5 Linguistic analyses with LTAG

5.1 Lexicalization

The grammar consists of

- A finite set of elementary structure, each of them associated with a lexical item, and
- operations for combining the elementary structures into larger structures.

Other lexicalized grammars for natural languages: categorial grammar.

Grammars that are not lexicalized in the above mentioned sense: LFG, HPSG.

Reasons for lexicalizing a grammar for natural languages:

1. The properties of a word depend on its context.

   E.g., the relation between singular and plural is not a relation between isolated words:

   (9) a. das Blatt
   b. die Blätter

   (10) a. kein Blatt vor den Mund nehmen
   b. ?keine Blätter vor den Mund nehmen

   Different readings depending on the context.

2. The properties of a constituent depend on the lexical items occurring in the constituent.

   E.g., sequences P P in a PP in French allowed only for certain prepositions.

   (11) a. Paul allait vers chez vous
   Paul went towards at you
   'Paul went to you(r place)'

   b. ??Paul allait vers dans les bois
   Paul went towards in the forest
   'Paul went into the forest'

   c. *Paul allait vers à la plage
   Paul went towards at the beach
   'Paul went to the beach'

   Other example: the structure of a VP depends on the subcategorization properties of its verb.

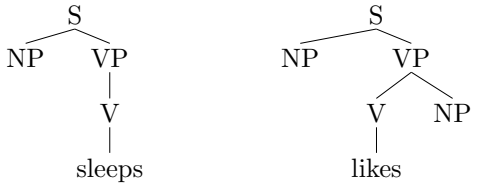
Therefore, a lexical item must specify the possible structures it can occur in.

In a non-lexicalized grammar, depending on different possible contexts, different categories (of verbs etc.) need to be distinguished. The constituent structures defined in a part of the grammar not linked to lexical items need to specify the categories they can combine with.

Ex.: $S \rightarrow NP\ VP$ $VP \rightarrow V_{intrans}$ $VP \rightarrow V_{trans}\ NP$ $V_{intrans} \rightarrow$ sleeps $V_{trans} \rightarrow$ likes

In a lexicalized grammar, the different possible contexts a lexical item can occur in are immediately linked to the lexical item.

Ex.:



5.2 Linguistic principles for LTAG

Grammar formalism: elementary trees + definitions of adjunction and substitution.

For natural languages, the following additional linguistic principles (Frank (1992); Abeillé (2002)) are taken into account concerning the form of the elementary trees. These principles are not part of the formalism.

- **Lexicalization** Each elementary tree has at least one non-empty lexical item, its lexical *anchor*.[9]
- **Predicate argument cooccurrence**: each predicate contains in the elementary tree associated with it argument slots (leaves with non-terminal labels, i.e., substitution nodes or foot nodes) for each of its arguments, i.e., for each of the elements it subcategorizes for including the subject.
- **Semantic anchoring**: elementary trees are not semantically void.
- **Compositionality principle**: an elementary tree corresponds to a single semantical unit.

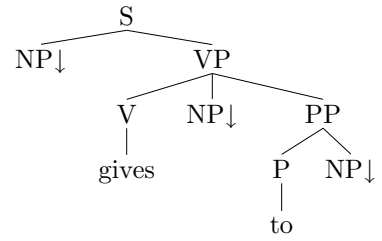
Only on the first two principles there is general agreement. The third one is adopted by most people but for example not completely respected in the XTAG grammar (where we have separate auxiliary trees for complementizers). The fourth one is very arguable. The intuition behind however is a minimality:

Elementary tree minimality An elementary tree contains argument slots only for the arguments of its lexical anchor, for nothing else.

Most argument slots are substitution nodes. Exception: The slots for complement clauses are foot nodes.

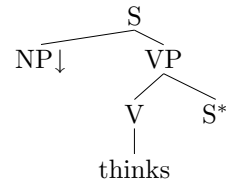
(12) a. John gives a book to Mary

b.



(13) a. John thinks that Mary comes

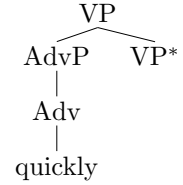
b.



Modifiers always have auxiliary trees with the modified category being the foot node:

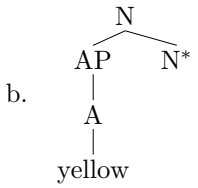
(14) a. John quickly read the book

b.



[9]Empty words are also allowed but only in combination with at least one non-empty lexical element.

(15) a. John read the yellow book

b. 

⇒ two types of auxiliary trees:

- *predicative* auxiliary trees: foot node on the same level as the preterminal of the lexical anchor.
- *modifier* auxiliary trees: foot node immediately dominated by root node.

Exercise 11 *Propose elementary trees for the following sentences:*

(16) John saw a man with a telescope

(17) John buys the house that Bill lives in

(18) Mary took a decision

5.3 Extended domain of locality and factoring of recursion

Constraints in TAG can be stated only within the local domains of elementary trees. But because of adjunction, these local domains can comprise (slots for) elements arbitrarily far away from each other in the final derived tree (TAG has an *extended domain of locality*). Recursive structures are put into separate elementary trees (*factoring of recursion*).

This is crucial for the way LTAG deals with unbounded dependencies and the constraints holding for them: The constraints are not stated in the grammar but they follow from the form of the elementary trees (that is guided by the above-mentioned linguistic principles) and the possibilities of adjunction.

Wh-movement: several constraints hold that are not equally strong (Kroch (1987); Frank (1992))

- Subjacency: a moved element may not cross more than one NP or S (cyclic movement via intermediate traces is allowed)

  (19) a. * a book which Karen met the man that had written t
  b. ? a book which I read Andy's review of t
  c. ? a book which Karen asked who had read t

- Condition on Extraction Domains (CED): extractions from constituents not appropriately governed is not allowed (in particular from subjects and adjuncts) (19a.)
- Empty Category Principle (ECP): a trace must be properly governed ⇒ extractions of subjects from tensed or infinitival clauses with overt complementizers is not possible:

  (20) a. * which book did Lenny say that t was very boring?
  b. * who did Lenny ask whether t had arrived yet?
  c. * who would Lenny have preferred for t to have married his daugther?

  Adjunct extractions out of islands: combination of subjacency and ECP violation

(21) a. * why did Karen ask who had read this book t?

b. * why did Karen know that Steven had read the book t?

LTAG analysis Kroch (1987): slot for moved element and gap in same elementary tree (see Fig. 14).

(22) which book did Harvey say Cecile had read t

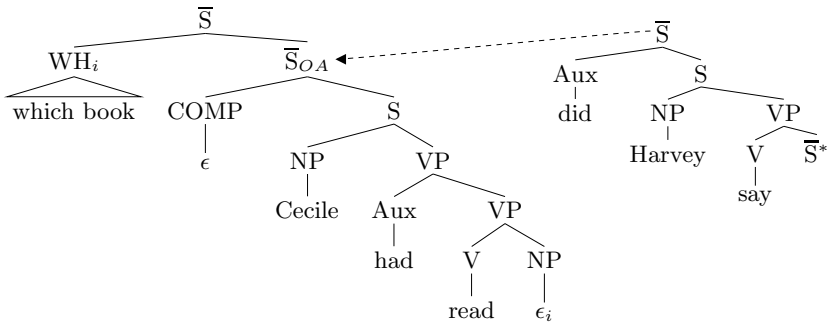


Figure 14: Derivation for (22)

Extraction from adjuncts is blocked with this analysis and the principles for elementary trees:

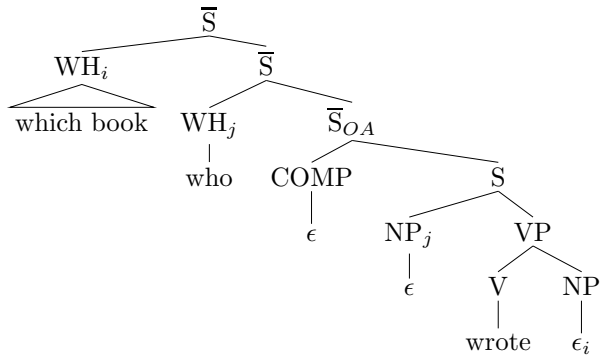
(23) * which movie did Georgette fall asleep after watching t?

Adjuncts are not present in elementary trees of the projections they modify. Instead they are adjoined at VP (minimality of elementary trees). ⇒ there is no elementary tree that might lead to (23).

For a wh-island violation as (24a.), the tree in (24b.) would be needed.[10]

(24) a. ? which book did Judy wonder who wrote t?

b.



This elementary tree satisfies the linguistic principles. But such an elementary tree is not allowed in English since multiple fronted wh-elements are not possible in English.

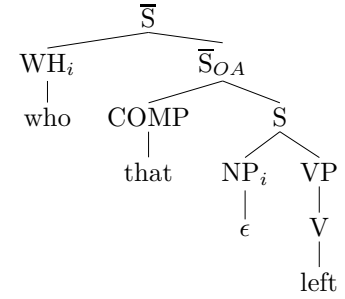
However, languages where such multiple fronted wh-elements are possible, e.g., Romanian, also allow wh-extraction out of an unbounded number of wh-islands.

that-trace effects (ECP violations) also can be excluded by not allowing the corresponding elementary tree (see the tree in (25b.)).

(25) a. * who did Alice say that t left?

[10] Actually, for (24b.) and, similar, (25b.) to be an elementary tree, the wh-phrases must be replaced with substitution nodes. We precompiled substitution here for better readability.

b.



⇒ constraints for unbounded dependencies follow from the elementary trees, i.e., can be stated locally.

Exercise 12 *Consider sentential subjects as in*

(26) That John wins perplexes Bill

Do you prefer adding them by substitution or adding the matrix verb to the sentential subject by adjunction (similar to sentential complements)?
(Note that extraction out of sentential subjects is not allowed.)
Give the elementary tree for perplexes *that you would choose.*

5.4 Constituency and dependencies

The *derived tree* gives the constituent structure of a sentence.

The *derivation tree* records how the elementary trees of the grammar were put together in order to obtain the derived tree. In a lexicalized grammar, each node in the derivation tree corresponds to (at least) one lexical item of the sentence.

For this section we adopt the predicate-argument cooccurrence principle and we assume following Frank (1992) that all functional elements (complementizers, determiners, auxiliaries, negation) are part of the elementary trees of the lexical item they are associated with. Then each substitution/adjunction corresponds to the application of a predicate to one of its arguments. Consequently, the derivation tree gives us the set of predicate-argument dependencies of a sentence. ⇒ the derivation tree is close to a semantic dependency graph (Candito and Kahane 1998).

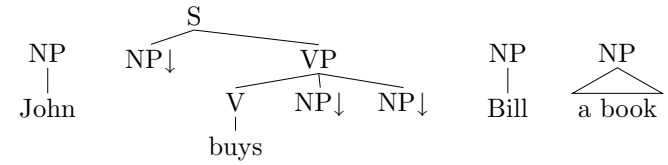
This is crucial since it is the reason why LTAG semantics is computed on the derivation tree.

Some sample derivations with corresponding derivation trees (a. gives the sentence, b. the elementary trees and c. the derivation tree):

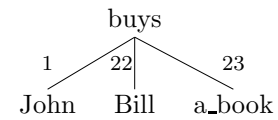
5.4.1 NP and PP complements

(27) a. John buys Bill a book

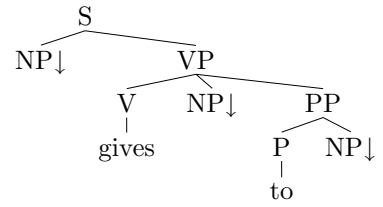
b.

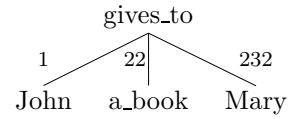


c.



(28) a. John gives a book to Mary

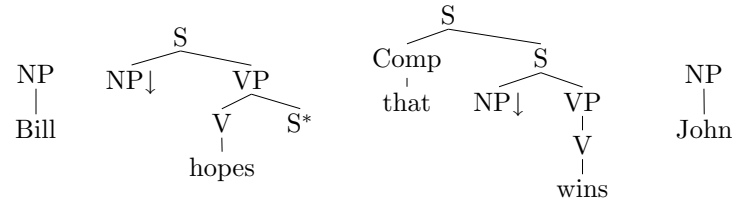
b. 

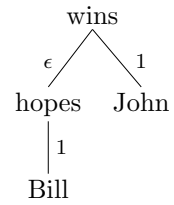
c. 

5.4.2 Clausal complements

Tensed complement clauses:

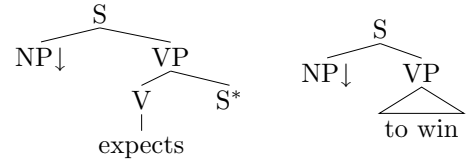
(29) a. Bill hopes that John wins

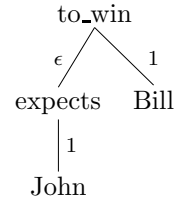
b. 

c. 

ECM verbs (case assignment handled via features):

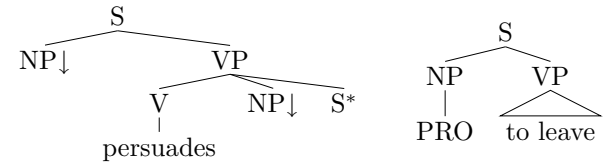
(30) a. John expects [Bill to win]

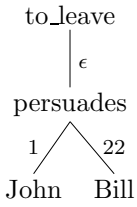
b. 

c. 

Control verbs (the identification of the controller with the argument of the embedded infinitive is handled in the semantics):

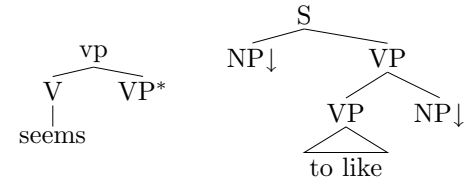
(31) a. John persuades Bill [PRO to leave]

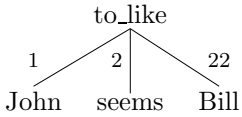
b. 

c. 

5.4.3 Raising verbs

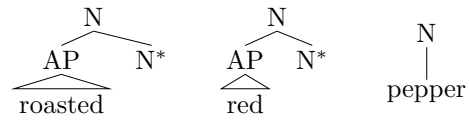
(32) a. John seems to like Bill

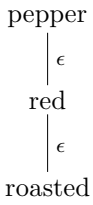
b. 

c. 

5.4.4 Cases where the derivation tree is not the semantic dependency structure

(33) a. roasted red pepper

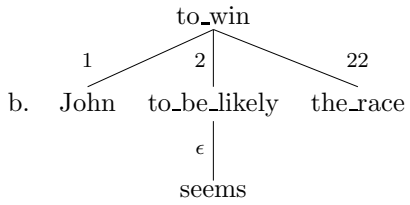
b. 

c. 

This is why Schabes and Shieber (1994) propose an alternative derivation with multiple adjunctions of modifier trees at the same node.

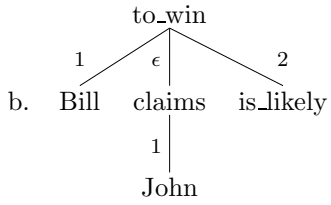
On the other hand, multiple adjunctions are not always desired: The dependency structure in (34b.) is correct.

(34) a. John seems to be likely to win the race



Another problematic case:

(35) a. John claims Bill is likely to win



Exercise 13 *Give the derivation trees for*

(36) John obviously is likely to win

(37) Who do you think Bill says will win the race?