

THE FULCRUM SYNTACTIC ANALYZER
FOR RUSSIAN

Paul L. Garvin

The Bunker-Ramo Corporation
Canoga Park, California

1. The Fulcrum syntactic analyzer is part of a bipartite system for accomplishing automatic translation from Russian into English.

A bipartite system is characterized by the fact that it consists of two major portions: a machine dictionary and a translation algorithm. It differs from a tripartite system which consists of three major portions: a machine dictionary, a parsing algorithm, and a separate table of grammatical and semantic rules. In the author's opinion, the alleged advantages of a tripartite algorithm turn out to be illusory, whenever in practice the use of a realistically extensive grammar, and not just a few basic rules, is attempted. The grammar table then becomes so complex that it can no longer be handled by a simple algorithm. Then either the algorithm has to be adapted to the table, which reduces its generality, or a secondary algorithm has to be written which will show the processor the way around the table. The algorithm of a bipartite system is essentially a linguistic pattern recognition algorithm which, instead of matching portions of sentences against rules stored in a table, directs searches at the different portions of the sentence in order to identify its grammatical pattern. Thus, the essential characteristic of the algorithm is the sequencing of searches and, in each search subroutine, only as much grammatical information is used as is appropriate to the particular search. The rules of the grammar are in fact applied by the algorithm in a definite order, and a given rule is not even called unless the previous searches have led to a point where its application becomes necessary. This means that the highly complex system of rules that makes up the real grammar of a language is distributed over a correspondingly complex algorithm which applies the rules in terms of the ordering that the structure of the language requires.

The Fulcrum approach is based on two fundamental principles: the concept of the fulcrum and the pass method.

2. The concept of the fulcrum implies the use of key elements within the sentence (fulcra) as starting points for the searches performed by the algorithm. This means that the algorithm, in searching through a sentence, does not simply progress from word to word, but in fact "skips" from fulcrum to fulcrum. It performs a little search sequence each time it has reached a fulcrum, and goes on to the next fulcrum when this particular search is completed.
3. The pass method means that not one but several passes are made at every sentence, each pass designed to identify a particular set of grammatical conditions pertinent to the recognition process. Consequently, each pass has its own set of fulcra and its own search sequences. The pass method reflects the orderly progression in which the determination of the structure of the sentence is made: first, the sentence components are identified individually, then the relations between components are established, and finally the structure of the sentence as a whole is established. To each of these intermediate parsing objectives there corresponds, roughly, a pass or series of passes in the algorithm. The correspondence is not exact, because there are many ambiguities and irregularities interfering with the recognition process, and the design of the Fulcrum algorithm reflects these added complexities.
4. The Fulcrum syntactic analyzer has been implemented on two levels: (a) an earlier, less sophisticated version called Fulcrum I, which has been fully coded and is now capable of producing experimental translations; (b) a recent more sophisticated version, called Fulcrum II, for which detailed plans have been drawn up and which, when implemented on a general-purpose computer, will be capable of producing translations for practical use. It is expected that the complete implementation of the Fulcrum II will take four to six years.
5. The major improvements in the Fulcrum II over Fulcrum I are the following:
 - a. The order of passes and individual search operations deviates further from the order of descriptive levels than was the case in Fulcrum I. This is due to the fact that the new search patterns are based primarily on the order in which the grammatical information becomes available to the program rather than the order in which a linguist would prefer to present his description.
 - b. The Fulcrum II uses an iterative principle. The same set of search operations is used repeatedly to establish the internal structure of a variety of syntactic units. Thus, inserted structures (such as parenthetical expressions) are treated by the same search operation as entire sentences; clauses of

different types, such as relative clauses or independent clauses, are likewise treated by essentially the same search sequence. A control cycle insures that the different search sequences are called in the right order so that these different units are identified and related to each other appropriately.

- c. The heuristic principle of trial and evaluation is applied throughout the Fulcrum II algorithm. In all parts in the program where decisions are not unequivocal, a capability will exist for labeling decisions as provisional trials, so that they can be revised later in the program by evaluation routines based on information available to the program subsequently.
6. The research principle of the Fulcrum approach is based on the interplay of theory and experimentation: by designing an experimental program based on an initial theoretical conception, this conception can be refined by computer experiments made possible by the availability of an experimental translation program at all times.
7. The theoretical basis for the Fulcrum approach is the Definitional Model of Language proposed by the author, the methodological foundations are those of the inductive method developed by the author.

OUTLINE OF FULCRUM-IP SYNTACTIC ANALYZER

- O. Dictionary lookup and stem affixing
- I. Pre-scan: paragraph numbering; determination of typographical sentence category
- II. Pre-processing
 - 1. Determine sentence boundaries by determining whether sentence is interrogative, or exclamatory. Otherwise leave unlabeled.
 - 2. Assign grammatical function to symbols, non-Cyrillic matter, abbreviations (coded in dictionary according to whether or not they can be sentence-final), etc. [flag when necessary].
 - 3. Combine grammatical information on hyphenated words found separately in the dictionary.
 - 4. Assign grammatical functions to missing words, initials and titles before last names (to form appositive nominal block with last name); include dictionary words ending in hyphen [flag when necessary].
 - 5. Preliminary homograph resolution (include чем and как, but not commas or ambiguous conjunctions or dashes, nor potential interrogative pronouns, interrogative modifiers and interrogative adverbs) [flag when necessary].
- III. Pre-syntax
 - 1. Set unambiguous clause boundaries using the following signals: subordinating conjunctions; relative and interrogative pronouns, modifiers, and adverbs; то after если and так как (check out comma), colons [flag when necessary]; if clause boundaries are set, label clauses as dependent or independent or relative according to established clause boundary and check whether both boundaries are ready to be set; otherwise leave unlabeled. If no clause boundaries are set, treat sentence as single unlabeled clause.
 - 2. Label чем after comparatives as potential introducer of independent structure [flag].

- IV. First processing cycle: identify inserted structures and parenthetic structures (including quotes), label them (from dictionary) as to function if possible (potential appositions, apposition introducers, complements, separate inserted sentences); this cycle includes a table lookup and, if the potential inserted structure is not found in the table, calls the main and final processing sub-routines.
- V. Second processing cycle: process dependent clauses (including relative clauses and indirect questions) in sequence, going right to left by calling the main processing subroutine.
- VI. Third processing cycle: process independent and unlabeled clauses in sequence, going right to left (or process entire sentence as undivided whole) by calling the main and final processing sub-routines.

MAIN PROCESSING SUBROUTINE, MPS
(detailed syntactic identification)

- MPS-A Preliminary sequence: coordinate adverbs (semantic check); attach же and и to adverb, modifier, or pronoun; assign и to nominals, as applicable; package adverbs modifying other adverbs with their heads, creating adverbial blocks.
- MPS-B Nominal blocking sequence
- First subsequence: From right to left perform each step (1-3) as a separate pass at the clause or sentence.
1. Assign adverbs to modifiers or nominals as appropriate (including assignment of BOT to demonstratives), label unattached adverbs.
 2. Coordinate modifiers (semantic check), package modifiers modifying other modifiers (including самый) with their heads [flag] .
 3. Assign modifier packages to nominals: label unassigned modifiers; label previously unlabeled unmodified nominals and personal, relative, and interrogative pronouns for further processing as nominal blocks.
- Second subsequence: From right to left perform the following steps (4-10) in sequence, for each individual unassigned adverb, unassigned modifier, or nominal block, in a single pass.
4. Resolve genitive ambiguity when resolvable by preceding nominal, but not otherwise; [flag] .
 5. For nominal blocks only; attach prepositional structures and coordinate prepositional structures and appropriate adverbs if attachable to same head (include как...так и in coordination); attach governed structures including governed clauses; attach genitives of subject and object, attach other genitive complements [set flag]; attach appositions; attach appropriate unassigned modifiers (including все, их; check commas) and governing modifier packages set flag when necessary (wipe out "missing mandatory head" label); coordinate governing modifier packages when applicable; attach complements and coordinate (include как...так и in coord.).
 6. Coordinate nominal blocks and unassigned modifiers (wipe out "unassigned" labels for modifiers included

in the coordination), using semantic test; include appropriate unassigned adverbs in coordination (remove "unassigned" label); include constructions "nominal † c † nominal" and КАК...ТАКИ in coordination [set flag].

7. Label nominal block or unassigned modifier as prepositional structure if thus found; label nominal block or unassigned modifier as equational structure (e. g., КАК) if thus found; assign unassigned adverb or НЕ to prepositional structure when applicable (wipe out "unassigned" label); assign apposition or appositional modifier to prepositional structure.
8. Assign relative clause (using existing clause boundaries as previously set) to antecedent nominal block; attach additional coordinate relative clauses if applicable.
9. Assign governed structures and complements (including complementary adverbs) to unassigned and/or governing modifiers, coordinate governed structures and complements when assignable to modifiers (including ТАК...КАК И in coordination); assign governing modifier package to head if head is found to the right, but not otherwise (wipe out "unassigned" label); indicate missing mandatory head.
10. Assign dependent structure to unassigned comparative modifier or adverb when the latter is followed by ЧЕМ ; assign dependent genitive structure to unassigned comparative modifier or adverb; assign prepositional structure to unassigned adverb; indicate missing mandatory head.

MPS-C

Predicate blocking sequence

As a general search instruction, do not cross commas, conjunctions, or dashes, except when coordination is called in a step.

First subsequence: From right to left, perform all steps (1-7) as a single pass at the clause or sentence.

1. Label infinitives and gerunds.
2. Assign adverbs to infinitive or gerund.
3. Assign НЕ, ЛИ and reflexive pronoun to infinitive or gerund; assign past passive predicative to

БЫТЬ and use government and complementation codes of past passive predicative in the subsequent steps; coordinate past passive predicatives when applicable and use government and complementation codes of last past passive predicative in further processing.

4. Assign objects and complements to infinitive or gerund block, coordinate governed and complementary prepositional blocks and adverbs when applicable (note whether infinitive or gerund can govern a modifier and in that case wipe out "missing mandatory head" or "unassigned" label from governed modifier); assign previously packaged ЧТО-clauses and indirect questions to infinitive or gerund.
5. Coordinate infinitive or gerund blocks; coordinate past passive predicative blocks assignable to forms of БЫТЬ when applicable.
6. Assign infinitive block to previously established nominal block when applicable.
7. Indicate missing mandatory clause members when applicable.

Second subsequence: From right to left, perform all steps (8-15) as a single pass.

8. Label predicatives; label ЕСТЬ as special type of predicative.
9. Assign adverbs to predicative.
10. Assign НЕ, ЛИ, БЫ and reflexive pronoun to predicative; connect finite form of БЫТЬ with predicative.
11. Assign objects and complements to predicative block [flag when ambiguous], coordinate governed and complementary prepositional blocks and adverbs when assignable to same predicative blocks (note whether predicative can govern a modifier and in that case wipe out "missing mandatory head" or "unassigned" label); cancel government code of comparative predicative when in ЧЕМ...ТЕМ clause; assign previously packaged ЧТО-clauses and indirect question clauses to predicative.
12. Assign infinitive block to predicative block to form extended predicative block.

13. Coordinate predicative blocks (semantic check).
14. Find and label subject, or genitive or dative of reference (semantic check), for predicative block (include infinitive block, or что-clause or indirect question as subject, when applicable; do not cross other predicatives in this search).
15. Label clause members and include them in potential clause but do not absorb clause boundary in potential clause; revise ad hoc clause packaging done at III; indicate missing mandatory clause members and check whether candidates for these are not included in previously completed coordination—if so, revise packaging.

MPS-D

Clause blocking sequence

As a general search instruction do not cross commas, conjunctions, or dashes, except when coordination is called in a step.

1. Find unattached infinitive blocks and check for dative of reference or бы, attach the latter to infinitive block, and label resultant package as predicative package and potential clause, as appropriate.
2. Check for missing mandatory head to modifier and attach modifier to head or relabel modifier as nominal, as appropriate; assign unattached adverbs, prepositional blocks and other potential complements (including special instrumentals) to either nominal to the right or entire clause, as applicable, and coordinate when applicable (including так...как и in coordination); include clause introducers in clause and indicate clause boundary where possible and not previously indicated.
3. Label previously unlabeled clauses as follows—decide whether что, чем and как introduce indirect questions; label что-clauses that are not indirect questions as such; label чтобы-clauses and special relative clauses (those separated by semicolons?) as potential dependent structures, but retain clause status for purposes of attaching dependent clauses to them later; label infinitive clauses introduced by если, чтобы, прежде чем, перед тем как, etc., as dependent clauses; label clauses containing conjunctions and predicatives as dependent clauses; label clauses containing dative of reference and infinitive as independent clauses; label clauses containing бы and infinitives as dependent clauses; label clauses containing interrogative pronouns, modifiers or adverbs as direct or indirect questions, as applicable; label remaining clauses as independent clauses

FINAL PROCESSING SUBROUTINE (FPS):
SENTENCE BLOCKING SEQUENCE

Perform each step separately from right to left through clause or sentence.

1. Resolve dash ambiguity (repeat predicative blocking sequence at MPS-C8 if dash is resolved as predicative).
2. Check for independent clause; interpret unattached commas, colons, and conjunctions as clause boundaries if surrounded by potential clauses (include flagged commas in search) or if likely, right boundary of relative clause—if not, check for other possible coordinations—and establish status of coordinated clauses as dependent or independent; resolve status of hitherto unassigned и and то (and similar particles).
4. Assign unattached dependent clauses to nearest whole clause, attach чем-clauses to тем-clauses.
5. Check functional potential of remaining unassigned structures and assign if possible.
6. Label inserted structures as potential nominal blocks, appositions, apposition introducers, potential complements, separate clauses or inserted sentences, as applicable.
7. Label unattached independent clause as whole declarative sentence.