# Unterspezifikation in der Semantik

# Scope Semantics in Lexicalized Tree Adjoining Grammars

Laura Kallmeyer
Heinrich-Heine-Universität Düsseldorf

Wintersemester 2011/2012





**Overview**

1. Lexicalized Tree Adjoning Grammars
   (a) The Formalism
   (b) LTAG and Natural Languages
   (c) Feature-Based TAG
   (d) Extraction Phenomena
2. The Syntax-Semantics Interface
   (a) Unification-Based LTAG Semantics
   (b) Quantifier Scope, Bridge Verbs
   (c) Raising, Control, Adverbs
   (d) Embedded Quantifiers

[Kallmeyer and Romero, 2008]





**LTAG: The Formalism (1)**

*Tree Adjoining Grammars (TAG)*: Tree-rewriting system: set of *elementary* trees with two operations:

- adjunction: replacing an internal node with a new tree. The new tree is an auxiliary tree and has a special leaf, the foot node.
- substitution: replacing a leaf with a new tree. The new tree is an initial tree

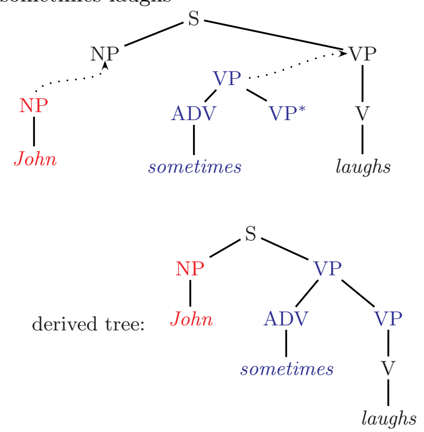
A good LTAG introduction: [Joshi and Schabes, 1997].





**LTAG: The Formalism (2)**

(1) John sometimes laughs

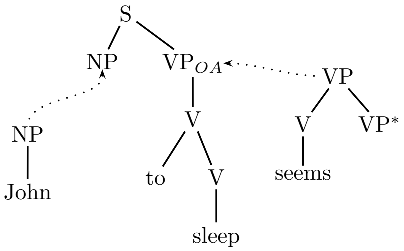


## LTAG: The Formalism (3)

Additionally, adjunction constraints specify for each node

1. whether adjunction is mandatory and
2. which trees can be adjoined.

(2) John seems to sleep








## LTAG and Natural Languages (1)

Important features of LTAG:

- Grammar is lexicalized
- Recursive parts are put into separate elementary trees that can be adjoined (Factoring of recursion, FR)
- Elementary trees can be arbitrarily large, in particular (because of FR) they can contain elements that are far apart in the final derived tree (Extended domain of locality)



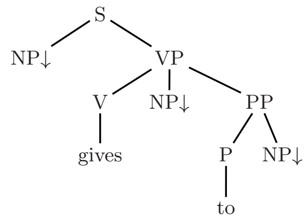


## LTAG and Natural Languages (2)

Elementary trees are extended projections of lexical items (Frank 2002). Recursion is factored away ⇒ finite set of elementary trees.

The elementary tree of a lexical predicate contains slots (non-terminal leaves) for all arguments of the predicate, for nothing more.

(3) John gives a book to Mary








## FTAG (1)

Feature-structure based TAG (FTAG) [Vijay-Shanker and Joshi, 1988]:

Each node has a top and a bottom feature structure (except substitution nodes that have only a top). Nodes in the same elementary tree can share features (extended domain of locality).

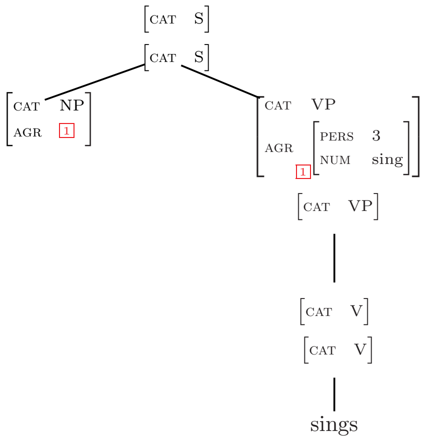
Intuition:

- The top feature structure tells us something about what the node presents within the surrounding structure, and
- the bottom feature structure tells us something about what the tree below the node represents.

In the final derived tree, both must be the same.

## FTAG (2)

Example:

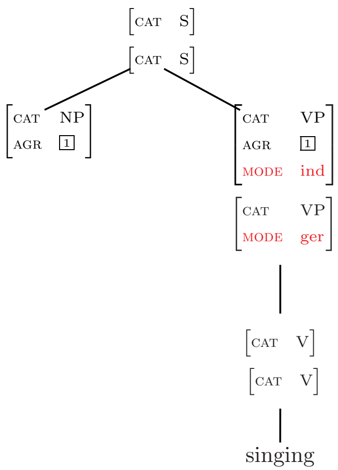







## FTAG (3)

Example:








## FTAG (4)

Unification during derivation:

- **Substitution**: the top of the root of the new initial tree unifies with the top of the substitution node
- **Adjunction**: the top of the root of the new auxiliary tree unifies with the top of the adjunction site,

  and the bottom of the foot of the new tree unifies with the bottom of the adjunction site.
- In the final derived tree, top and bottom unify for all nodes.

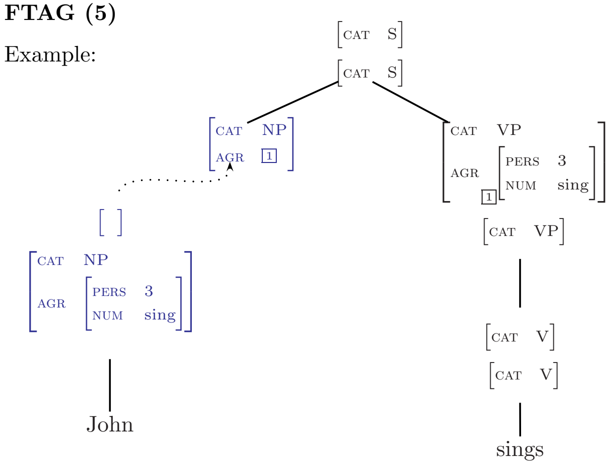
In FTAG, adjunction constraints can be expressed via feature values.





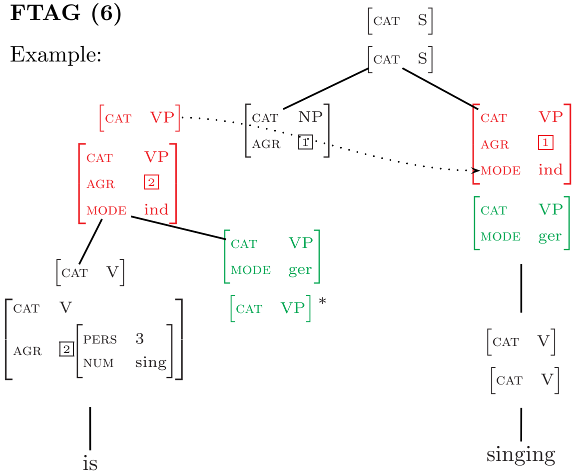
## FTAG (5)

Example:



**FTAG (6)**

Example:








**Unification-Based LTAG Semantics (1)**

Observations:

- Elementary trees for lexcial predicates contain slots for all arguments of the predicate.
- These slots get filled via adjunction or substitution.

⇒ semantic representations are assigned to whole elementary trees (not to single nodes) and the adjunctions and substitutions determine semantic composition.





**Unification-Based LTAG Semantics (2)**

[Kallmeyer and Romero, 2008], [Gardent and Kallmeyer, 2003]: Syntax-Semantics Interface for LTAG

Idea: Each elementary tree is paired with

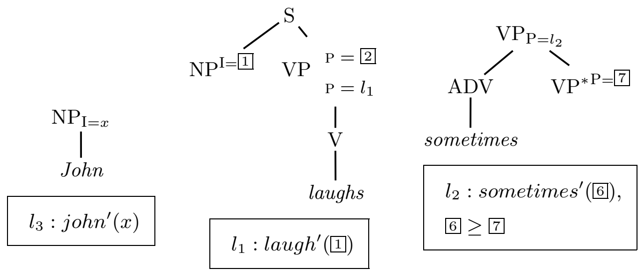
- A set of typed predicate logic expressions and of scope constraints (i.e., constraints on sub-term relations)
- a feature structure that characterizes a) which arguments need to be filled, b) which elements are available as arguments for other elementary trees and c) the scope behaviour.

  The features are linked to positions in the elementary tree.





**Unification-Based LTAG Semantics (3)**



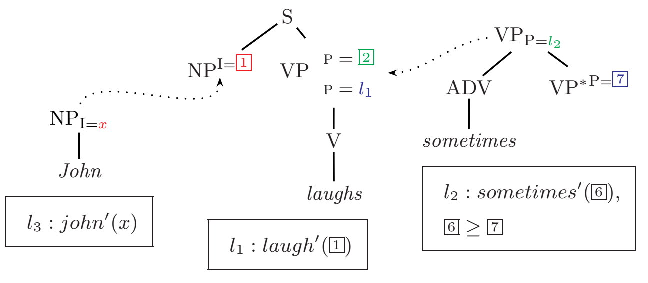
## Unification-Based LTAG Semantics (4)

- There is no proper functional application.
  Instead, depending on substitutions and adjunctions, we perform unifications that lead to assignments for the metavariables in the semantic representations.
- The result can be underspecified. Therefore, a further disambiguation is needed in order to obtain the different readings.





## Unification-Based LTAG Semantics (5)




$\rightsquigarrow$ $l_3 : john(x)', l_1 : laugh'(x), l_2 : sometimes'(\boxed{6})$
$\boxed{6} \geq l_1$





## Unification-Based LTAG Semantics (6)

Disambiguation: Function assigning propositional labels to the remaining propositional metavariables while respecting the scope constraints.

$l_1 : laugh'(x), l_0 : john'(x), l_2 : sometimes'(\boxed{3}),$
$\boxed{3} \geq l_1$

Only one disambiguation: $\boxed{3} \rightarrow l_1$. Leads to

$l_0 : john'(x), l_2 : sometimes'(l_1 : laugh'(x))$

The resulting set is interpreted conjunctively.

This yields $john'(x) \wedge sometimes'(laugh'(x))$



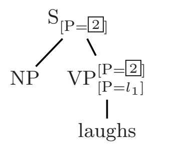


## Unification-Based LTAG Semantics (7)

Global features: Some features are linked to elementary trees as a whole.

Example: Minimal proposition contributed by a verb. ⇒ global feature MINS.

- Each tree has global features.
- Requests for global features of other trees can be put on specific node positions.




$[\text{MAXS} = \boxed{1}, \text{MINS} = l_1]$

$l_1 : laugh'(j),$
$\boxed{1} \geq \boxed{2}$

### Quantier Scope (1)

Quantificational NPs can in principle scope freely; their scope is not directly linked to their surface position.

(4) Exactly one student admires every professor
$\exists > \forall, \forall > \exists$

(5) Two policemen spy on someone from every city
$\forall > \exists > 2$ (among others)

(6) John seems to have visited everybody
*seem* $> \forall, \forall >$ *seem*

Attitude verbs block the scope of embedded quantificational NPs:

(7) Mary thinks John likes everybody
thinks > everybody, *everybody > thinks





### Quantifier Scope (2)

Two things must be guaranteed:

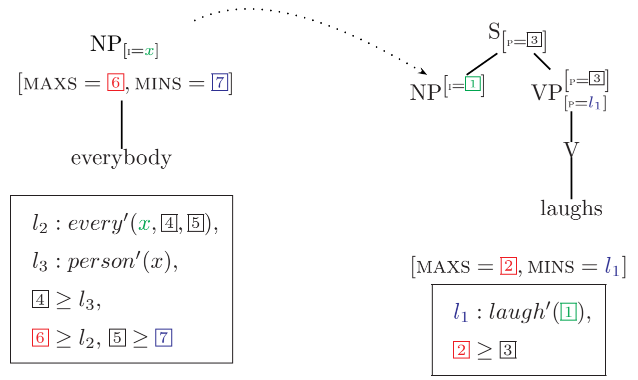
1. the proposition to which a quantifier attaches must be in its nuclear scope
2. a quantifier cannot scope higher than the next finite clause

LTAG analysis: scope window delimited by some maximal scope MAXS and some minimal scope MINS for a quantifier.





### Quantifier Scope (3)








### Quantifier Scope (4)

Result:

$l_1 : laugh'(x)$,
$l_2 : every'(x, \boxed{4}, \boxed{5})$, $l_3 : person'(x)$
$\boxed{2} \geq l_1, \boxed{2} \geq l_2$,
$\boxed{4} \geq l_3, \boxed{5} \geq l_1$

Disambiguation:
$\boxed{2} \rightarrow l_2, \boxed{4} \rightarrow l_3, \boxed{5} \rightarrow l_1$
yields $every'(x, person'(x), laugh'(x))$

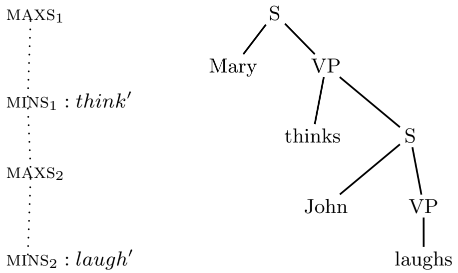
This analysis generates an underspecified representation for

(8) Exactly one student admires every professor

**Bridge verbs (1)**

In LTAG, because of the extended domain of locality, it is straightforward to define some elements as blocking the embedded scope window and defining a new one for higher quantifiers:

(9) Mary thinks John laughs

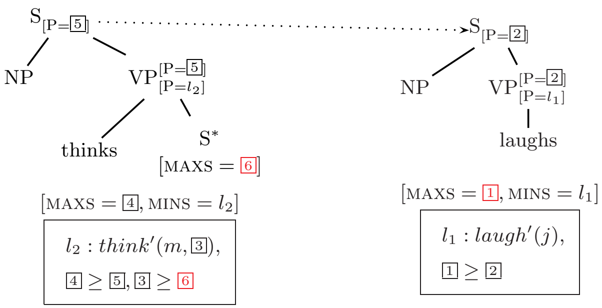







**Bridge verbs (2)**

(10) Mary thinks John laughs








**Bridge verbs (3)**

(11) Mary thinks John likes everybody

- *everybody* has to take scope within the scope window of *likes*.
- *thinks* scope over the MAXS of *likes*, and consequently also over *everybody*

⇒ only one scope order:

(12) $think'(m, every'(x, person'(x), like'(j, x)))$





**Adverbs, Raising and Control Verbs (1)**

In contrast to quantificational NPs, the scope of a quantificational element attached to the verbal spine is fully determined by the surface syntax:

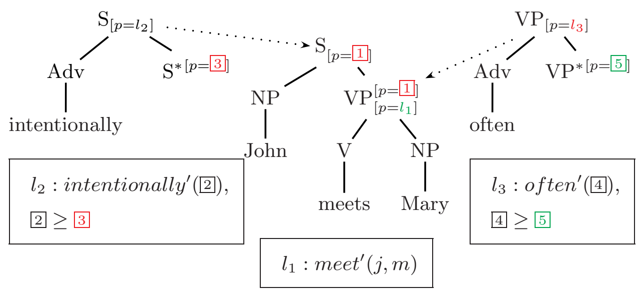
(13) John seems to sometimes laugh
   a. $seem'(sometimes'(laugh'(j)))$
   b. * $sometimes'(seem'(j, laugh'(j)))$

(14) John wants Mary to sometimes laugh.
   a. $want'(j, sometimes'(laugh'(m)))$
   b. * $sometimes'(want'(j, laugh'(m)))$

Relevant features: P features that decorate the verbal spine.

### Adverbs

(15) Intentionally, John often meets Mary



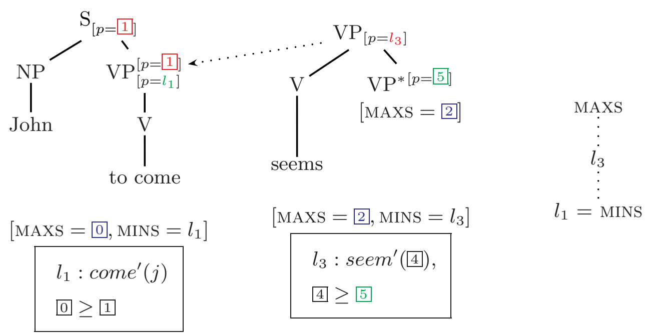





### Raising verbs

(16) John seems to come

Raising verbs are similar to adverbs:








### Control verbs (1)

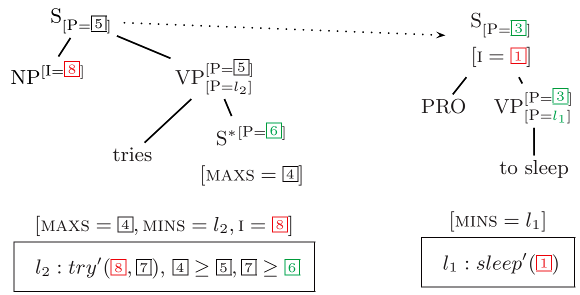
- Scopal behaviour as adverbs and raising verbs: embedding of lower P and providing of a new, higher P value.
- Passes its MAXS to the MAXS of the embedded infinitive (no blocking of quantifier scope).
- Provides the controller as a global I feature.





### Control verbs (2)

(17) John tries to sleep



**Embedded quantifiers (1)**

(18) Two policemen spy on someone from every city.

So far, we have only excluded the reading

$^*$*some* > *two* > *every*

But: According to the literature, the following order is not possible either:

$^*$*every* > *two* > *some*

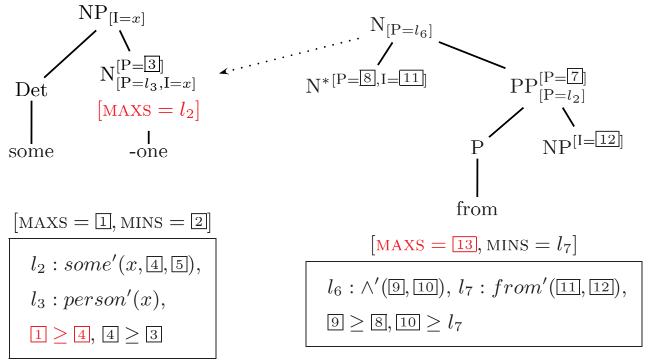
LTAG analysis: exclude this order by deriving a constraint saying that the maximal nuclear scope of *every*′ is the *some*′ proposition.
⇒ *every*′ can take scope over *some*′ but if it does so, then it has to take immediate scope over *some*′.

The constraint concerning the max. scope becomes $maxs \geq svar$, where *svar* is the nuclear scope variable.





**Embedded quantifiers (2)**




1. $\boxed{1} \geq \boxed{4}$ instead of $\boxed{1} \geq l_2$, and 2. $l_2$ gets passed to adjoining N modifiers as their MAXS.





**Embedded quantifiers (3)**

(19)

$$\begin{array}{l}
l_1 : spy'(x,y),\\
l_2 : 2(x,\boxed{3},\boxed{4}),\ l_3 : policeman'(x)\\
l_4 : some'(y,\boxed{7},\boxed{8}),\ l_5 : person'(y) \wedge \boxed{18},\\
l_7 : from'(y,z)\\
l_8 : every'(z,\boxed{13},\boxed{14}),\ l_9 : city'(z)\\
\boxed{1} \geq l_1,\\
\boxed{3} \geq l_3,\ \boxed{1} \geq \boxed{4},\ \boxed{4} \geq l_1,\\
\boxed{7} \geq l_5,\ \boxed{1} \geq \boxed{8},\ \boxed{8} \geq l_1\\
\boxed{18} \geq l_7,\\
\boxed{13} \geq l_9,\ l_4 \geq \boxed{14},\ \boxed{14} \geq l_7
\end{array}$$





# References

[Abeillé et al., 2010] Abeillé, A., Kallmeyer, L., and Rambow, O. (2010). *Introduction to Tree Adjoining Grammars*. Morgan and Claypool. In preparation.

[Frank, 2002] Frank, R. (2002). *Phrase Structure Composition and Syntactic Dependencies*. MIT Press, Cambridge, Mass.

[Gardent and Kallmeyer, 2003] Gardent, C. and Kallmeyer, L. (2003). Semantic Construction in FTAG. In *Proceedings of EACL 2003*, pages 123–130, Budapest.

[Joshi and Schabes, 1997] Joshi, A. K. and Schabes, Y. (1997). Tree-Adjoning Grammars. In Rozenberg, G. and Salomaa, A., editors, *Handbook of Formal Languages*, pages 69–123. Springer, Berlin.

[Kallmeyer and Romero, 2008] Kallmeyer, L. and Romero, M. (2008). Scope and situation binding in LTAG using semantic unification. *Research on Language and Computation*, 6(1):3–52.

[Kroch, 1989] Kroch, A. (1989). Asymmetries in long-distance extraction in a Tree Adjoining Grammar. In Baltin and Kroch, editors, *Alternative Conceptions of Phrase Structure*. University of Chicago.

[Vijay-Shanker and Joshi, 1988] Vijay-Shanker, K. and Joshi, A. K. (1988). Feature structures based tree adjoining grammar. In *Proceedings of COLING*, pages 714–719, Budapest.