MERGING - THE ART OF REPRESENTING DIFFERENT LEVELS OF SENTENCE STRUCTURE IN A SINGLE ANALYSIS TREE.

Frank Van Eynde

Departement Linguistiek, Blijde Inkomststraat, 21
B - 3000 Leuven, Belgium

1. An unproblematic example

The first example illustrates the merging of a part of speech analysis with a functional analysis.

1.1. Part of speech analysis and the on line principle

A part of speech analysis consists in the assignment of category labels to lexical units. By adding parsing rules to the categorial analysis one could also carry out the next step of grouping those lexical units into larger syntagmatic units.

The principle governing the part of speech analysis is the on line principle: syntagmatic units are analysed from the left to the right without changing the word order.

Example: (1) Harry promised me a new car

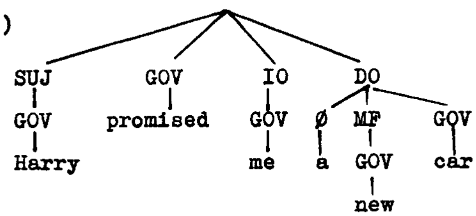
(1') $[_S [_N \text{Harry}] [_V \text{promised}] [_{PRO} \text{me}] [_{ART} \text{a}] [_A \text{new}] [_N \text{car}]]$

1.2. Functional analysis and the dependency principle

A functional analysis consists in the assignment of function labels to syntagmatic units.

The principle governing the functional analysis is the dependency principle: every syntagmatic unit ($\leqslant$ sentence) contains one and only one lexical unit functioning as its head.

Example: (1")



The sister nodes of each GOV-node are

either terminal nodes, in which case there is no function label specified. cf. the node of the indefinite article.

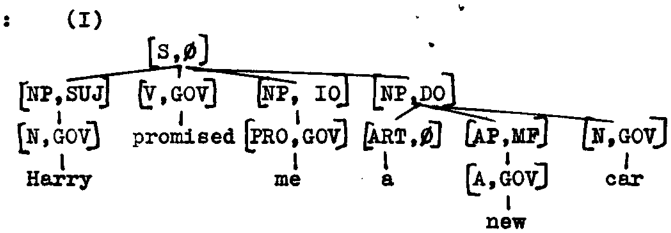
or non-terminal nodes, in which case the function label takes one of the following values: SUJ, DO, IO, MF (= modifier), ...

The function labels specify the kind of relation holding between the head and its dependents. For instance, "Harry" is the subject of "promised", "new" is a modifier of "car", etc.

1.3. A synthetic representation

Assumption: the part of speech analysis and the functional analysis can be represented in the same tree, since the on line principle and the dependency principle are compatible.

Example: (I)



2. A problematic example: the result clause

I´ll be concerned with the external structure of the result clause only, not with its internal structure; the result clause will, consequently, be treated as an unanalysed syntagmatic unit.

2.1. A part of speech analysis of the result clause

Example: (2) He left so early on Tuesday that we missed him

(2´) $[[_{PRO}\text{ he}]\ [_{V}\text{ left}]\ [_{Adv}\text{ so}]\ [_{Adv}\text{ early}]\ [_{P}\text{ on}]$

$[_{N}\text{ Tuesday}]\ [_{S}\text{ that we missed him}]]$

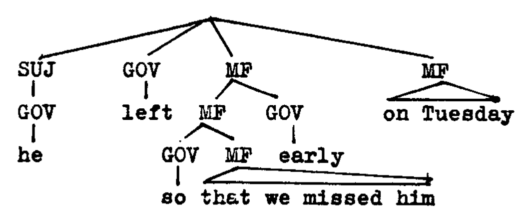
We could add some further structure to this bracketing by subsuming the adjacent adverbs under one node:
$[[_{Adv}\text{ so}]\ [_{Adv}\text{ early}]]$
We can, however, not incorporate the "that"-clause into this syntagmatic unit, since the prepositional phrase "on Tuesday" intervenes between both parts.

2.2. A functional analysis of the result clause

The crucial question concerning the functional analysis of a sentence like (2) is: which constituent governs the "that"-clause? In other words, where do we have to attach the S-node of the result clause?

For reasons (to be given in the full version of this paper) we propose the following dependence structure for (2):

(2")

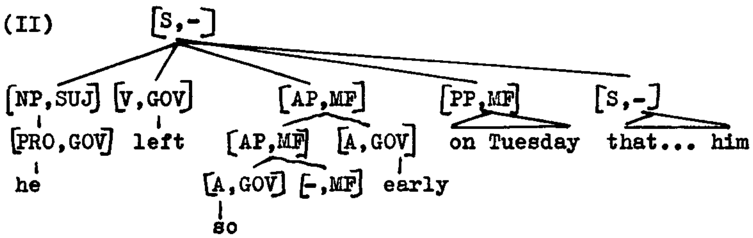


2.3. A synthetic representation

2.3.1. Unfortunately, the part of speech analysis and the functional analysis do not match (cf. the disturbance of the word order in (2")).

A marging of the labeled bracketing with the dependency structure would give the following result:

(II)



The lowest MF-node dominates no lexical material, and has, consequently, no category label. The righmost S-node, on the other hand, has no functional label, since it does not bear any dependency relation to the verb.

Important to note is the fact that both deficient nodes are complementary, and that they in fact refer to each other. In order to make this relation explicit, I propose to add an index (an arbitrary integer) to both nodes:

(II) $\ldots [-,MF]_1 \quad \ldots [S,-]_1$

Thanks to this coindexing device we are able to merge two levels of sentence structure, although they do not seem to be compatible at first sight.

2.3.2. The computation of (II) can be performed in a straightforward way:

1. $[_{S}, \ldots \text{so} \ldots] \rightarrow [_{S}, \ldots [_{AP,MF} [_{A,GOV} \text{so}] [_{-MF} \emptyset]_i] \ldots]$
2. If there is a that-clause in S', then give it a $[S,-]_i$-node, and attach it immediately under the S'-node.
   If there is no such clause, then delete the $[-,MF]_i$-node.

Comparison with a transformational treatment.

2.3.3. Similar analyses can be given for all kinds of comparison clauses.

3. A third example: deep and surface subjects

3.1. On the notion "surface subject"

3.2. On the notion "deep subject"

3.3. A synthetic representation

In a sentence like

(3) it seems that Steve likes her

it could be argued that the surface subject and the deep subject do not coincide. Making use of some new notations (to be explained in the full version of this paper) and of the device already known from section 2.3.1. I propose the following analysis tree for (3):

```
(III)          [S,-,-]

      [GN,SUJ-]1  [V,GOV,PRED]  [S,-,ARG1]1
          |             |
      [PRO,GOV]       seems       that Steve likes her
          |
          it
```

4. Summary

Merging is a technique of representing different levels of syntactic (and semantic) structure in one analysis tree.

In order to make merging work one has to make sure that:

1. each level of analysis is properly defined, i.e.
   - that there exists a list of possible values for the labels
   - that there is an algorithm for assigning those values in each particular case
   - that there is a unifying principle at each level (for instance, the dependency principle at the level of functional analysis).
2. The relations between the different levels are properly defined. In order to warrant the latter I have pleaded for

adding reference indices to the nodes, thus introducing a new formal device in the grammar

a third dimension in the analysis tree.

Some merits of the merging technique in a program for automatic translation.