Tutorial:
TAG Semantics

Maribel Romero

University of Konstanz

maribel.romero@uni-konstanz.de

TAG+9

June 6-8, 2008

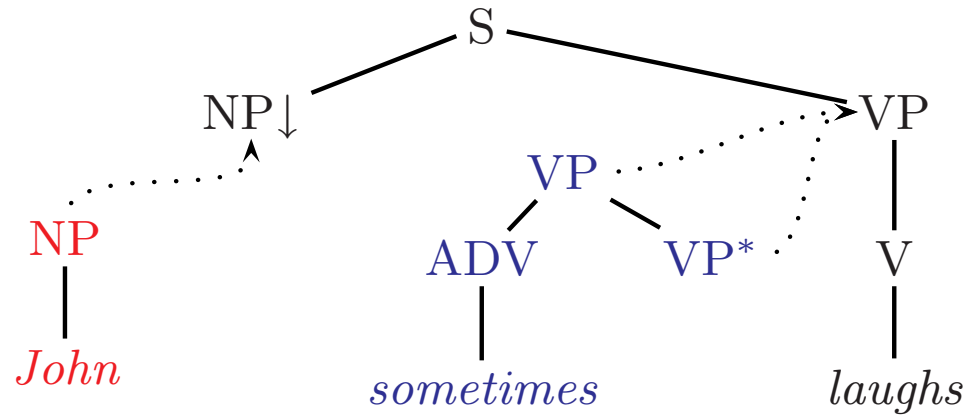
Plan

1. Summary of LTAG syntax
   relevant for LTAG semantics
2. Semantic Composition in LTAG
3. Data on scope
4. Analysis of the scope data
5. Conclusions

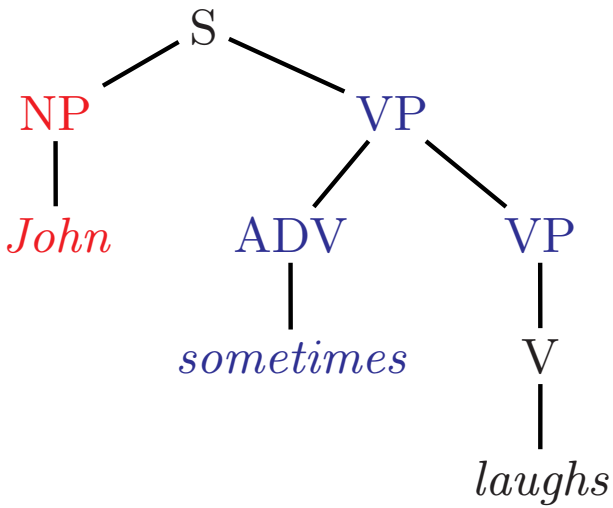
Syntactic Architecture

- Two syntactic operations:
  a. Substitution: replacing a leaf with a new tree
  b. Adjunction: replacing an internal node with a new tree
- The result / output of carrying out the substitutions and adjunctions is the DERIVED TREE.
- The history of how the elementary trees were put together is recorded in the DERIVATION TREE.

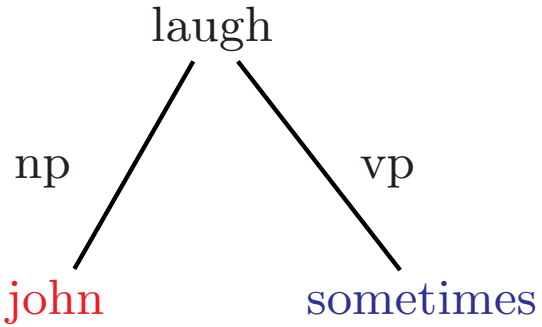
(1) John sometimes laughs



derived tree:

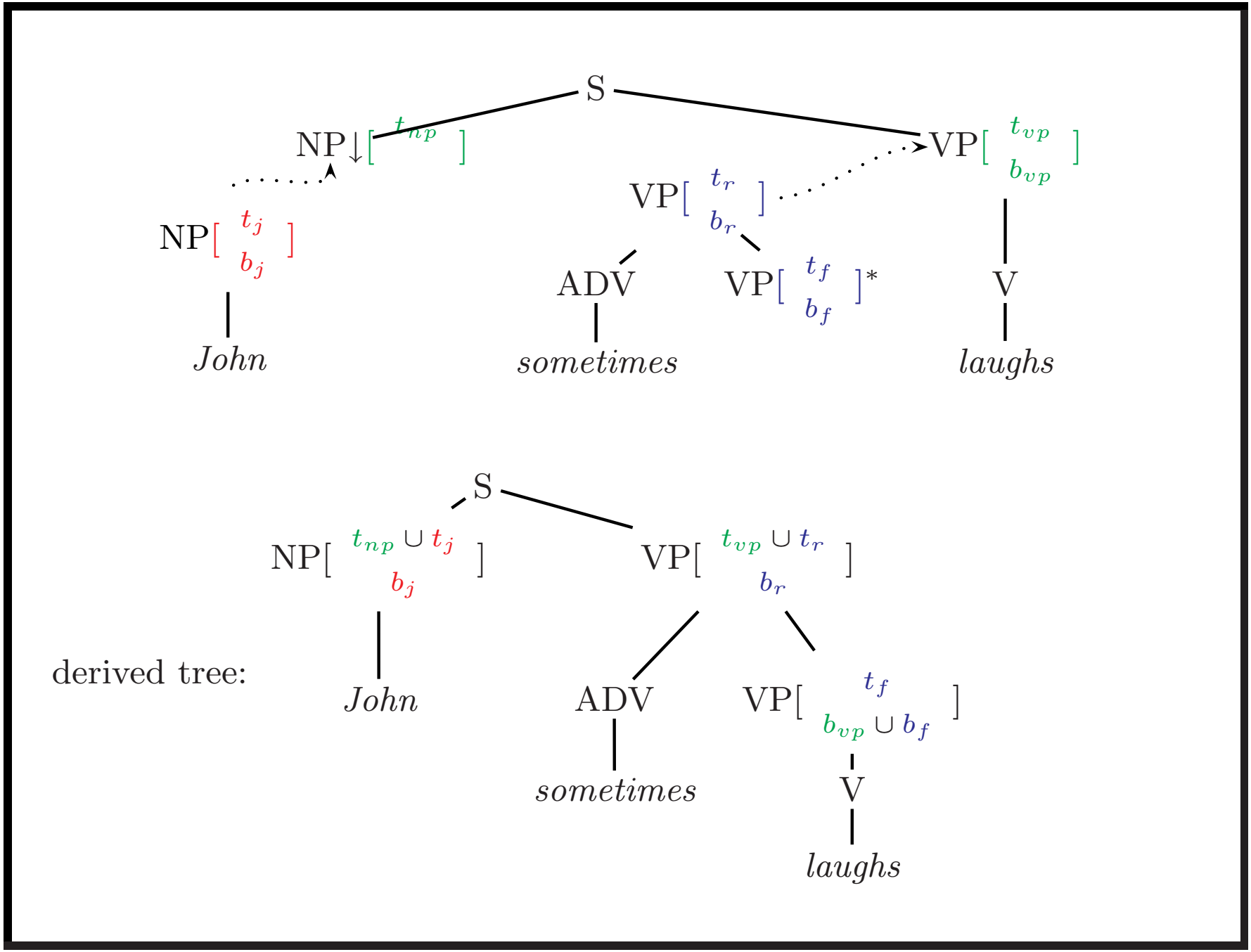


derivation tree:



Feature-structure based TAG (FTAG): Feature unification in syntax

- 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).
- 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.


S
NP↓[t_{np}]
VP[t_{vp} / b_{vp}]
NP[t_j / b_j]
VP[t_r / b_r]
ADV
VP[t_f / b_f]*
V
John
sometimes
laughs
derived tree:
S
NP[$t_{np} \cup t_j$ / b_j]
VP[$t_{vp} \cup t_r$ / b_r]
John
ADV
VP[t_f / $b_{vp} \cup b_f$]
sometimes
V
laughs

Plan

1. Summary of LTAG syntax
   relevant for LTAG semantics
2. **Semantic Composition in LTAG**
3. Data on scope
4. Analysis of the scope data
5. Conclusions

Semantic composition (1)

- The DERIVATION TREE records the history of how the elementary trees –the syntactic-semantic units– are put together.

  $\Rightarrow$ Compute semantics (partly/exclusively) on DERIVATION TREE
- Two main avenues:
  - Derivation tree as **sole** syntactic input to semantics (Joshi and Vijay-Shanker 1999; Kallmeyer and Joshi 2003; Kallmeyer and Romero 2008).
  - Synchronous TAG: the derivation tree serves as **partial** (or underspecified) input to build both the derived syntactic tree and the "derived semantic tree", isomorphically. The ordering specifications made during the construction of the derived syntactic tree affect the derived semantic tree. (Shieber 1994; Shieber and Schabes 1990; Nesson and Schieber 2006)

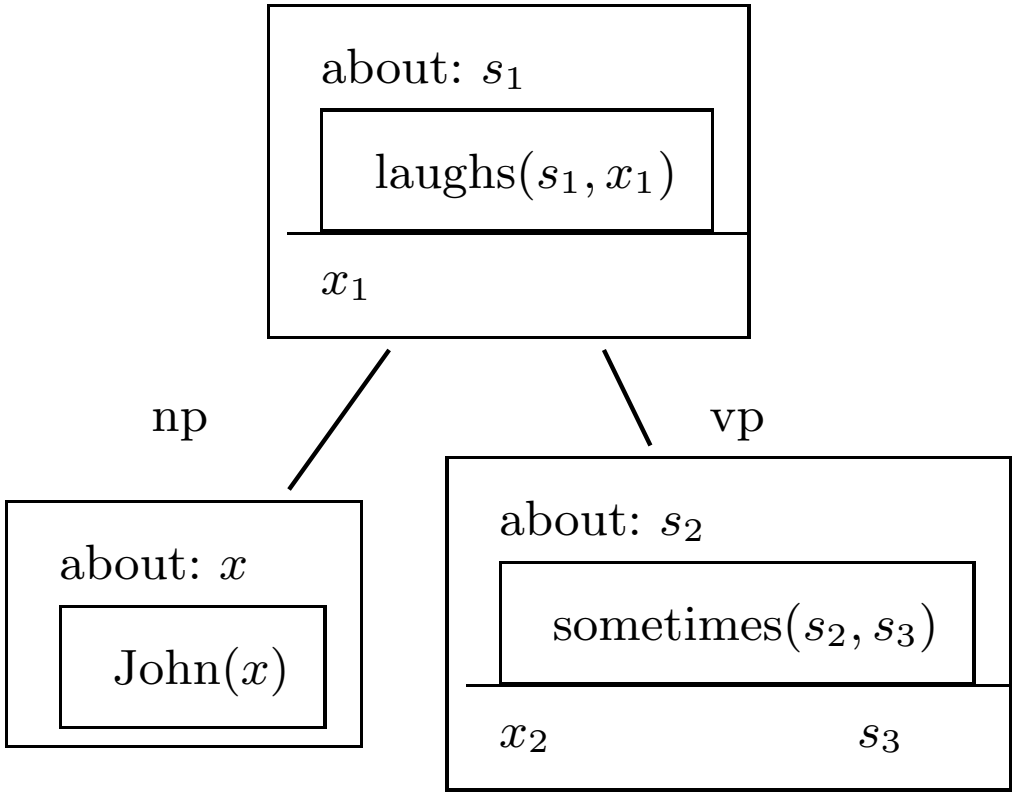
Semantic composition (2)

- Here we will concentrate on the approach that takes the derivation tree as sole syntactic input.
- Each elementary tree has a (set of) formula(s) as its semantic representation. When elementaries trees compose, their formulas are interpreted conjunctively.
- Furthermore, when trees compose, semantic information has to be passed from one to the other. This will be done by:
  - associating each elementary tree to an array of variables or to a semantic feature structure description (cf. λ-abstraction), and
  - performing unifications –i.e., identifications– between the variables or between the semantic feature structutes (cf. λ-conversion).

    $\Rightarrow$ Sem composition = conjunction + sem unification.

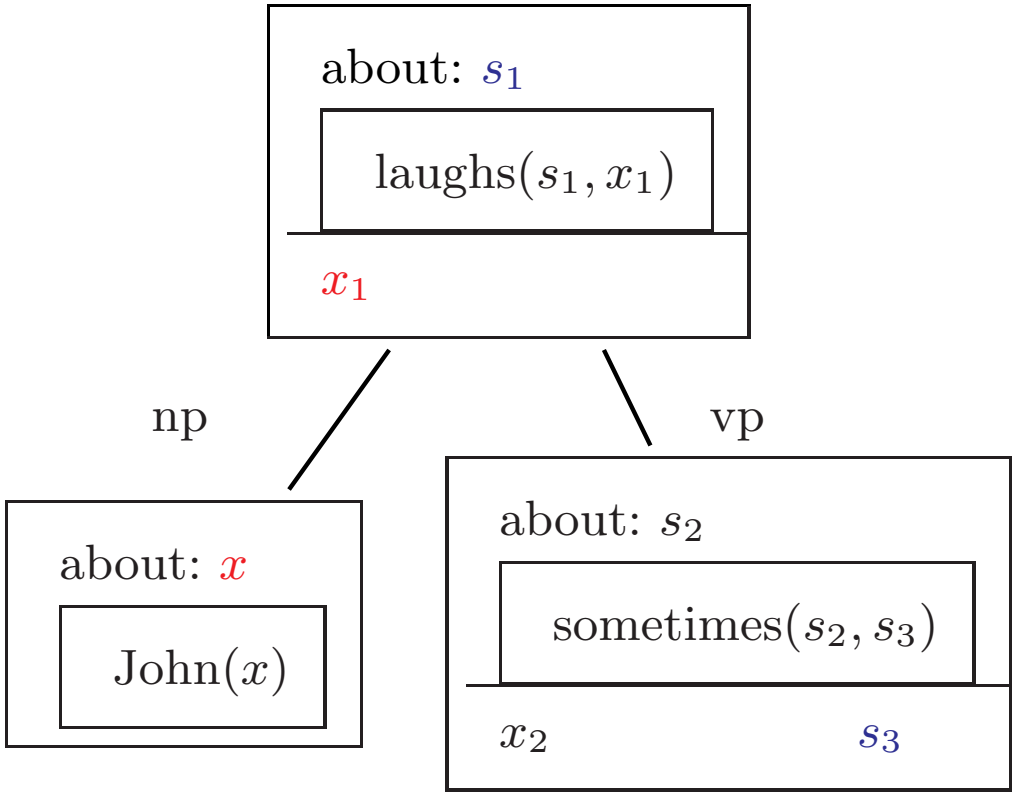
Semantic composition (3.1): A la Joshi and Vijay-Shanker 1999

(1) John sometimes laughs



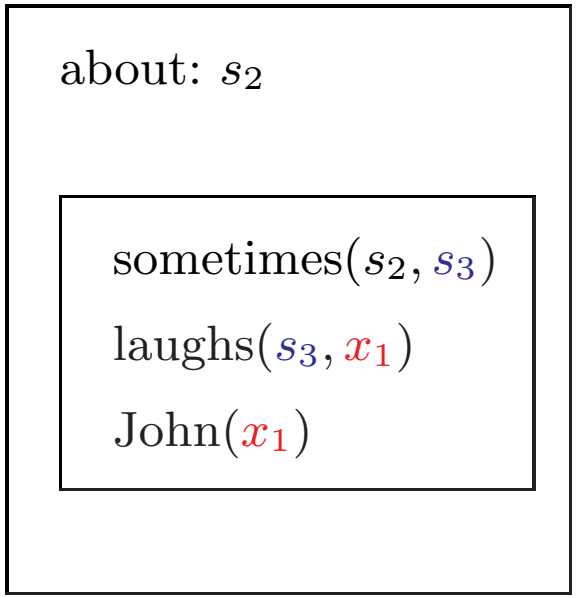
Semantic composition (3.2): A la Joshi and Vijay-Shanker 1999

(1) John sometimes laughs



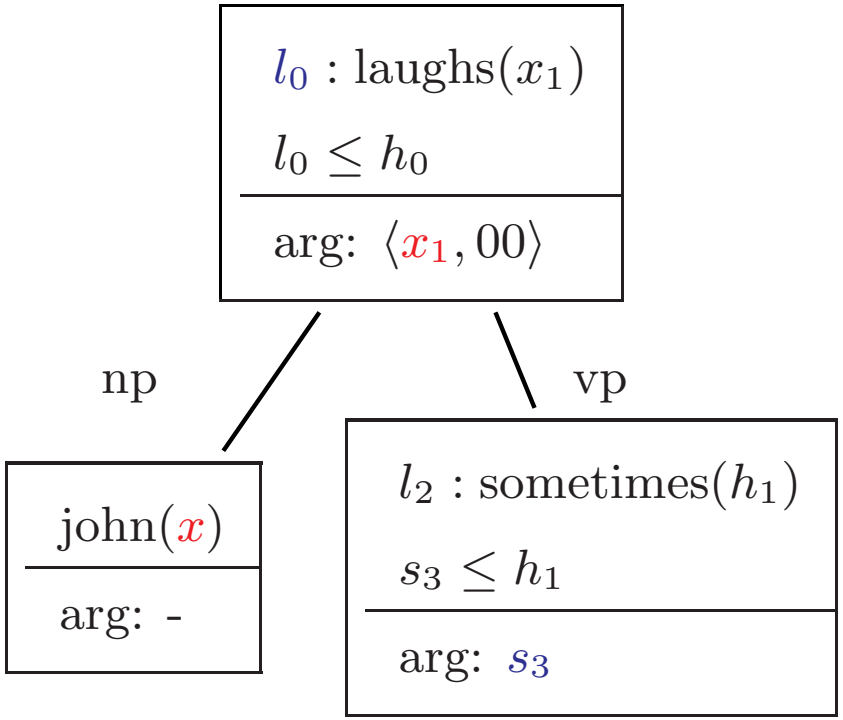
Semantic composition (3.3): A la Joshi and Vijay-Shanker 1999

(1) John sometimes laughs



Semantic composition (4.1): A la Kallmeyer and Joshi 2003

(1) John sometimes laughs



Semantic composition (4.2): A la Kallmeyer and Joshi 2003

(1) John sometimes laughs

$$
\begin{array}{l}
l_0 : \text{laughs}(x) \\
\text{john}(x) \\
l_2 : \text{sometimes}(h_1) \\
l_0 \leq h_1 \\
l_0 \leq h_0
\end{array}
$$

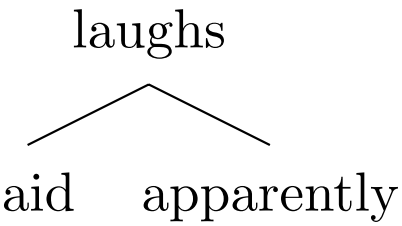
Semantic composition (5.1): The missing link problem

- Case A:

  (2a) John, Paul claims Mary seems to love

  (2b) John said Mary apparently laughs

  Derivation tree for (2b):



  Desired semantics (simplified): $\text{said}(j, \text{apparently}(\text{laughs}(m)))$

  Impossible reading: $\text{apparently}(\text{said}(j, \text{laughs}(m)))$

  QUESTION: Are the following counterexamples to Case A?

  (3) Sandy rarely visited a friend because of El Niño.

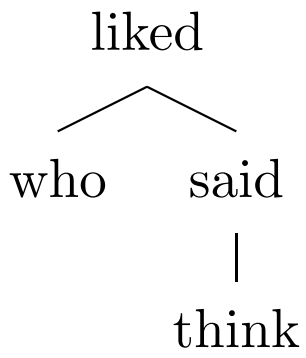
  (4) Usually, Pat allegedly drives a cadillac.

Semantic composition (5.2): The missing link problem

- Case B:

  (3) Who does Paul think John said Bill liked?

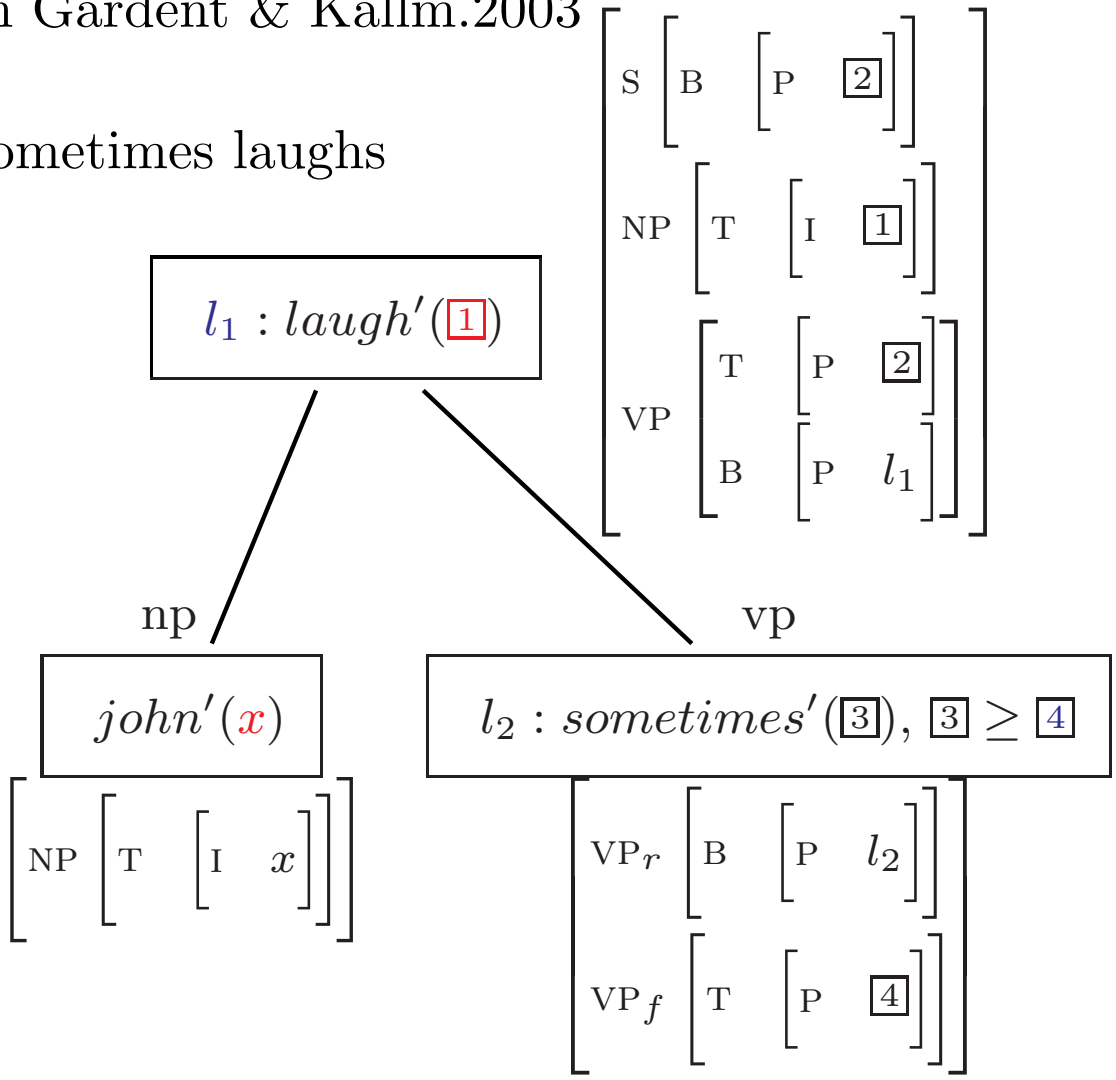
  Derivation tree:



  Desired semantics (simplified):
  $\text{who}(x, \text{think}(p, \text{said}(j, \text{like}(b, x))))$

Semantic composition (6.1): A la Kallmeyer & Romero 2008 building on Gardent & Kallm.2003

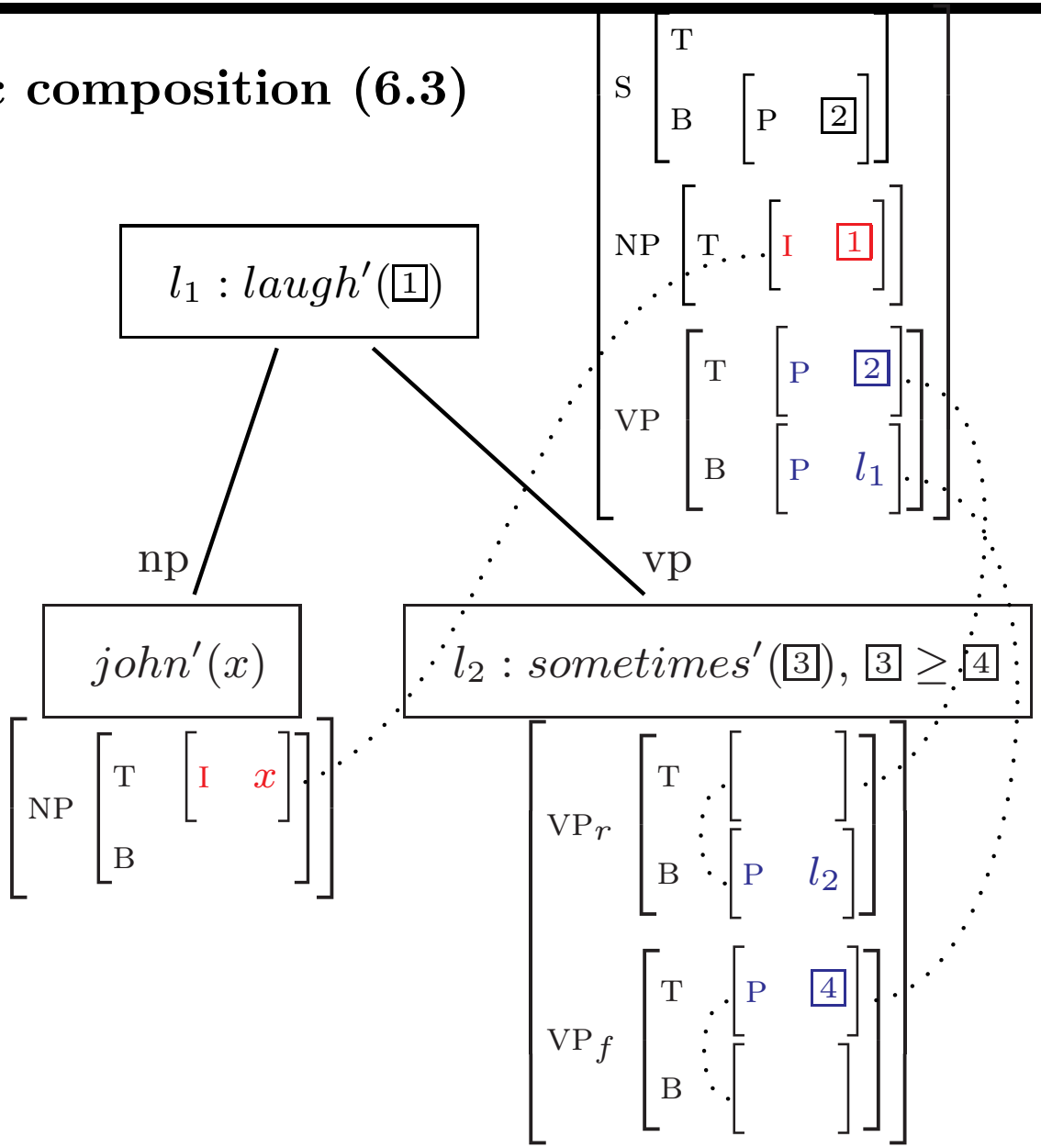
(1) John sometimes laughs



Semantic composition (6.2)

- Semantic feature structure unification proceeds parallel to syntactic feature structure unification.
- For each edge in the derivation tree from γ_1 to γ_2 with position p:
  - Substitution: the top of p in γ_1 and the top of the root in γ_2 are identified
  - Adjunction: the top of p in γ_1 and the top of the root in γ_2 are identified, and the bottom of p in γ_1 and the bottom of the foot node of γ_2 are identified.
- Furthermore, for all γ and all p in γ such that there is no edge with position p from γ to some other tree: the top and bottom of $\gamma.p$ are identified.

Semantic composition (6.3)



Semantic composition (6.4)

Identifications: $\boxed{1} = x$ (substitution), $\boxed{2} = l_2, \boxed{4} = l_1$ (adjunction and top-bottom unification)

After unification the union of the semantic representations is built and the assignments obtained from the unifications is applied to it.

Result:

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

Underspecified representation.

Semantic composition (6.5)

Disambiguation: Function δ from the set of propositional metavariables to the set of propositional labels such that after applying it, the resulting subordination is a partial order (Bos 1996, among others).

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

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

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

The resulting set is interpreted conjunctively.

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

Plan

1. Summary of LTAG syntax relevant for LTAG semantics
2. Semantic Composition in LTAG
3. **Data on scope**
4. Analysis of the scope data
5. Conclusions

Data on quantifier scope (1)

- Quantificational NPs (non-nested): can in principle scope freely; their scope is not directly linked to their surface position.
- Quantificational elements attached to the verbal spine (adverbs, raising verbs, attitudes verbs, control verbs): fix surface scope; scope over everything that is lower on the verbal spine.

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

(3) John seems to sometimes laugh:
$seem > sometimes, sometimes \not> seem$

(4) John said Mary apparently laughs:
$say > apparently, apparently \not> say$

(5) John seems to have visited everybody: $seem > \forall, \forall > seem$

Data on quantifier scope (2)

For quantificational NPs, two things must be guaranteed:

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

(6) A student said you met every professor: $a > every, every \not> a$

(7) A student wants to meet every professor: $a > every, every > a$

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

Data on quantifier scope (3)

Furthermore, when an NP_2 is nested inside an NP_1, the embedded NP_2 is allowed to scope above NP_1, but only immediately above NP_1 (Larson 1987).

(8) Two policemen spy on someone from every city

(9)
a. $2 > \exists > \forall$
b. $\exists > \forall$,2
c. $2 > \forall > \exists$
d. $\forall > \exists > 2$
e. $* \forall > 2 > \exists$

QUESTION: Why do we have the partially ordered reading (b) as opposed to two separate, fully ordered readings (b') $\exists > \forall > 2$ and (b") $\exists > 2 > \forall$?

Empirical generalizations

1. For elements on VP-spine, scope is fully determined by surface syntax: they take scope exactly where they appear.

2. For NPs, scope is underspecified within a scope window. The lower limit is the predicate the NP combines with. The upper limit depends on the syntactic configuration up the tree:

   2.1 When the predicate the NP combines is finite, that is, when the first operator up the tree is an attitude verb like *say* and *think (that)*, the upper limit remains inside the finite clause.

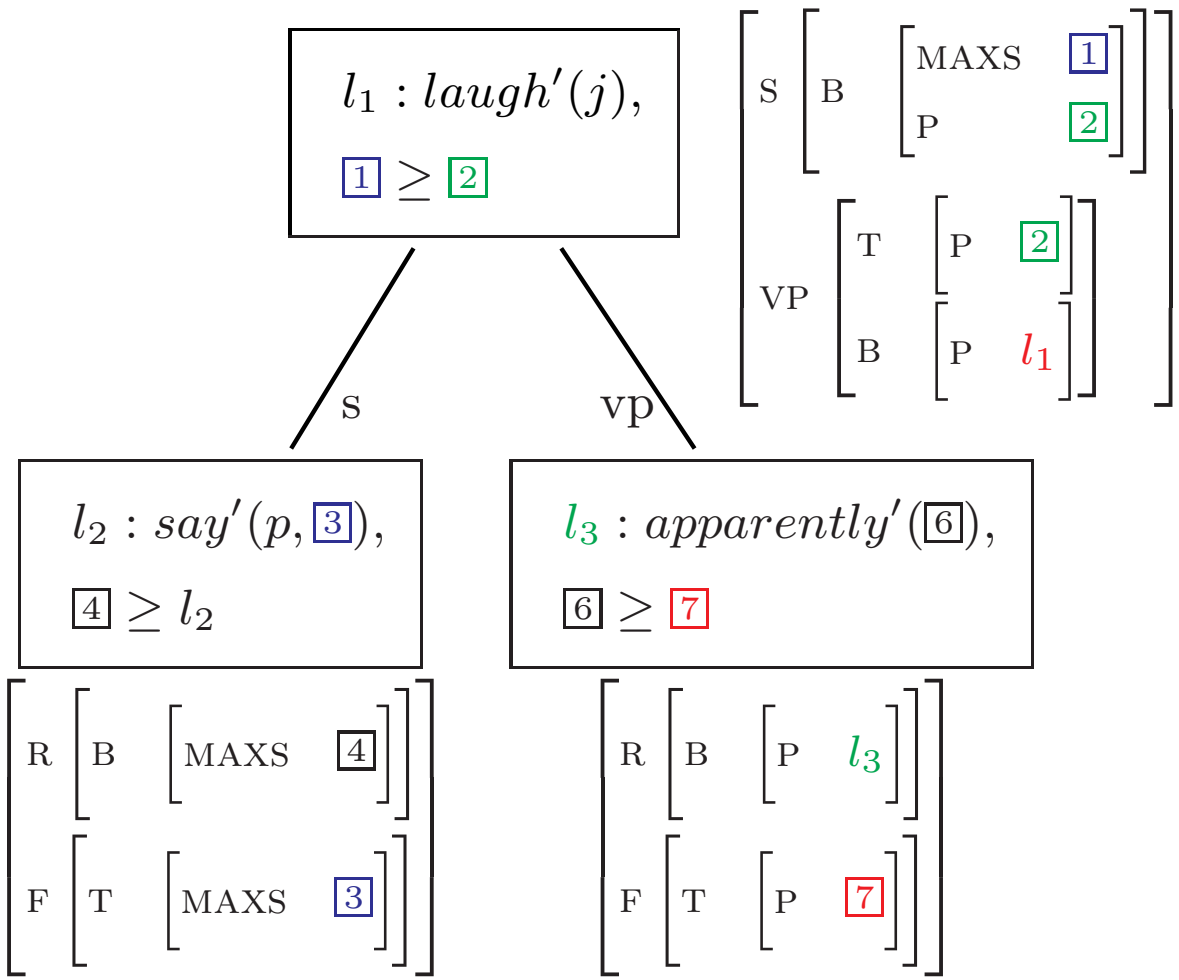
   2.2 When the predicate the NP combines is non-finite, ie. when the first operator up the tree is a control/ECM verb like *try*/*want*, the upper limit goes beyond the non-finite clause.

   2.3 When the NP is nested inside another NP, the upper limit goes immediately above the embedding NP.

Plan

1. Summary of LTAG syntax relevant for LTAG semantics
2. Semantic Composition in LTAG
3. Data on scope
4. **Analysis of the scope data**
5. Conclusions

Case 1: fix scope for VP-spine attachments



leads to scope order $\boxed{4} \geq l_2 > \boxed{1} \geq l_3 > \boxed{6} \geq l_1$, i.e. $say > apparently$.

Case 2: underspecified scope for NPs

Scope window: features MAXS and MINS.

NP
|
everybody

$$l_2 : every'(x, \boxed{4}, \boxed{5}),$$
$$l_3 : person'(x),$$
$$\boxed{4} \geq l_3,$$
$$\boxed{6} \geq l_2, \boxed{5} \geq \boxed{7}$$

$$\left[\text{NP}\ \left[\text{T}\ \left[\begin{array}{ll}\text{I} & \text{x} \\ \text{MAXS} & \boxed{6} \\ \text{MINS} & \boxed{7}\end{array}\right]\right]\right]$$

Case 2 (Cont.)

(10) Everyone laughs

$l_1 : laugh'(\boxed{1}),$
$\boxed{2} \geq \boxed{3}$

np

$l_2 : every'(x, \boxed{4}, \boxed{5}),$
$l_3 : person'(x),$
$\boxed{4} \geq l_3,$
$\boxed{6} \geq l_2,\ \boxed{5} \geq \boxed{7}$

$$\begin{bmatrix} \text{S} & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{MAXS} & \boxed{2} \\ \text{P} & \boxed{3} \end{bmatrix} \end{bmatrix} \\ \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{3} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_1 \end{bmatrix} \end{bmatrix} \\ \text{NP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{I} & \boxed{1} \\ \text{MAXS} & \boxed{2} \\ \text{MINS} & l_1 \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \text{NP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{I} & x \\ \text{MAXS} & \boxed{6} \\ \text{MINS} & \boxed{7} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Case 2 (Cont.)

Result:

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

Disambiguation: $\boxed{2} \rightarrow l_2,\ \boxed{4} \rightarrow l_3, \boxed{5} \rightarrow l_1$

This yields: $every'(x, person'(x), laugh'(x))$

(11) Exactly one student admires every professor

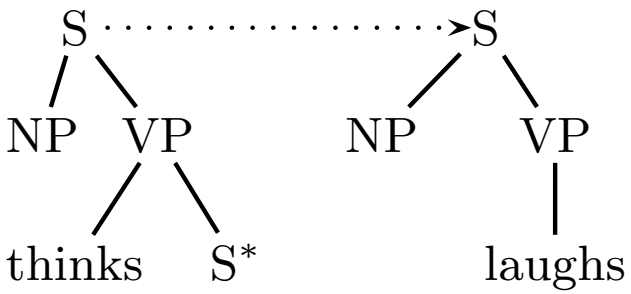
Result:

$$..., \boxed{2} \geq l_2(every), \boxed{5} \geq l_1$$
$$\boxed{2} \geq l_4(exactly - one), \boxed{8} \geq l_1$$

Ambiguous between $\exists > \forall, \forall > \exists$.

Case 2.1: NP in a finite clause

(12) Mary thinks John laughs



$l_1 : laugh'(j)$,

$\boxed{1} \geq \boxed{2}$

S

$l_2 : think'(m, \boxed{3})$,

$\boxed{4} \geq \boxed{5}$, $\boxed{3} \geq \boxed{6}$

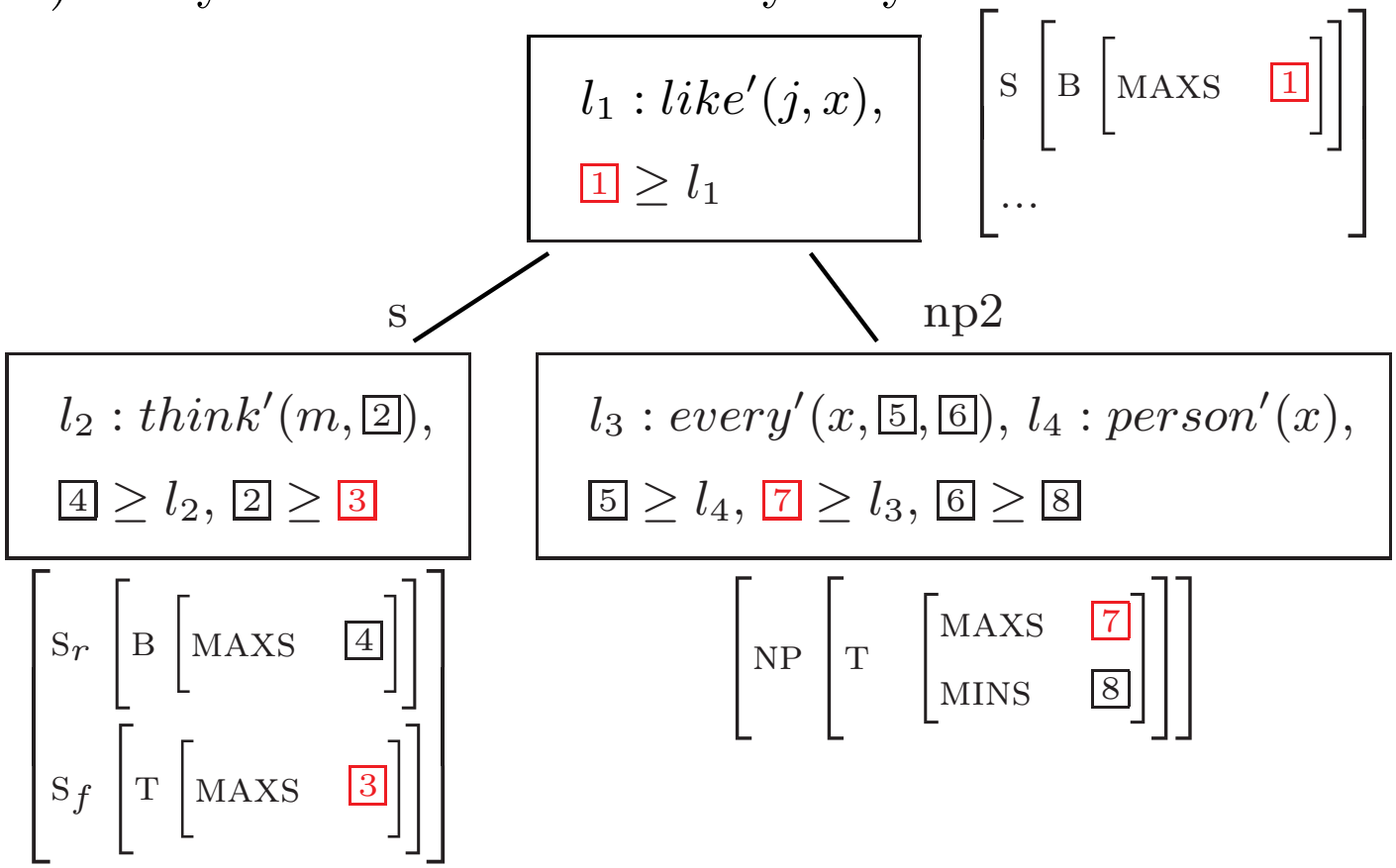
$$\begin{bmatrix} \text{S} & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{MAXS} & \boxed{1} \\ \text{P} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{2} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_1 \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \text{S}_r & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{MAXS} & \boxed{4} \\ \text{P} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{5} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_2 \end{bmatrix} \end{bmatrix} \\ \text{S}_f & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{MAXS} & \boxed{6} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Argument of attitude verb embeds MAXS of embedded verb

Case 2.1: NP in a finite clause (Cont.)

(13) Mary thinks John likes everybody

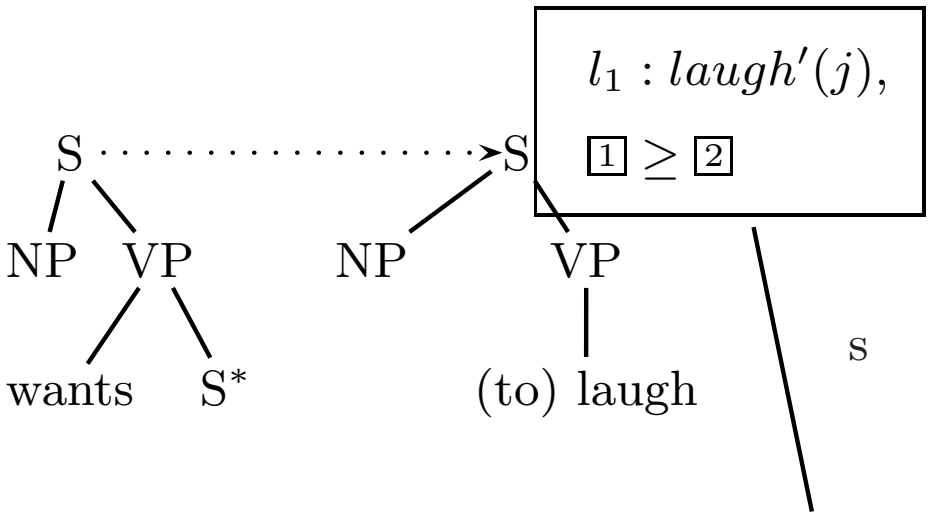


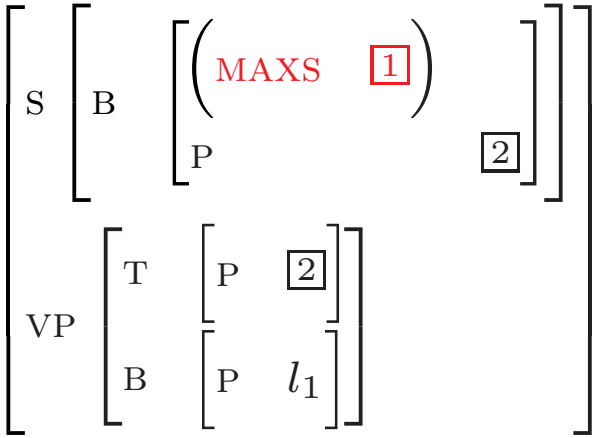
We have: $l_2 : think'(m, \boxed{2})$, $\boxed{2} \geq \boxed{3} = \boxed{1} = \boxed{7} \geq l_3$, $l_3 : every'(x, \boxed{5})$

Only one scope order: $think'(m, every'(x, person'(x), like'(j, x)))$

Case 2.2: NP in a non-finite clause

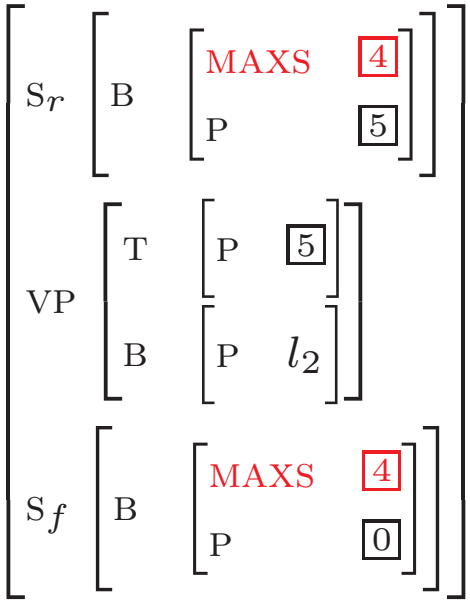
(14) John wants to laugh





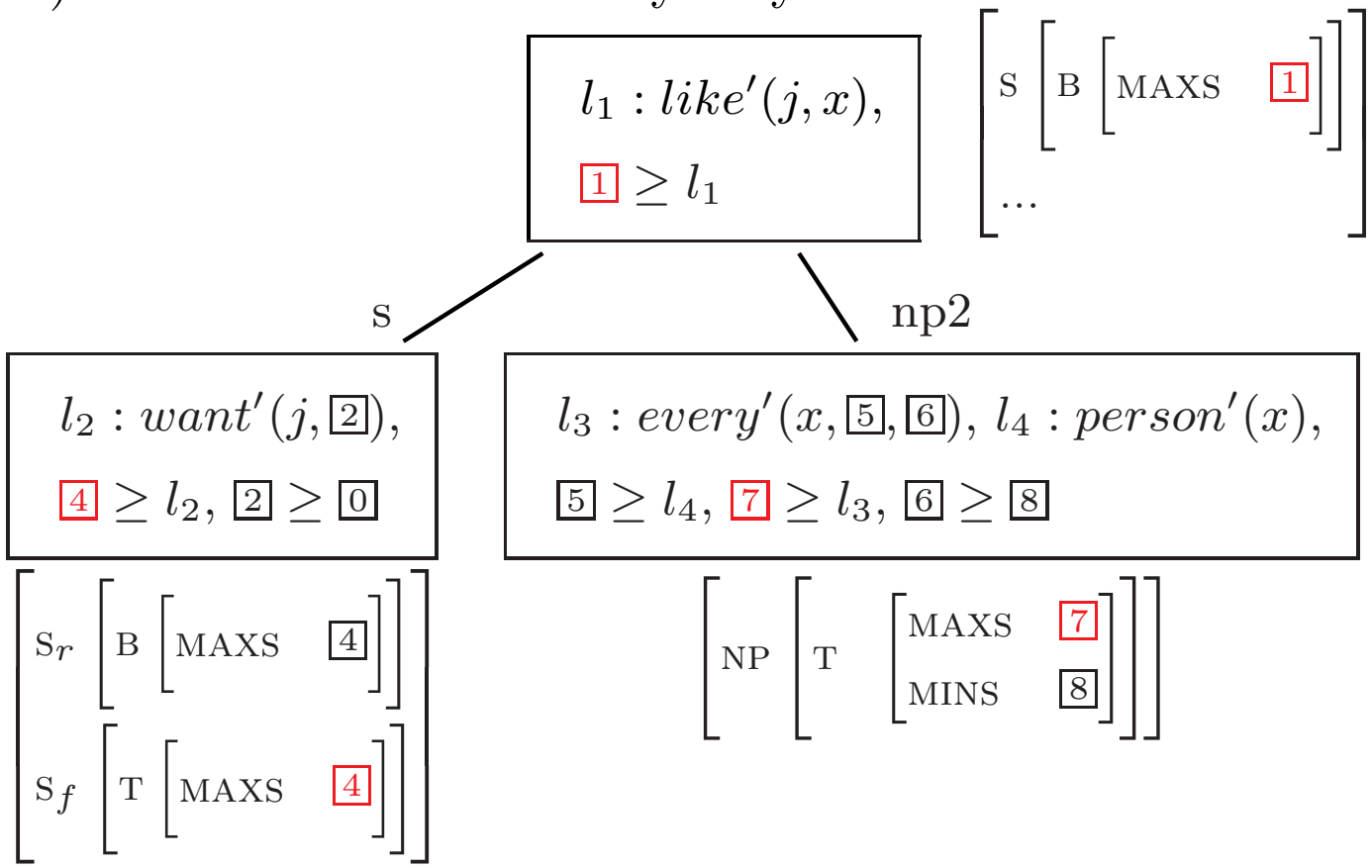
Argument of control/ECM verb lets MAXS of embedded verb pass

$l_2 : want'(j, \boxed{3})$,
$\boxed{4} \geq \boxed{5}$, $\boxed{3} \geq \boxed{0}$



Case 2.2: NP in a finite clause (Cont.)

(15) John wants to like everybody



We have: $\boxed{4} = \boxed{1} = \boxed{7}$ over $l_2 : want'(j, \boxed{2})$ and over $l_3 : every'(x, \boxed{5})$
Ambiguous between $want > every$ and $every > want$.

Case 2.3: NP nested inside NP

(16) Two policemen spy on someone from every city

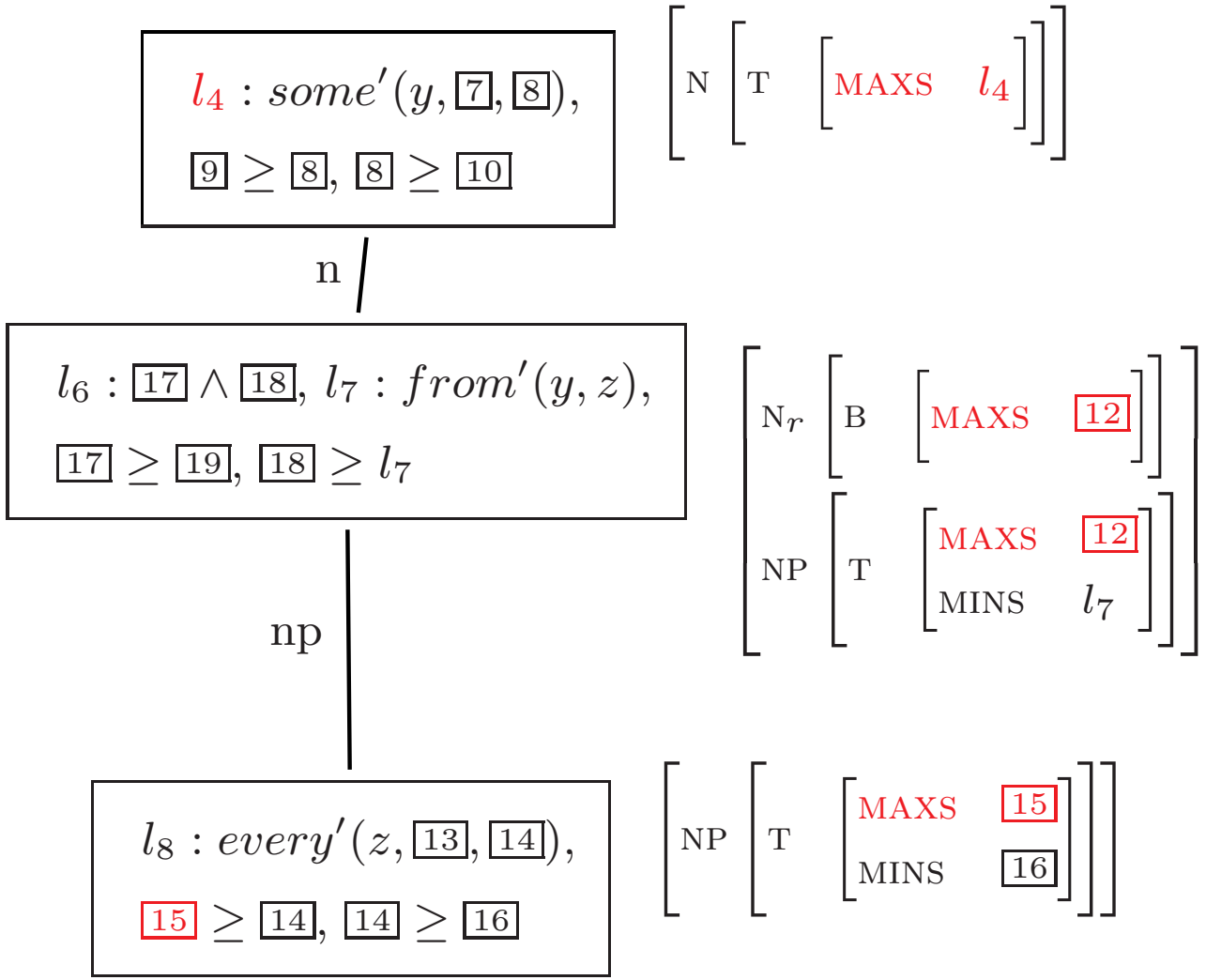
(17)
a. $2 > \exists > \forall$ b. $\exists > \forall$,2
c. $2 > \forall > \exists$ d. $\forall > \exists > 2$ e. $* \forall > 2 > \exists$

- Reading (e) needs to be ruled out: when the nested $\forall$ has scope over its host $\exists$, it must have **immediate** scope over it and thus the quantifier 2 is not allowed to intervene.

Case 2.3 (Cont.)

- Problem: how to express **"immediateness"** for the nested $\forall$? So far, we can only express:
  - Nuclear scope of $\exists$ $\geq$ MAXS of $\forall$. But readings d and e.
  - Nuclear scope of 2 $\geq$ MAXS of $\forall$. But readings b and e.
- Proposal: two changes in our formal lg of underspecified representations:
  1. So far, our scope constraints had the form $\boxed{n} \geq l$ or $\boxed{n} \geq \boxed{n}$. Now we'll also have $l \geq \boxed{n}$.
  2. Consider the quantifier sem representations $l_1 : Qu_1(x, \boxed{1}, \boxed{2})$ and $l_2 : Qu_2(x, \boxed{3}, \boxed{4})$. So far, to express that Qu_1 must scope over Qu_2, we used the constraint $\boxed{2} \geq l_2$. Now we'll express it using the constraint: $\boxed{2} \geq \boxed{4}$.

(18) someone from every city



Case 2.3 (Cont.)

(19) Two policemen spy on someone from every city

(20)
a. $2 > \exists > \forall$ b. $\exists > \forall$,2
c. $2 > \forall > \exists$ d. $\forall > \exists > 2$ e. $* \forall > 2 > \exists$

Result:

- $\boxed{0} \geq$ nuclear scope of 2 $\geq l_1 : spy(x, y)$
  $\boxed{0} \geq$ nuclear scope of *some* $\geq l_1 : spy(x, y)$
- $l_4(some) \geq \boxed{14}$, i.e. nuclear scope of *every* $\geq l_7(from)$
  $\Rightarrow$ The nuclear scope of the nested *every* must be under or equal to $l_4 : some'(y, \boxed{7}, \boxed{8})$. This means the nested *every* must scope under *some* or immediately above *some*.

Conclusions

Using feature unification in a way parallel to the syntax of TAG, and taking the derivation tree as sole syntactic input, we have captured the scopal behavior of different quantificational elements:

1. Attachments to VP-spine have fix surface scope
2. Quantificational NPs have underspecified scope within a scope window. The upper limit of the scope window is determined by the operators up the tree:
   - 2.1 *Say/think*: take as their complement the MAXS of embedded clause, thus stopping that MAXS.
   - 2.2 Control/ECM verbs and prepositions (e.g *from*) do not take the embedded MAXS as their complement but pass it up by coindexation.
   - 2.3 A host quantificational NP equates MAXS from the embedded predicate to its own formula.