

Conceptual analysis of several grammatical theories

Eugeniu Costetchi

October 25, 2015

1 Introduction

This paper sets to analyse the concepts used in several popular grammatical theories. The goal is to synthesize the all the concepts into a Generalized Annotation Model suitable for representing annotations and parses.

The models are OWL-DL compatible and expressed in UML graphical notation.

2 Stanford dependency structure

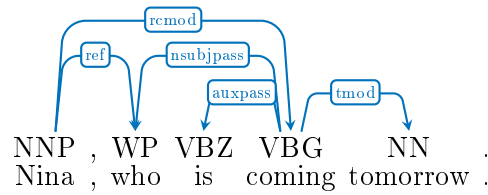


Figure 1: Stanford dependency example

2.1 Description of the concepts in use

The parse structure is a graph. Nodes correspond to words and are connected by labelled edges. Some nodes that correspond to punctuation marks, prepositions or conjunctions are orphan and are not included in the connected graph of the sentence, even if they are instantiated. Edges are dependencies distilled in a fine grained typology. Each node beside the word literal carries: part of speech (POS), lemma, position in the sentence, and eventually the named entity (NE).

2.2 Modelling Stanford dependency structure

A minimal model of SD is represented in Figure 2.2 where nodes carry all the features as fixed set of attributes and have a dependency relation to other nodes.

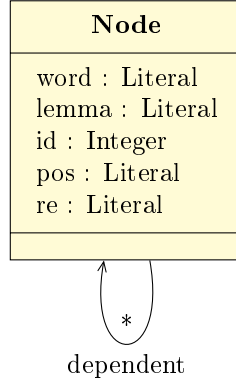


Figure 2: Minial model for Stanford dependency structure

As an alternative, that would allow more flexibility in adding a unrestricted number of attributes on nodes would be to treat them as feature structures. A feature structure is a set of attribute value pairs where the attributes are strings representing a name and the values can be a data value (String, number, date, etc.) or another feature structure.

Thus I propose to model the nodes as feature structures that contain several slots for each type of information: word, lemma, POS etc. However in order avoid mixing the relation types, it is still important to keep the node concept separated from the feature structure concept and treat the node as having an associated feature structure.

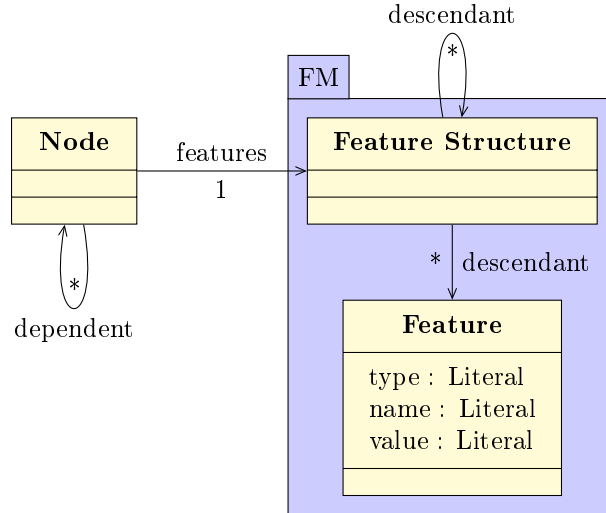


Figure 3: Model of the Stanford dependency theory with feature structures

The edges or relations in the model are generalised under the *dependent* relation. But in SG there is a whole taxonomy of relations. This is easy to

model in OWL language but it is not easy to represent in UML diagram. When the formal model is created the *dependent* relation shall be the root relation of the SD taxonomy, and the taxonomy itself shall be created in a separated ontology extending the current one.

Actually, Stanford dependency becomes increasingly popular framework, and perhaps it shall be a part of the core model. The only problem is that there are other sets of dependency relations, for example those used by Malt parser, which are different from Stanford relations and they are school specific.

3 Word Grammar structure

Hudson's Word Grammar is a dependency grammar. It uses with a different set of word classes than Stanford grammar which relies on Penn POS set; it introduces the inflection features; and most importantly it distinguishes the order of the words by distinction *post-dependent* and *pre-dependent*.

In Stanford Dependency the node order is indicated and controlled via the *id* attribute which represents the order number of the node in the sentence. However in Word grammar, there is a specific mention of the order relation. This fundamental relation of node order is transitive and is modelled as in Figure 3

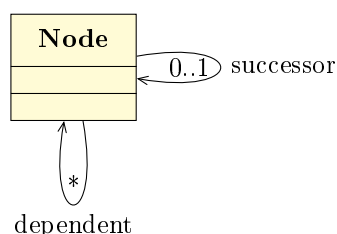


Figure 4: Minial model of dependency node with a successor relation

One may argue that this is yet another typology of dependency relation and we shall not complicate the model. But if needed the *successor* relation can be introduced and composed with *dependent* relation in order to enforce the above distinction. The composition rules will then look like in 1 and 2.

- (1) $pre-dependent(x, y) = dependent(x, y) \circ successor(y, x)$
- (2) $post-dependent(x, y) = dependent(x, y) \circ successor(x, y)$

4 Phrase structure

In linguistics, phrase structure grammars are all those grammars that are based on the constituency relation, as opposed to the dependency relation

associated with dependency grammars; hence phrase structure grammars are also known as constituency grammars.

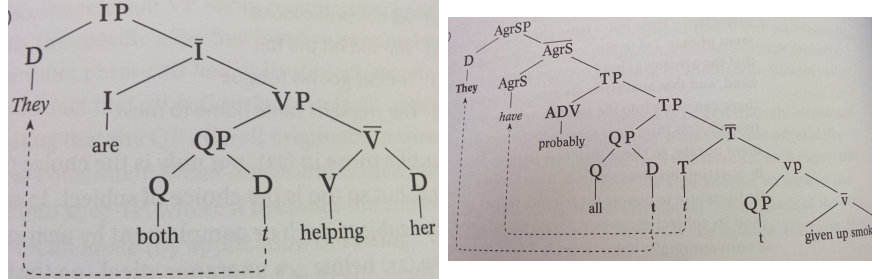


Figure 5: Phrase structure examples with traces

4.1 Description of the concepts in use

The phrase structure is a constituency structure represented as a tree. The tree nodes are either composite constituents or leaf nodes corresponding to words. The nodes usually represent the class of the constituent but may carry additional features, mainly agreement features, such as gender, number and case. Then the leaf nodes may correspond to a word, and empty constituent. Then there are trace elements, marked with t or t , that result from either head (verb & noun) movement or operator movement (aka wh-movement).

The edges of the tree are of only *constituent* edges, where the parent node is composed of child nodes as parts. However there are also secondary orthogonal relations for marking (a) movement and (b) agreement phenomena. The orthogonal edges can occur only between leaf nodes.

In the case of movement the relation stands between the trace and the moved element (or a transition trace). Also depending on the type of movement, the traces and the moved element can (a) share all the features, (b) have distinct sets of features or (c) have a shared set and another non-shared set of features.

In the case of agreement the relation seems to play a similar role. This needs a further investigation because the meaning of agreement relation is different from the one representing movement.

4.2 Modelling phrase structure

A model of phrase structure is presented in Lemon ontology ?? depicted in Figure 6.

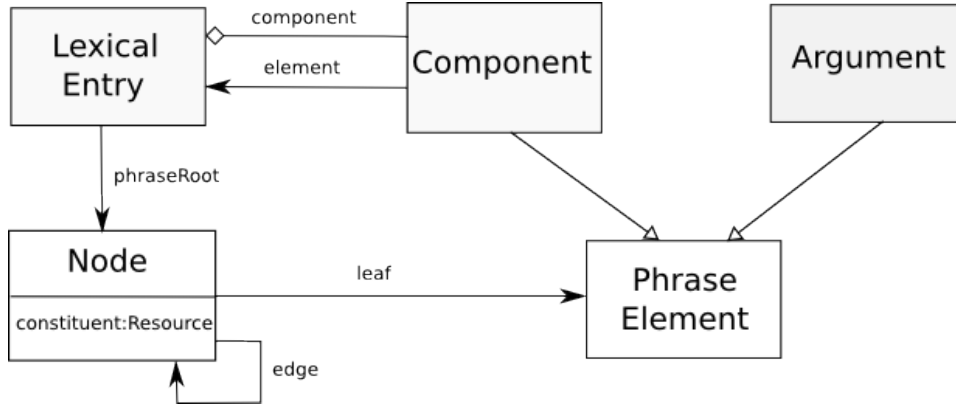


Figure 6: Phrase structure model in Lemon ontology

Lemon model originated as a need to model lexical entries. Below in Figure 7, I propose a model of phrase structure coming from the parsing and annotation direction.

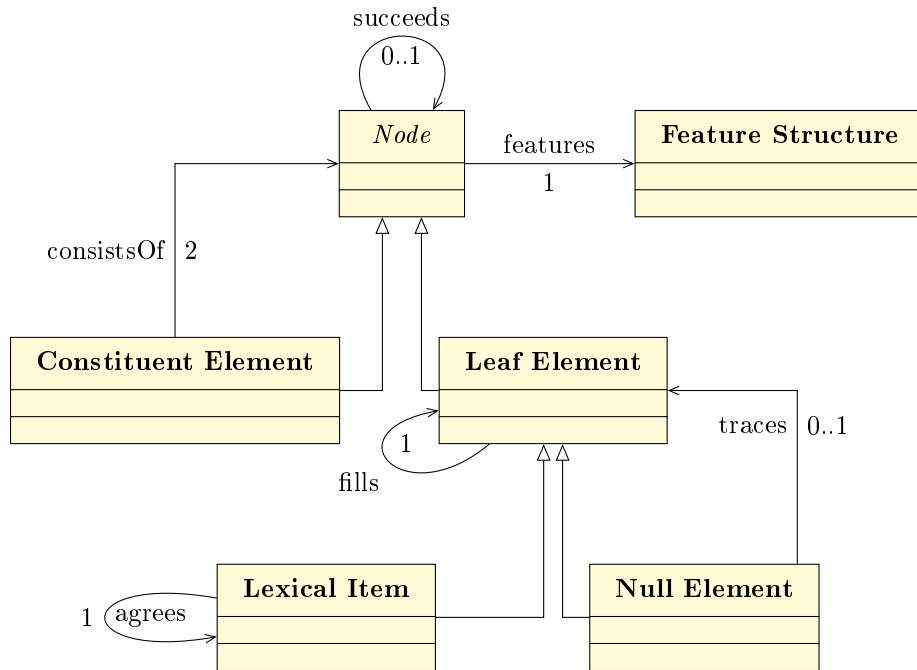


Figure 7: Phrase Structure Model

I distinguish between *constituent* and *leaf* nodes, which are either *lexical items* or *null elements*. The constituents must have two ordered children which can be either another constituent or a leaf node. The ordering of nodes I propose to model, just like in the case of Word Grammar (Section

3), via *successor* relation. Just like in the case of dependency grammar, the node class and any other features are organised in a feature structure.

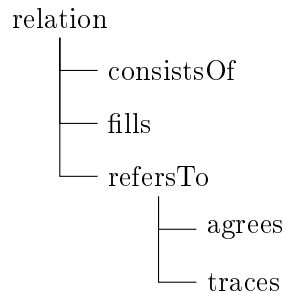


Figure 8: Phrase structure relation taxonomy

The relation in phrase structure is also organised in a taxonomy as depicted in Figure 8.

I did not consider for the moment semantic bindings PS. I do address though the syntax-semantics binding in the next section about Lexical Functional Syntax.

5 Lexical Functional Grammar

Lexical Functional Grammar (LFG) is a type of generative phrase structure grammar. It is a reaction to the direction the transformational grammar was taking. It mainly focuses on syntax with relations to morphology and semantics.

LFG views language as being made up of multiple dimensions of structure: argument, functional and categorial structures. Each of these dimensions, Figure 9 is represented as a distinct structure with its own rules, concepts, and form.

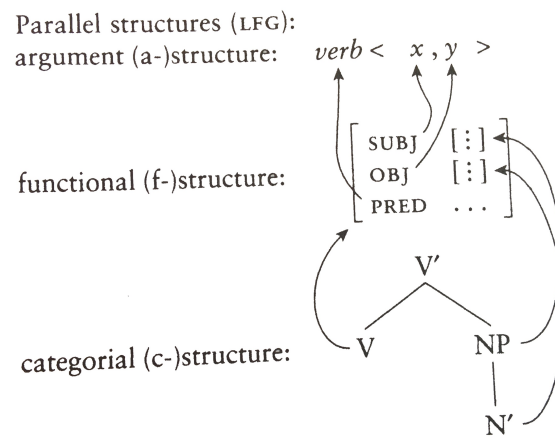


Figure 9: SFG parallel structures

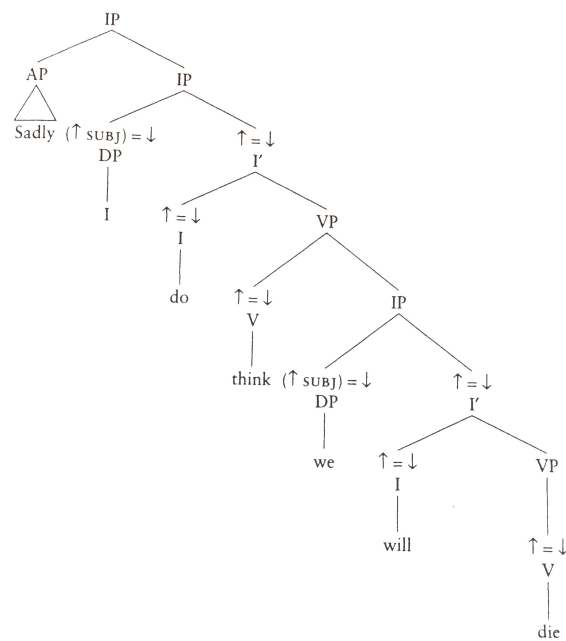


Figure 10: SFG categoryal structure with embedded functional markers

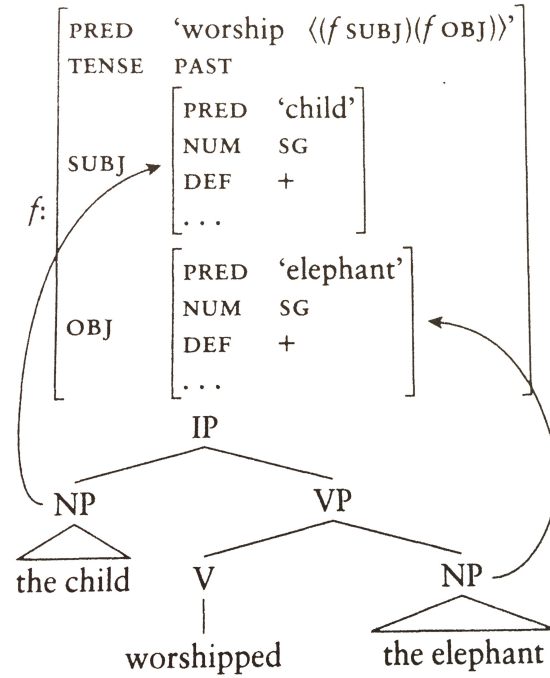


Figure 11: SFG categorical structure mapped onto functional structure represented as feature structure

The categorical structure is constituency based and is very similar to the phrase structure described in Section 4. The functional structure can be modelled as a feature structure described in the Section 2.

6 Systemic Functional meta model

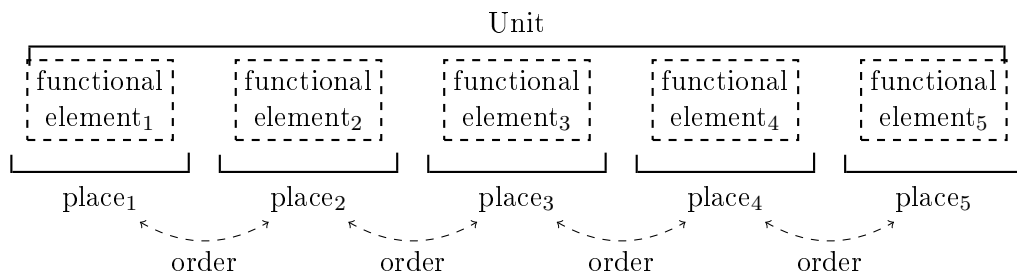


Figure 12: The graphic representation of (unit) structure

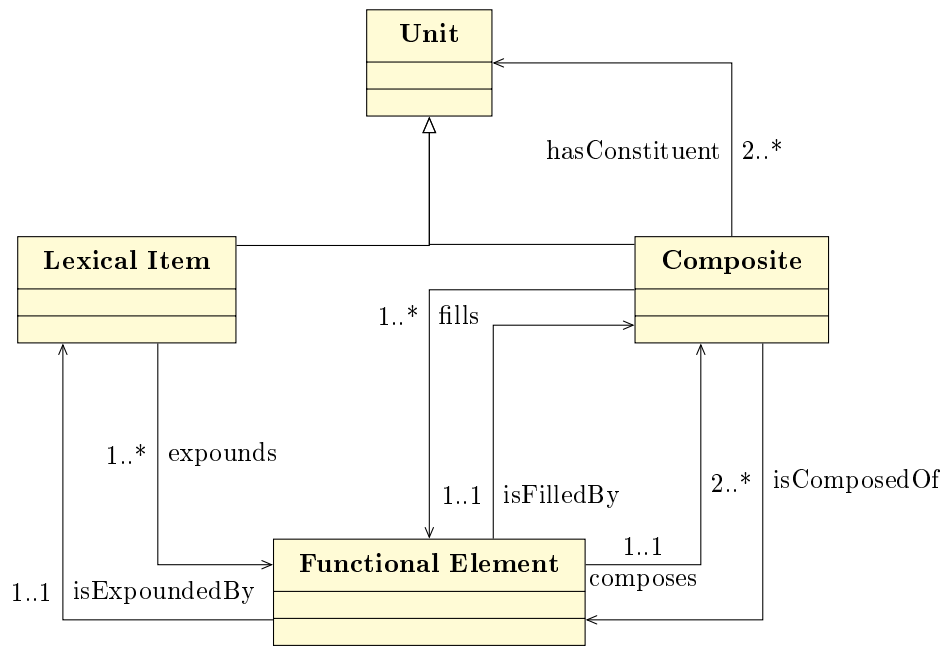


Figure 13: SFL theory of grammar: the meta model

isComposedOf == hasFunction
 functionsAs == fills isFilledBy ==

6.1 Cardiff grammar: instantiation example

- (3) two hundred books.
- (4) She took them in.
- (5) Ask for three croissants and some milk.

6.2 Sydney grammar: instantiation example

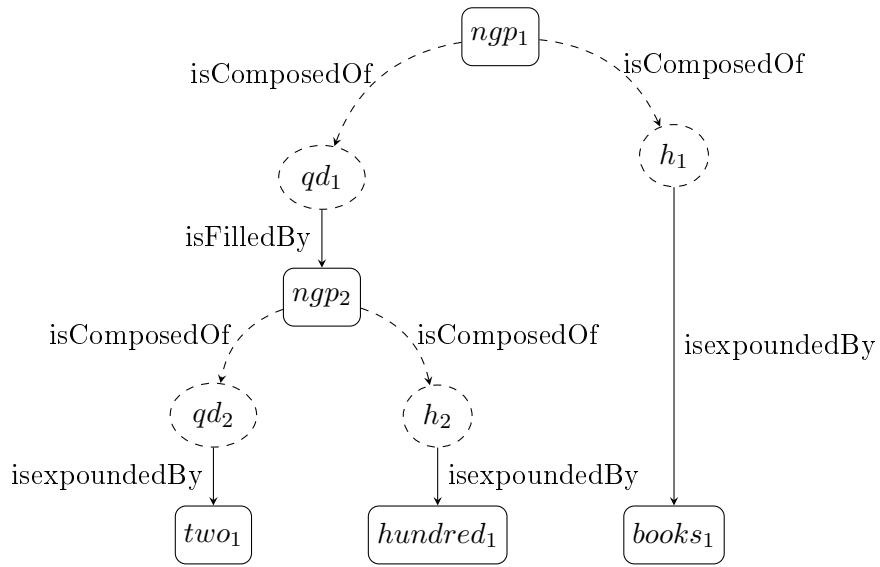


Figure 14: bla bla