Syntactic Analysis by Alternating Computation and Inspection.

By Gustav Leunbach
The Danish Institute for Educational Research.

Automated procedures for analyzing a given text of some language into a sequence of morphemes representing word classes and flexions, is important for machine translation, for automatic abstracting and indexing, and possibly for other technical purposes. It is a commonplace that an automatic procedure cannot resolve every possible utterance of the language, and it is one among several problems for programmers of such procedures whether to leave some sequences unresolved or to present them for inspection to a person acting on the basis of some linguistic knowledge (theoretical and/or practical).

It is also a question of theoretical interest to what extent a given automatic procedure will function and how much it will be improved by the addition of one or another set of rules. The purpose of this paper is not to study existing programs from this viewpoint, but rather to build up a program from nothing, investigating step by step the economy of various additions to it.

In addition to the word Inspection for which a definition has been attempted above, the title of the paper contains another key-word, Computation, which is taken to mean any manipulation of symbols by a fixed set of rules. In this context the symbols are linguistic entities such as phonemes, letters, morphemes, words or sentence clauses, as well as formal symbols, e.g. numerals, used by definition to represent these entities and relationships be-

tween them.

Computation may be performed either by means of an electronic computer, a datamat, or by clerical assistance. The advantages of the use of datamats can be in the main summed up into three areas:

1. Accuracy. Errors due to malfunction of the datamat will in almost all cases be easily distinguished from correct results. There will, however, often be a need for coding to a datamatic medium, and the errors associated with clerical computation may be expected to appear in this part of the work.

2. Speed. My personal experience - competing with too many others for access to a datamat with too frequent technical breakdowns - has taught me not to overemphasize this advantage.

3. Controllability of instruction. If a clerk performs computational work on a text of a language he knows, it is not possible to prevent his common sense from interfering with the program he is performing, often with benefit, but at times in ways that are harmful to the purpose of the investigation. If the language is completely unknown to him, and he knows that it is - incorrect knowledge is even more harmful than correct knowledge - he will tend to develop headaches and lose both accuracy and speed. The datamat performs exactly the instructions contained in the program - programming errors may be much harder to detect than errors due to mechanical malfunctioning, but this matter pertains to the field of program writing technique in general.

The text which I have used in my study is a written one - I lack the necessary facilities for handling oral language - a Western novel in Danish coded on paper tape for use in typographical machinery - of course in a tape format almost incompatible with those used by datamats, but with some inventiveness it has been transferred character by character into type of normal format for the datamat in question. - In the composition of the novel no literary merit was intended, but since stylistic analysis is not a part of my program this is not a serious defect.

The symbols of the code are readily divided into letters and non-letter symbol, the latter being normal symbols of written language (space, case shifts, comma, point etc.) and typographical symbols such as italizing and de-italizing codes.

In the first computational phase the text is sorted into words and separators between words; a word is defined as an unbroken sequence of letter symbols, with the exception that a lower case symbol is allowed if the word begins with a capital.

The words are sorted alphabetically and each word is given a number above 200; all non-letter symbols have numbers below 100 (with separate values of each in upper and lower case); a space directly between two words is suppressed, else the text is stored as a sequence of numerals in the two mentioned intervals. The text is broken up into units of a maximum size determined by the storage rules of the datamat, but as far as possible terminated by full stop, question mark or the like.

Thus, when a later computational phase implies typing out the context of a certain expression, the context will not be a fixed number of words before and after, but in most cases a linguistically relevant section of the text.

The occurrences of each word are counted, not with the purpose of investigating any of the current theories of word frequencies, but for two specific reasons: 1. The words which occur only once are listed for inspection with the purpose of finding as many misprints as possible for correction in a later computational program (the tapes available for this study are the input before proof-reading).

2. A list of most frequent words is printed. This will contain some words directly related to subject matter - the name of the novel's leading character is placed at ab. no. 15 in frequency order - but mostly it will be words whose meaning is largely defined by their place in the syntactic structure, and many of the instructions in a program for computational analysis will deal with the treatment of these words. Even when words of concrete denotation enter this list, their frequency (in this text) may make it good economy to add information about word class etc. for them.

A further computational program in the first phase compares all words in the alphabetic list with a set of flexions and indicates which words possibly may be derived from other words in the list. This information is added to the word list.

At this stage the first set of general computational rules for sentence construction is introduced, partially by inspection of the list of most frequent words. This list is likely to contain instances of the following word classes:

Personal pronouns in subject form (I, he, we etc.)
Personal pronouns in non-subject form (me, him, us etc.)
Pronouns with nominal function (who etc.)
Pronouns with nominal or adjectival function.
Prepositions whose usual function is to adverbialize the following nominal clause.
Auxiliary verbs which fulfill the function of the finite verb, but are usually followed by other verb forms.
Conjunctions.
Adverbs which are characterized by their position in relation to the finite verb.
Particles which form a complete sentence (yes).
(Note: Concrete examples are supplanted by their English counterparts whenever possible, which it often is, due to the structural similarity between Danish and English. Three important differences may be noted: The definite article is a separate word if an adjective is present, else it is appended to the noun as a flection. The present tense of a verb is always different from the infinitive and has no personal flexion. The past participle is different from the past tense.)

Homonyms may occur in the list with one meaning in one of the above classes and another meaning in a class of words of concrete denotation. Example:"så" (at about the 20th place by frequency order) may be an adverb

translated as "so" or "then" or the verb form "saw". The rules by which such words are treated must contain a warning; it may be reasonable to include word class information for the second meaning.

The numbers between 100 and 200 are available for the coding of the grammatical information mentioned here.

The text may contain sentences which are completely analyzed by the first set of rules:"He likes me." "He has given it to her." Such sentences establish certain words as finite verb forms (reserving the possibility that they may be homonyms) and others as participles. Other sentences will with great likelihood establish certain words as nouns or adjectives.

Semicolons are sentence separators with at least the same degree of certainty as full stop, only they do not cancel out a subsequent upper case shift as a signal of a proper name. Commas may be sentence separators, but in many instances they are alternatives to the conjunctions "and" and "or"; computational rules must thus to some extent treat these conjunctions and comma as equals ("and/or" may also be sentence separators).

Now examples of incompletely analyzed sentences will be typed out for inspection with computed tentative assignments to word classes, including computed hypotheses of homonymy. A count is made of the proportion of the text which has been analyzed. The inspector may judge it necessary to have typed out the sentences on which the assignment of certain words depend - guided by the frequency count: with very rare words it may be useless,

with very frequent words it may be necessary to restrict output.

The inspected sentences may reveal other less frequent words which ought to be assigned to the above-mentioned structured word classes. Or they may point to the necessity of assigning word classes to some words of concrete denotation (particularly homonyms), or to accept or reject computed analyses of words into root morpheme and flexion morpheme. Or supposed sentence separators may be revealed to be abbreviation points and re-coded accordingly (this may lead to general computational rules such as re-coding all instances of "Mr.").

After this, the whole text is again computed, and examples of incompletely analyzed sentences "on a higher level" are presented for inspection, etc. (If a "hard core" remains, this may contribute to the list of instances of unresolvable ambiguity for future treatises of structural linguistics.) The important point is that every inspection phase is strictly limited; else computation would be of no help.