values have to do orthography, e.g., an an initially-capped unfound word will receive a SUBCHG value of 859. In this case, the SUBCHG value would be permanent. 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 pattern-rules will commonly have a superform value in the form field of one or more of its SWORKs to enable to pattern-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 pattern-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.   + SWORKs in effect are entities (objects) in the same sense that NL words are entities or objects.   + SWORKs are synonymous with SAL words. * An SWORK has three fields:   + Word class (expressed as a number, e.g., wc for nouns = 1. for verbs = 2, etc.   + 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.   + 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.   + 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.   + Once concatenated, elements of the noun phrase such as determiners, adjectives, and modifying nouns are no longer displayed in the higher TRANs.   + 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 pattern-rule. When composing a pattern-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 pattern-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 pattern-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 pattern-rules (also called SP pattern-rules). * The output of TRAN1 serves as input to TRAN2, etc. |
| TRAN Pattern-rules | Pattern-rules associated with the TRAN modules, one body of such pattern-rules for each of the TRANs.   * TRAN pattern-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 pattern-rulebases much in the way that words in a NL text are matched against a dictionary. * TRAN pattern-rules have four data components:   + Comment line (for linguist’se only)     - The line typically summarizes in telegraphic style the pattern-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 pattern-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 pattern-rule is backspacing on, i.e., where the next match will begin. It’s absence means no backspace.     - -A\*1, a backspace indicator for pattern-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.’   + SP line, the semantico-syntactic pattern line (SAL Pattern) to which the pattern-rule pertains and by which the pattern-rule is indexed.   + Constraint line, the conditions which must be satisfied. When the SP line of a pattern-rule is found to match some portion of the input stream, any and all constraints in the Constraint line must be satisfied otherwise the pattern-rule will not fire.     - Constraints can pertain to almost anything previously understood and recorded about the sentence by RES or some earlier TRAN pattern-rule.     - Constrain lines test inter- or intra-TRAN communication Cells or Scons, q.v.)   + Action line, or VTR (q.v.), where the action takes place once a pattern-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 pattern-rule is specified. |
| 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 pattern-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..   * + 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.   + 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 pattern-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 pattern-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 pattern-rules whose SP line has stretch elements, i.e., where the actual number of elements in the pattern-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 pattern-rule.   * Like the VTR of a TRAN pattern-rule, the pattern-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 pattern-rule. I.e., the VTR of that pattern-rule optionally links to a 30-table specific to that particular pattern-rule, in order to accomplish some desired target action on the SP pattern which the TRAN pattern-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--**