Gunnel Källgren
Institutionen för lingvistik
Stockholms universitet
106 91 Stockholm

HP - A HEURISTIC FINITE STATE PARSER BASED ON MORPHOLOGY

In the project "Satsanalys utifrån morfologiska kriterier", financed by HSFR, I work with a computerized model for parsing of Swedish sentences. (A more detailed description of this in Swedish is Källgren 1983.) It is connected with other work carried out by Benny Brodda, Stockholm, so we use the general label Heuristic Parsing (HP) for our joint efforts.

To be able to explain the ability of human beings to perceive and interpret language as quickly and as correctly as is actually done, we have to make certain assumptions. We must e g assume that humans have a highly developed skill for parallel processing, not only in the sense that one can take part in a conversation while driving a car and rubbing one´s nose at the same time. More important in this connection is that language perception and language understanding can be fruitfully regarded as a system of parallel processes. Another important assumption is that most of these processes work in a heuristic manner, in the sense that they strive for optimal probability rather than absolute correctness, that in choice situations they take a fair guess or a short-cut rather than a toilsome testing of every alternative.

In our HP-project, we try to draw some consequences of these assumptions. For a start, we have picked out what we believe to be one of the many processes involved and furthermore we treat it as consisting in its turn of several parallel subprocesses. We also strive to make it work "heuristically" in the way sketched above, i e more by trial and error than by careful stepwise analysis. In this way we will try to build and study a model that will perform a few of the many tasks a human mind can manage.

The major process we have chosen has to do with perception and interpretation of surface morphological features. This, we think, plays an important role in word class assessment and deciding of constituent structure. To be able to really interpret the full text, enormous amounts of semantic as well as pragmatic information and general knowledge of the world is necessary. This, we do not even try to do. Instead we want to stick strictly to a purely morphological surface level in order to investigate how much information about language structure can be gathered from that level. It may not be possible to say precisely HOW much, but we have already found that it is surprisingly much, so much that it has to be payed more heed to it also in more lexically based models.

To give a technical overview of the system, it can be described as a series of Finite State Pattern Matching machines (FS-PM), where the superordinate system governing the subprocesses can also be regarded as a Finite State machine. This will make the system equivalent to an Augmented Transitory Network.

HP - A Heuristic Finite State Parser Based on Morphology
Gunnel Källgren
Proceedings of NODALIDA 1983, pages 155-161

Each of the subordinate FS-PM machines corresponds to a separate computer program. For computational reasons the programs are run sequentially on texts, but are meant to mimic parallel processing. Each program delivers its result as an updated version of the original text. The analysis is entered directly into the text file in the form of a marked bracket notation (but without the brackets to save space, cf below).

Normally the programs are used for ordinary running text in computer readable form. Before and after the analysis programs proper, programs for standardizing and cleaning up input and output are run, but apart from that, no pre-processing is needed. A full lexicon is not needed either. Literally any Swedish text can be taken and processed in a short time and to a low cost, but the results will improve somewhat if some rules are designed to handle the peculiarities of a specific text type, e g rules for legal texts as opposed to fictional prose.

The analyzing programs are all based on pattern matching procedures on different levels. Strings and configurations in the text file are matched against the rules of the programs that scan the text in the manner of a Turing machine. The first three analysis programs have single words and parts of words as their range. The next three build up different kinds of constituents and the last one takes in larger sentence patterns.

The description to follow covers the state of the system in the autumn of 1983. It is important to remember that the system is still only in the first part of its development. There is much left to be improved and refined, but even so, the present results are quite interesting.

The total number of lexicon rules in the system is at present about 300. Most of them are in the first program, MK (Mark). MK marks the words for word class according to the code given in Figure 1, which is used throughout the analysis. The code letter is put before and after the identified word, thus making up bracket and mark in one.

(Fig. 1)

a = adverb(ial)	n = noun (phrase)
b = particle	o = possessive
c = clause	p = prepositional phrase
d = determiner	q = quantifier
e = preposition	r = pronoun
f = copula	s =
g = auxiliary	t =
h = form of 'have'	u = supine verb
i = infinite verb	v = finite verb
j = adjective	w = finite verb or noun
k = conjunction	x = finite verb or adjective
l = proper name	y = noun or adjective
m = infinite marker	z = adverb or adjective

MK contains function words from closed word classes, e g prepositions, pronouns and conjunctions. The number of words in the lexicon will probably have to be increased with several more adverbs, as they are often difficult to identify on purely morphological and distributional criteria. Still, the lexicon will never grow very big, it will in general be common to all text types and it will mostly contain words with a grammatical function rather than real content words.

Some typical results from MK are shown in (2).

(2) ePÅe ´on´, kOCHk ´and´, fÄRf ´is´

MK does not perform a real analysis, just an identification of entire word forms. The next program, SM (Swedish Morphology), carries out an advanced morphological analysis of every word. Identified prefixes and derivational and flexional suffixes are segmented and marked, as are some segmentable final letters. If information about word class can be extracted from the internal structure of a word, "bracket" marking according to the code in Figure 1 is also entered.

Words ending in e g NING or SION/TION (plus possible flexional endings) are always nouns in Swedish and get marked as such by SM, while words ending in LIG/LIG´T/LIG´A are likely to be adjectives and marked as such, etc.

Some heuristic devices come to play in this program. Some word forms can e g be either nouns or adjectives, like VAKEN, which is either the definite form of the noun VAK "hole in the ice", or the n-gender singular form of the adjective "awake". In cases like this, the definite decision is simply postponed until more information is available. The ending EN is segmented and the word is marked as ambiguous between noun and adjective. The same method is used for several other word class ambiguities.

Some output from SM is shown in (3). It can be a word that is segmented and marked, either as unambiguous or as ambiguous, or segmented and not marked, where the morphological information is not sufficient for identification at this stage. Words that have no recognizable inner analysis are left unaffected by the program.

(3) nFÖR>KORT<NING=ENn ´the abbreviation´
yVAK=ENy ´awake´/´hole in ice´
FÖR>KLAR´A ´explain´

The next program, PF (prefix), builds on the analysis done by SM. PF contains a lexicon with irregular verb forms that cannot be identified as verbs on morphological basis. They are listed here rather than in MK as they often appear as parts of conjoined words and MK only can work on unanalyzed words. PF will recognize words from its lexikon both as single words and as word stems, marking as verbs both SÅG "saw" and FÖR>SÅG "provided". Furthermore, the program has some heuristic default values. Words with a prefix and ending in vowel + R are also supposed to be verbs, this time in the present tense, as FÖR>SER "provides". Words with a prefix and ending in a vowel are too ambiguous to be marked yet, like FÖR>KLAR´A infinite

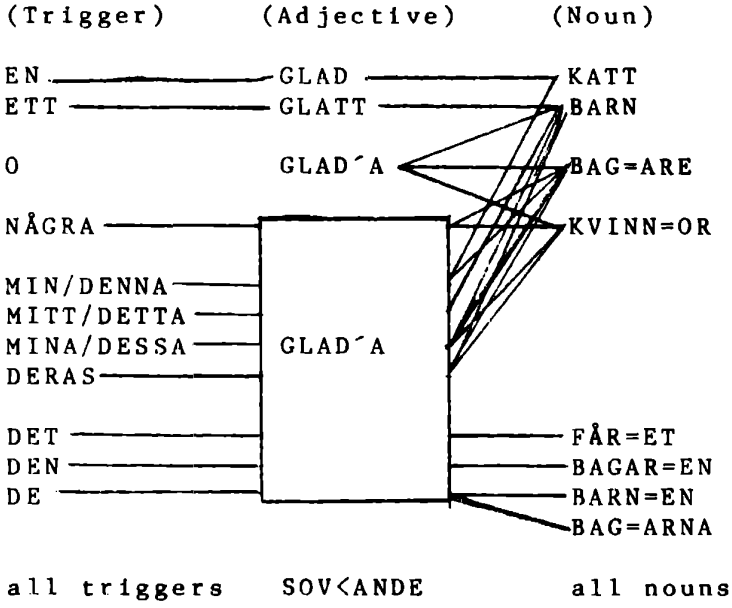
verb "explain", FÖR>MÅG´A indefinite noun "ability", FÖR>SYNT´A adjective "timid". Words with a recognizable adjective ending were identified already in SM, FÖR>KLAR<LIG´A "explicable", as were preterite verbs, FÖR>KLAR=ADE "explained". All other words with a prefix are simply assumed to be nouns. This holds to a remarkably high degree but is a point where we have to swallow some errors in the hope that they will be amended further on.

PF also contains some other types of possible word stems that are treated in a similar way. Some outputs from PF:

(4) vFÖR>SÅGv ´provided´
vFÖR>SERv ´provides´
nFÖR>SLAGn ´suggestion´

This is as far as one can get on the level of single words. Next come programs for building up constituents; noun phrases, prepositional phrases, and infinite phrases in turn. The monstrous diagram in (5) is an attempt at drawing a flow chart for the combinability patterns in Swedish noun phrases. The diagram is not exhaustive but it covers a fair amount of the noun phrases encountered in running text and gives a picture of the complexities involved.

(Fig. 5)



The implementation of diagram (5) in the program NP provides a very clear illustration of the finite state character of the programs. Every type of trigger is given its unique internal state and the different forms of modifiers and nouns have conditions on what internal states they accept. A condition can contain one single state, as that associated with nouns ending in =ORNA, which can only be preceded by the plural definite article DE, possibly followed by adjectives ending in ´A or NDE. A condition can also contain a whole set of states, like that for words ending in =ARE that have a very high combinability. Even so, the intervening adjectives must always be checked EN GLAD BAG=ARE "a happy baker" is OK, as is NÅGRA

GLAD´A BAG=ARE "some happy bakers", but neither *EN GLAD´A BAG=ARE nor *NÅGRA GLAD BAG=ARE, which are both excluded by the rules.

The NP-rules also dissolves several of the ambiguities noted in earlier programs. To return to the word VAK=EN that SM marked as ambiguous, it might appear in constructions like (6) and (7).

(6) qENq yVAK=ENy FLICK´A ´an alert girl´

(7) dDENd MÖRK´A yVAK=ENy ´the dark hole in the ice´

After the indefinite trigger EN in (6) the ending =EN is allowed to appear on an adjective but not on a noun while the unmarked word ending in ´A is a possible noun but excluded as an adjective. In (7) however, ´A is one of the two possible adjective endings after the definite article and =EN as a noun ending is congruent with both DEN and ´A. The noun phrases will then be:

(6´) nEN+VAK=EN+FLICK´An

(7´) nDEN+MÖRK´A+VAKENn

Had the article instead been DET, =EN would have been excluded as either noun ending or adjective ending, and the noun phrase would have terminated after MÖRK´A, treating the adjective as a nominalization, which is also quite possible and not uncommon in Swedish.

The program PP builds up prepositional phrases by connecting a preposition (marked ´e´ by MK) to a following noun phrase. It also identifies nouns by connecting prepositions and words that were formerly unmarked or marked as ambiguous between noun and something else. PP also builds up conjoined noun phrases within the range of a preposition. Some results are shown in (8).

(8) pMED+GLÄDJEp ´with joy´

pTILL+FLICK=AN+,+POJK=EN+OCH+DERAS+MORp
´to the girl, the boy and their mother´

IF (for InFinite) identifies infinite constructions. Those are often discontinuous. When they are triggered by the infinite marker ATT (which can also be a subordinating conjunction) adverbials are allowed to intervene between the trigger and the infinite verb. When the trigger is an auxiliary verb, adverbials and at most one noun phrase or one pronoun are allowed to intervene.

The IF-rules are alerted by one of the above triggers and then scans the string, only allowing constituents of the types mentioned to pass, until either a disallowed constituent breaks the scan or a possible infinite is encountered. Possible infinites are unmarked words with one of three possible surface forms: ending in a segmentable A, HOPP´A "jump", FÖR>LOR´A "loose"; monosyllables ending in any vowel, STÅ "stand"; or words with a segmented prefix and a monosyllabic stem as above,

FÖR>STÅ "understand". This is again based on probabilities, but the pattern "possible trigger (+ allowed constituent) + word of the described form" is strong enough to pick out almost every infinite verb with very few overgenerations. Chains of auxiliary verbs can also be managed. Illustrations:

(9) nDETn gKUNDEg rHANr aINTEa iFÖR>KLAR´Ai
´that, he could not explain´

mATTm aTROLIGENa aINTEa aBARAa iGÅi
´to probably not just go´

The last of the analyzing programs, DA (DisAmbiguation), works on the level of sentence syntax. The development of this program has only just started. Much more can be done on this level, e g identification of clausal structure, but at present it is mainly used for disambiguating words that are marked as ambiguous.

A word like FÅNG=AR will have been marked with ´w´by SM, as it can be either a verb in present tense, "catches", or a plural noun, "prisoners". If nothing else has changed the marking during the processing, it is likely that DA will.

(10) wFÅNG=ARw vFLYDDEv . ´prisoners escaped´
(11) aDÄRIFRÅNa vFLYDDEv wFÅNG=ARw . ´from there escaped prisoner
(12) nFLICK=ANn wFÅNG=ARw nHUND=ENn . ´the girl catches the dog´

In (10) and (11) ´w´ appears before or after ´v´, e g a finite verb. Two finite verbs are not allowed in the same clause and there is nothing to signal clause boundary, so the w-word must be a noun. In (12) ´w´ both precedes and follows a noun. Either condition would be sufficient to turn ´w´ into ´v´ for finite verb, as noun phrases are very rarely given on a row with nothing between them. DA contains several similar mechanisms for changing earlier marks.

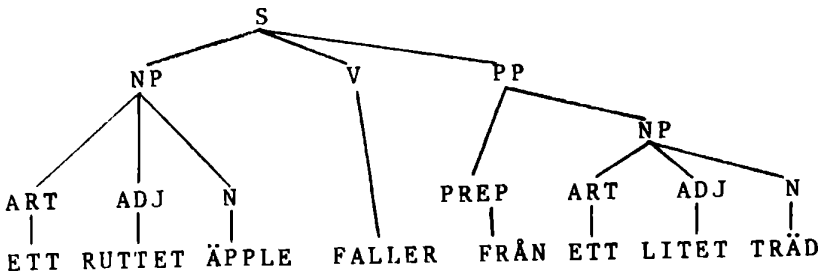
(10´) nFÅNG=ARn vFLYDDEv .
(11´) aDÄRIFRÅNa vFLYDDEv nFÅNG=ARn .
(12´) nFLICK=ANn vFÅNG=ARv nHUND=ENn .

This is the end of the analysis part of the system. Some analyzed sentences are shown in (13). When buildin constituents I take away their inner analysis and just put in +-signs. This is a deliberate choice. If the information was kept throughout the processing the output would look even messier than in (13), but it would contain literally all information, except the dominating S but including labels, that is needed to construct the tree structures in (14). (The underlined words are the only ones that have appeared in a lexicon.)

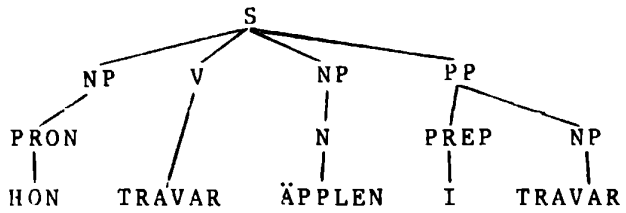
(13) a, nETT+RUTT=ET+ÄPPLEn vFALL=ERv pFRÅN+ETT+LIT=ET+TRÄDp .
´a rotten apple falls from a small tree´

b, nHONn vTRAV=ARv nÄPPL=ENn pI+TRAV=ARp .
´she piles apples in piles´

(14) a,



b,



All this shows clearly that this heuristic system can be used as a parser. It can parse the sentences of any Swedish text quickly and cheaply and with a result that is quite good, even when it is not perfect. This is fair enough, I am surprised myself that it works so well with so simple means, but what makes it interesting is not the degree of success or failure, but the fact that it is a truly linguistic system. The results are not reached by ad hoc solutions and smart programming in general, but by implementing linguistic patterns and probabilities based on linguistic intuitions.

We must ask ourselves why this model works so well. How can the information gained from morphology and a closed set of function words be sufficient to build up the whole structure of a sentence? I want to claim that morphology has an impact on language perception that has been grossly underestimated. Language is an economical system, we would not carry around so much morphology in language if we did not use it for something important. It seems likely that a "morphological interpreter" plays an important part as one of the several parallel processes at work in natural language understanding. Its results are of course always checked against the results from different semantic and pragmatic interpreters. In case of discrepancies, the semantics will probably always win, but where the results are in accordance, which they mostly are, the morphological interpreter may well be the process that gives sentences their grammatical structure.

If we regard the HP-system as a first attempt at modelling this morphological interpreter it means that the system also has a psycholinguistic relevance that deserves to be seriously considered and investigated.

Källgren, Gunnel (1983): Substantivjakten.
To be published as IRI-PM, Institutet för Rättsinformatik, Stockholms universitet.