

Introduction to Tree Adjoining Grammar Natural Language Syntax with TAG

Timm Lichte

DGfS-CL Fall School 2011

2. week, 1. session

05.09.2011



- **LTAG as a model for natural language syntax**
 - Principles underlying the shape of elementary trees (Monday)
 - XTAG-analyses of raising/control (Tuesday) and extraction (Wednesday)
- **How to do NLP with an LTAG?**
 - How to implement an LTAG? (Thursday)
 - How to run and test an LTAG? (Friday)

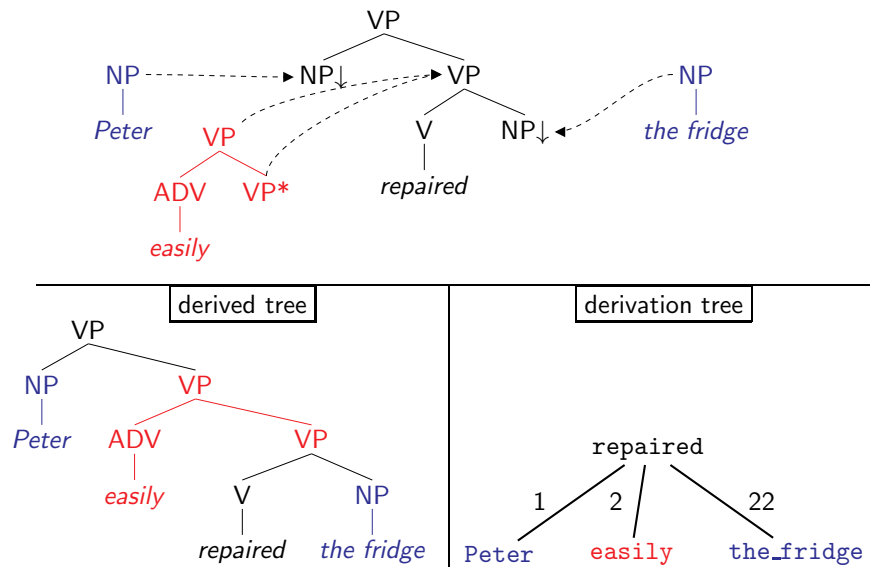
Outline

- 1 Overview: The second week
- 2 Recapitulation of LTAG
- 3 Design principles for elementary trees
- 4 Sample derivations

Outline

- 1 Overview: The second week
- 2 Recapitulation of LTAG
- 3 Design principles for elementary trees
- 4 Sample derivations

Elementary trees, derived tree, derivation tree



Outline

- 1 Overview: The second week
- 2 Recapitulation of LTAG
- 3 Design principles for elementary trees
- 4 Sample derivations

Design principles for elementary trees

What is an elementary tree, and what is its shape?

elementary trees



syntactic/semantic properties of linguistic objects

⇒ Syntactic design principles from [Frank, 2002]:

- Lexicalization
- Fundamental TAG Hypothesis (FTH)
- Condition on Elementary Tree Minimality (CETM)
- θ -Criterion for TAG

⇒ Semantic design principles [Abeillé and Rambow, 2000]

⇒ Design principle of economy

Design principles (1): Lexicalization

Each elementary tree has at least one non-empty lexical item, its lexical **anchor**.

⇒ All widely used grammar formalisms support some kind of lexicalization!

Reasons for lexicalization: [Joshi and Schabes, 1991],

[Schabes and Joshi, 1990]

- **Formal properties:** A finite grammar has finitely many analyses per string (finitely ambiguous).
- **Linguistic properties:** Idiosyncrasies of lexical items can be accounted for more directly.
- **Parsing:** The search space can be delimited (grammar filtering).

Design principles (2): Fundamental TAG Hypothesis

Fundamental TAG Hypothesis (FTH)

Every syntactic dependency is expressed locally within an elementary tree. [Frank, 2002]

“syntactic dependency”

- valency/subcategorization
- modification
- binding
- ...

“expressed within an elementary tree”

- terminal leaf (i.e. lexical anchor)
- nonterminal leaf (substitution node and footnode)
- marking an inner node for obligatory adjunction

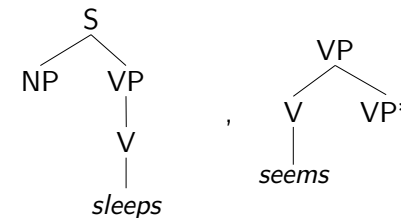
Design principles (4): θ -Criterion for TAG

θ -Criterion (TAG version)

- If H is the lexical head of an elementary tree T, H assigns all of its θ -roles in T.
- If A is a frontier non-terminal of elementary tree T, A must be assigned a θ -role in T.

[Frank, 2002]

⇒ Valency/subcategorization is expressed only with nonterminal leaves!

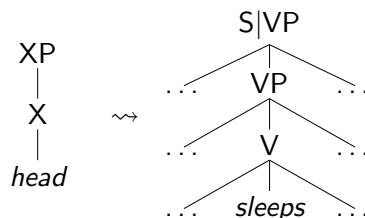


Design principles (3): Condition on Elementary Tree Minimality

Condition on Elementary Tree Minimality (CETM)

The syntactic heads in an elementary tree and their projections must form the extended projection of a single lexical head. [Frank, 2002]

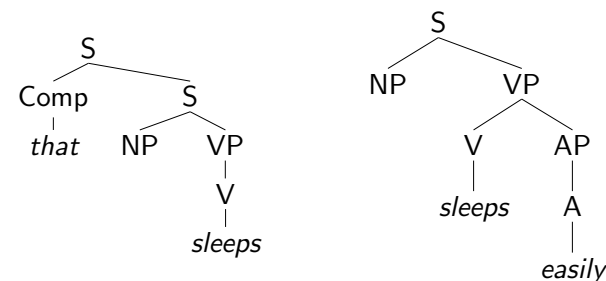
Note: We only use simple, non-extended projections!



Modification and functional elements

How to insert **modifiers** (*easily*) and **functional elements** (complementizers, determiners, do-auxiliaries, ...)?

- Either by separate auxiliary trees (e.g., XTAG grammar),
- or as co-anchor in the elementary tree of the lexical item they are associated with.



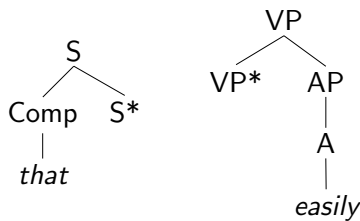
Modification and functional elements

In XTAG, modifiers and functional elements are generally represented by auxiliary trees.

⇒ Footnodes/Adjunctions indicate both complementation and modification.

⇒ Enhancement of the CETM: (see [Abeillé and Rambow, 2000])

core tree (following CETM) + spine



Design principle of economy

Design principle of economy

The elementary trees are shaped in such a way, that the size of the elementary trees and the size of the grammar is minimal.

Principles related to semantics

See [Abeillé and Rambow, 2000].

Predicate-argument cooccurrence:

Each elementary tree associated with a predicate contains a non-terminal leaf for each of its arguments.

Semantic anchoring:

Elementary trees are not semantically void (*to*, *that*.)

Compositional principle:

An elementary tree corresponds to a single semantic unit.

Outline

- 1 Overview: The second week
- 2 Recapitulation of LTAG
- 3 Design principles for elementary trees
- 4 Sample derivations

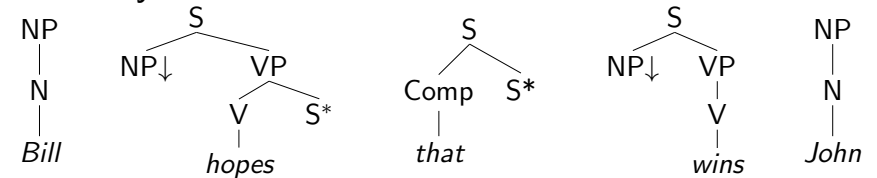
Sample derivations

- **Complementation**
with NPs, PPs, adjectives, clauses (raising, controlling), ...
- **Modification**
with PPs, adjectives, particles, temporal clauses, relative clauses, ...

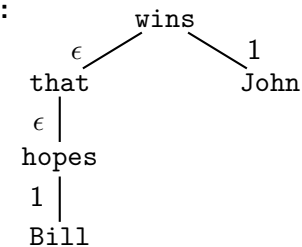
Sample derivations: Sentential complements (1)

(2) Bill hopes that John wins.

Elementary trees:



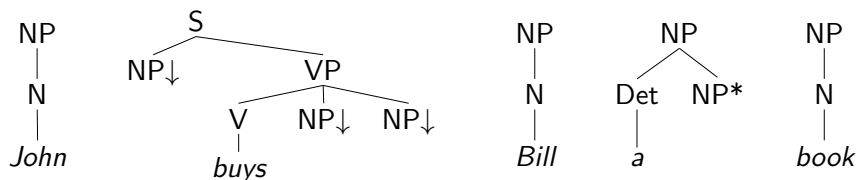
Derivation tree:



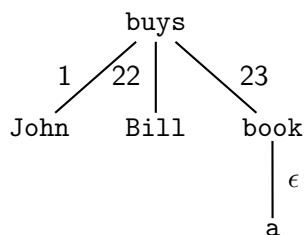
Sample derivations: NP complements

(1) John buys Bill a book.

Elementary trees:



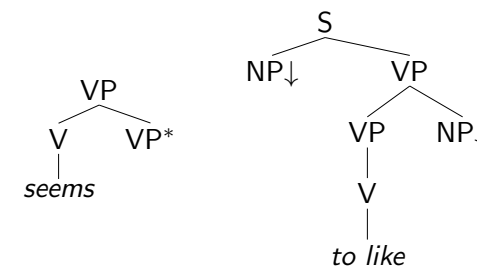
Derivation tree:



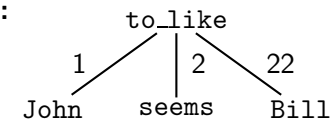
Sample derivations: Sentential complements (2)

(3) John seems to like Bill.

Elementary trees:



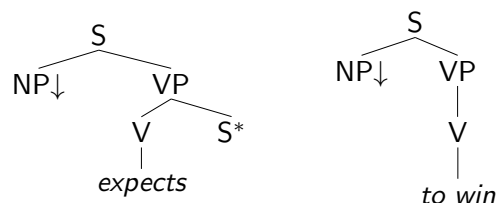
Derivation tree:



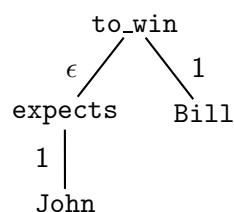
Sample derivations: Sentential complements (3)

(4) John expects [Bill to win].

Elementary trees:



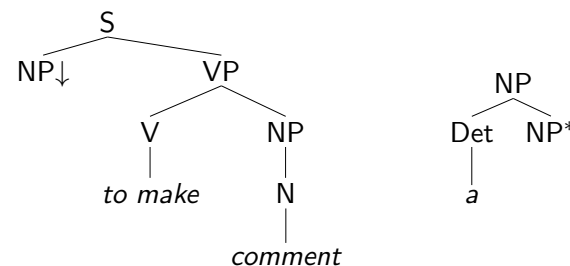
Derivation tree:



Sample derivations: Multiple anchors

Multiword expressions and light verb constructions can be represented by elementary trees with multiple anchors:

(6) John expected [Mary to make a comment].



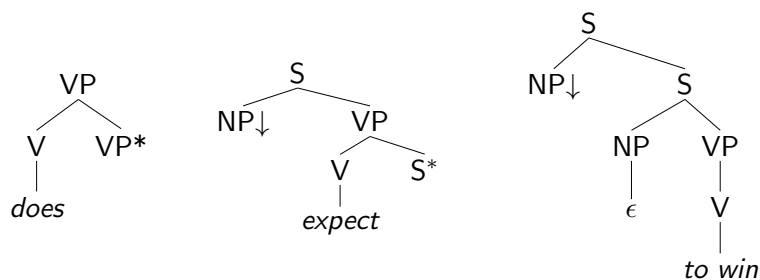
Sample derivations: Sentential complements (4)

Question: Why is the sentential object represented as a footnode?

The sentential object is realised as a foot node in order to allow extractions:

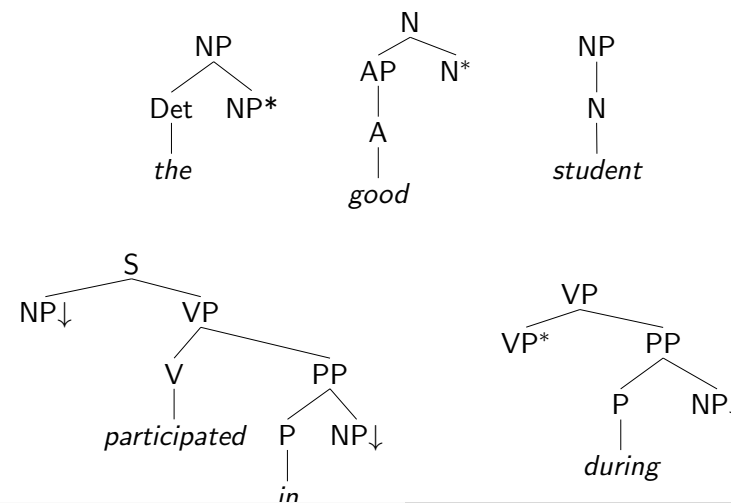
(5) **Who** does John expect **to win**?

Elementary trees:

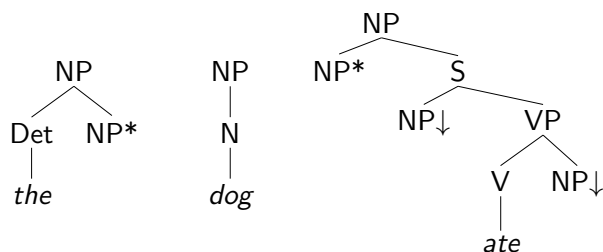


Sample derivations: Modifiers

(7) The good student participated in every course during the semester.



(8) The dog [who ate the cake].



Problem: Extraposed relative clauses:

(1) Somebody; lives nearby [who; has a CD-burner].

- TAG derivations are described by derivation trees.
- In LTAG, elementary trees for lexical predicates contain slots for all arguments of these predicates, for nothing else. Recursion is factored away.
- The derived tree describes the constituent structure while the derivation tree is close to a semantic dependency graph.

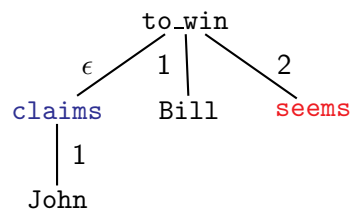
Derivation trees = Semantic dependency structure ?

The derivation tree is not always the semantic dependency structure, due to:

- indiscernibility of complementation and modification in adjunction, and
- missing links.

Example for a missing link:

(2) John **claims** [Bill **seems** to win]



]



Abeillé, A. and Rambow, O. (2000).

Tree adjoining grammar: An overview.

In Abeillé, A. and Rambow, O., editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analyses and Processing*, volume 107 of *CSLI Lecture Notes*, pages 1–68. CSLI Publications, Stanford.



Frank, R. (2002).

Phrase Structure Composition and Syntactic Dependencies.

MIT Press, Cambridge, MA.



Joshi, A. K. and Schabes, Y. (1991).

Tree-Adjoining Grammars and lexicalized grammars.

Technical Report MS-CIS-91-22, Department of Computer and Information Science, University of Pennsylvania.



Schabes, Y. and Joshi, A. K. (1990).

Parsing with lexicalized tree adjoining grammar.

Technical Report MS-CIS-90-11, Department of Computer and Information Science, University of Pennsylvania.