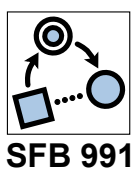
Grammar Implementation with Lexicalized Tree Adjoining Grammars and Frame Semantics

Further linguistic analyses

Laura Kallmeyer, Timm Lichte, Rainer Osswald & Simon Petitjean

University of Düsseldorf

DGfS CL Fall School, September 13, 2017



Outline of today's course

1. Extraction phenomena in LTAG
2. Generalization and factorization within the elementary trees
   - Tree families
   - LTAG & metagrammar specification
3. LTAG semantics
   - Synchronous TAGs for semantics
   - Unification-based LTAG semantics with predicate logic
   - Unification-based LTAG semantics with frames
4. Summary & outlook

Outline of today's course

1. Extraction phenomena in LTAG
2. Generalization and factorization within the elementary trees
   - Tree families
   - LTAG & metagrammar specification
3. LTAG semantics
   - Synchronous TAGs for semantics
   - Unification-based LTAG semantics with predicate logic
   - Unification-based LTAG semantics with frames
4. Summary & outlook

Extraction: some examples

- certain constructions permit an element in one position to fill the grammatical role associated with another position

Extraction: some examples

- certain constructions permit an element in one position to fill the grammatical role associated with another position
- the positions can be arbitrarily far apart

Extraction: some examples

- certain constructions permit an element in one position to fill the grammatical role associated with another position
- the positions can be arbitrarily far apart
- filler – gap constructions
  - topicalization
  - wh-movement
  - relative clause

Extraction: some examples

- certain constructions permit an element in one position to fill the grammatical role associated with another position
- the positions can be arbitrarily far apart
- filler – gap constructions
  - topicalization
  - wh-movement
  - relative clause
- long-distance dependencies
  - subject extraction
  - object extraction
  - preposition stranding
  - AP complement extraction

Topicalization

Topicalization

Placing a constituent (subject, object, ...) into a sentence-initial position.

Topicalization

Topicalization

Placing a constituent (subject, object, ...) into a sentence-initial position.

(1)	a.	Adam gave an apple to Eve.	(base configuration)
	b.	an apple$_i$, Adam gave _$_i$ to Eve.	(object NP)
	c.	Eve$_i$, Adam gave an apple to _$_i$.	(NP from PP)
	d.	To Eve$_i$, Adam gave an apple _$_i$.	(PP)
	e.	*Adam, _$_i$ gave an apple to Eve.	(no subject topicalization!)

Topicalization

Unbounded dependency

The dependency between an extracted constituent and its trace may extend across more *clause boundaries*.

Topicalization

Unbounded dependency

The dependency between an extracted constituent and its trace may extend across more *clause boundaries*.

(2) a. The apple$_i$, Adam ate _$_i$.
b. Apples$_i$, Eve knows (that) Adam loves _$_i$.
c. The apple$_i$, Adam believes (that) Eve knows (that) the snake ate _$_i$.

Wh-constructions

Wh-questions

wh-questions involve a (possibly long-distance) extraction of a constituent as a **wh-phrase**.

Wh-constructions

Wh-questions

wh-questions involve a (possibly long-distance) extraction of a constituent as a **wh-phrase**.

(3) a. [Who]$_i$ _$_i$ ate my apple?
b. [What]$_i$ did Eve eat _$_i$?
c. [Which apple]$_i$ did Adam say Eve had eaten _$_i$?

Wh-constructions

Wh-questions

wh-questions involve a (possibly long-distance) extraction of a constituent as a **wh-phrase**.

(3) a. [Who]$_i$ _$_i$ ate my apple?
b. [What]$_i$ did Eve eat _$_i$?
c. [Which apple]$_i$ did Adam say Eve had eaten _$_i$?

Subject-auxiliary inversion

wh-questions involve **subject-auxiliary inversion**: The auxiliary verb (‘do’, ‘have’, ‘be’, …) precedes the subject.

Subject-auxiliary inversion

- **Obligatory subject-auxiliary inversion** in direct questions with object extraction:

(4) a. What$_i$ **does** Adam eat _$_i$?
b. *What$_i$ Adam **does** eat _$_i$?
c. *What$_i$ Adam eats _$_i$?

Subject-auxiliary inversion

- **Obligatory subject-auxiliary inversion** in direct questions with object extraction:

  (4) a. What$_i$ **does** Adam eat _$_i$?
  b. *What$_i$ Adam **does** eat _$_i$?
  c. *What$_i$ Adam eats _$_i$?

- **No subject-auxiliary inversion** in embedded wh-questions:

  (5) a. Eve wonders [what$_i$ Adam eats _$_i$].
  b. *Eve wonders [what$_i$ **does** Adam eat _$_i$].

Subject-auxiliary inversion

- **Obligatory subject-auxiliary inversion** in direct questions with object extraction:

  (4) a. What$_i$ **does** Adam eat _$_i$?
  b. *What$_i$ Adam **does** eat _$_i$?
  c. *What$_i$ Adam eats _$_i$?

- **No subject-auxiliary inversion** in embedded wh-questions:

  (5) a. Eve wonders [what$_i$ Adam eats _$_i$].
  b. *Eve wonders [what$_i$ **does** Adam eat _$_i$].

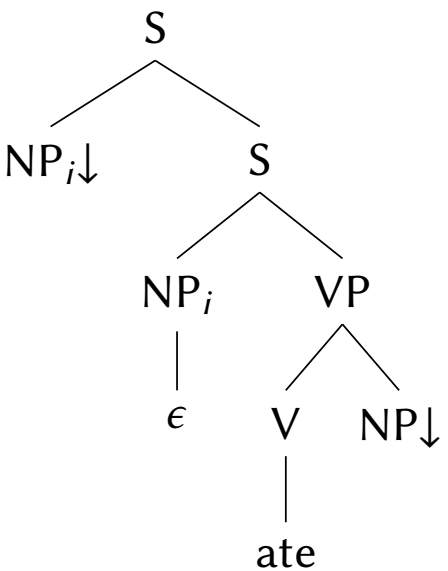
- **No subject-auxiliary inversion** in topicalization:

  (6) a. *[The apple]$_i$, **has** Adam eaten _$_i$.
  b. [The apple]$_i$ Adam **has** eaten _$_i$.

Extraction: elementary trees

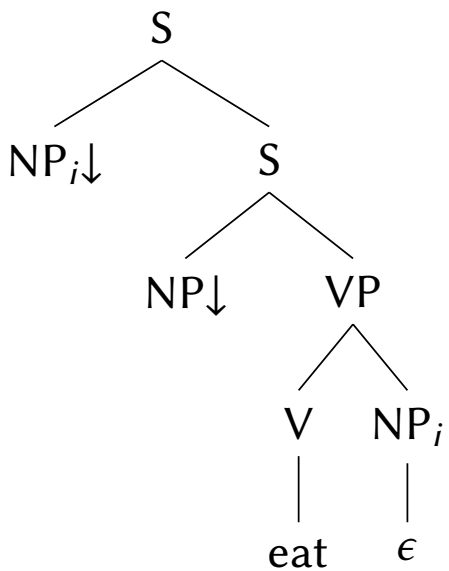
subject extraction

Who$_i$ _$_i$ ate the apple?



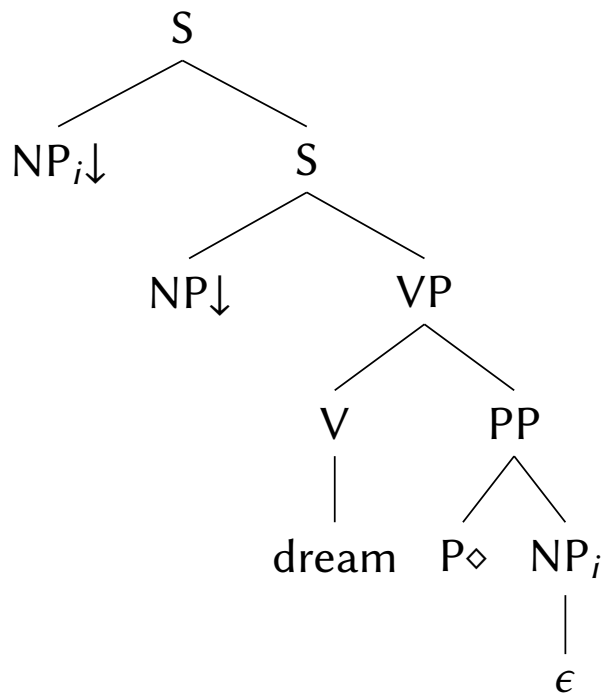
object extraction

What$_i$ did Adam eat _$_i$?



preposition stranding

What$_i$ does Adam dream of_$_i$?



Extraction: features

Features for extraction, taken from the XTAG grammar (XTAG Research Group 2001)

- EXTRACTED := + | − indicates extraction in the S-node
- WH := + | − indicates the presence of a wh-pronoun
- INV := + | − indicates inversion

Extraction: features

Features for extraction, taken from the XTAG grammar (XTAG Research Group 2001)

- EXTRACTED := + | − indicates extraction in the S-node
- WH := + | − indicates the presence of a wh-pronoun
- INV := + | − indicates inversion

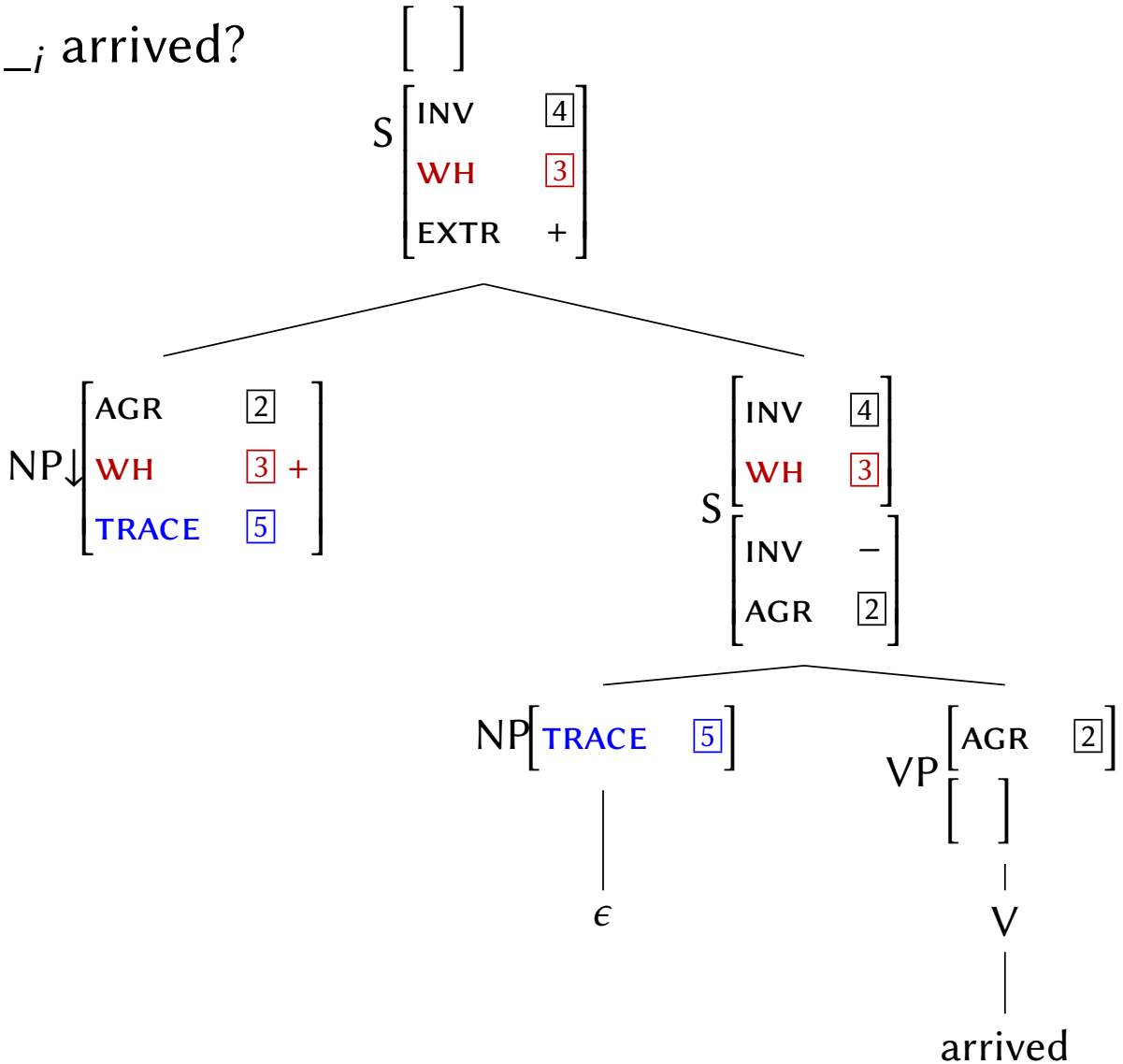
Handling:

- no inversion with topicalization (*Books$_i$, people read _$_i$.*)
- no topicalized subject (**People$_i$, _$_i$ read books.*)
- no inversion with subject wh-extraction (*Who$_i$ _$_i$ read books?*)
- inversion with object wh-extraction (*What$_i$ do people read _$_i$?*)

Extraction: elementary trees with features

Elementary tree for subject extraction:

(7) Who$_i$ _$_i$ arrived?

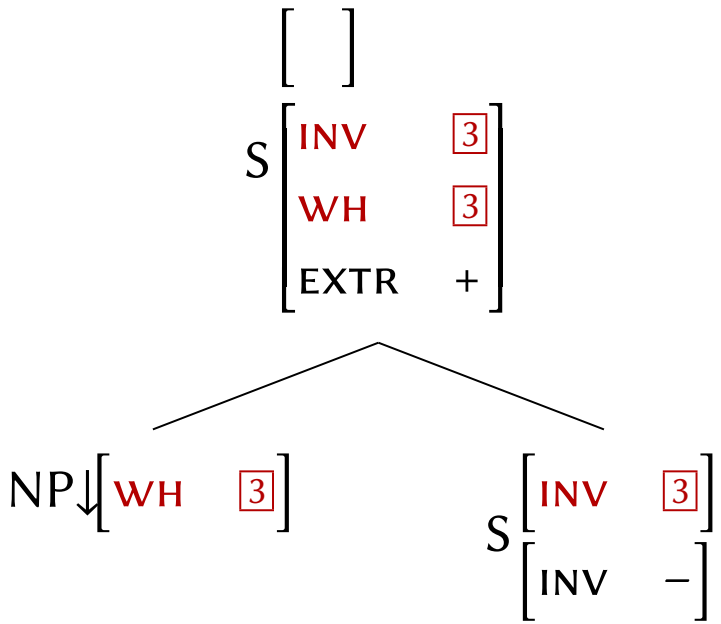


Inversion with object extraction

- in case of object extraction
  - topicalization → no inversion
  - wh-questions → inversion

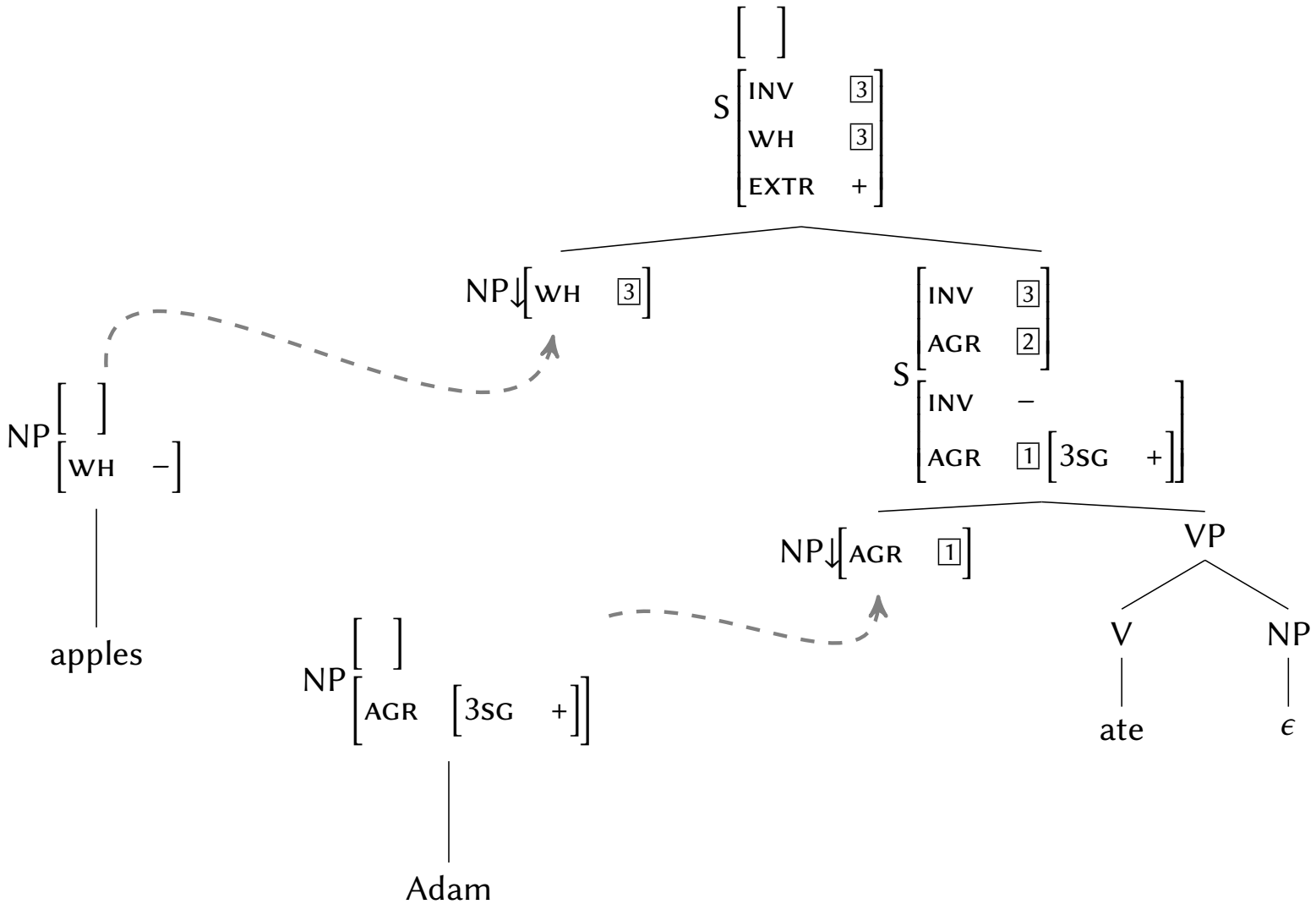
Inversion with object extraction

- in case of object extraction
  - topicalization → no inversion
  - wh-questions → inversion
- ⇒ equation of the WH feature of the extracted NP and the upper INV feature of the lower S node:



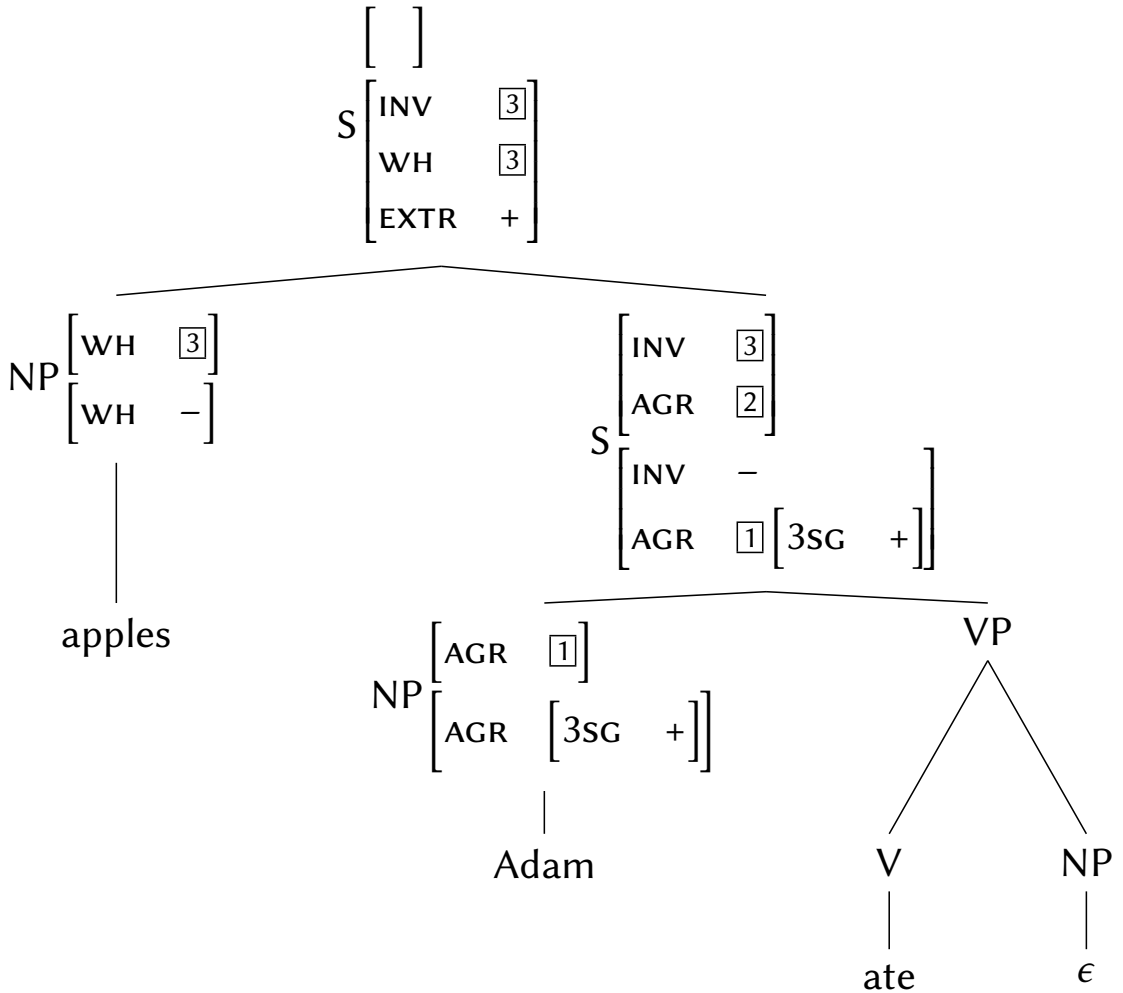
Analyses

(8) Apples, Adam ate.



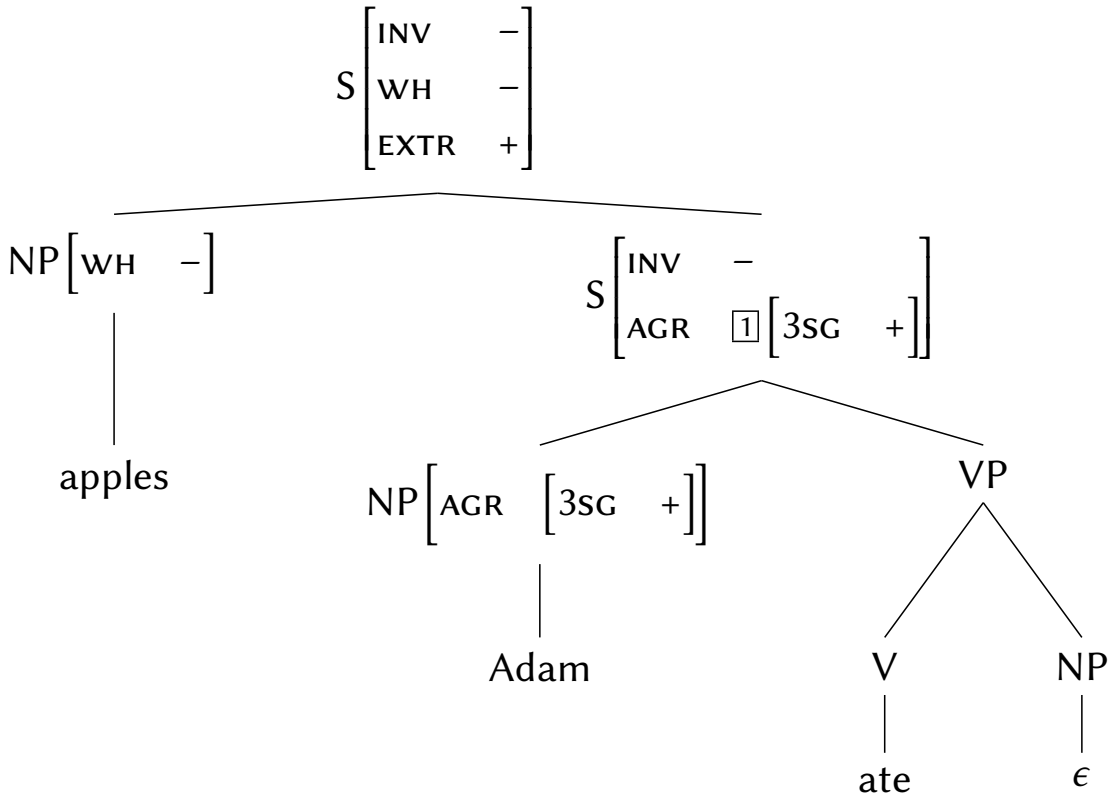
Analyses

Derived tree with top and bottom feature structures:



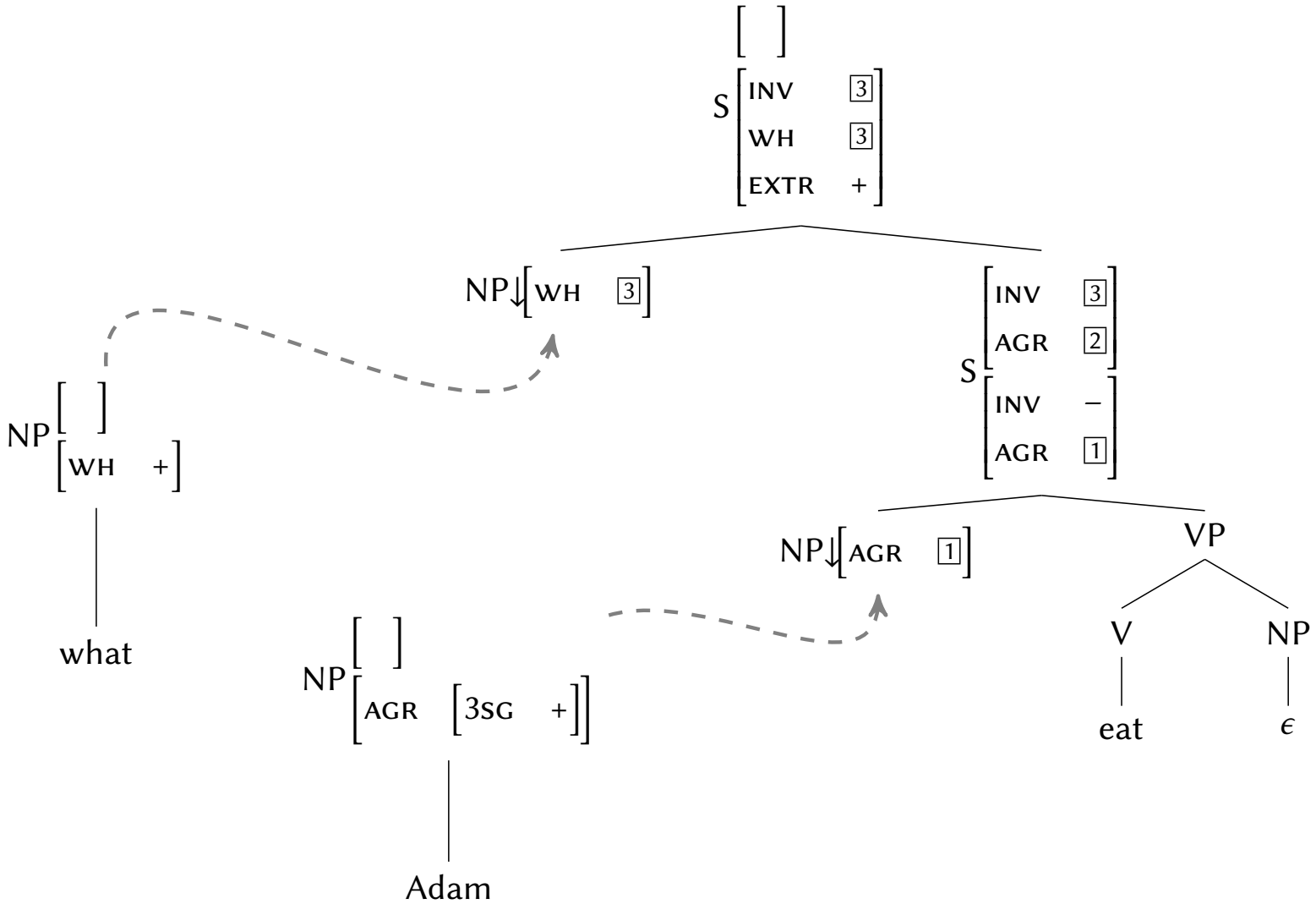
Analyses

Final derived tree after top-bottom unification:

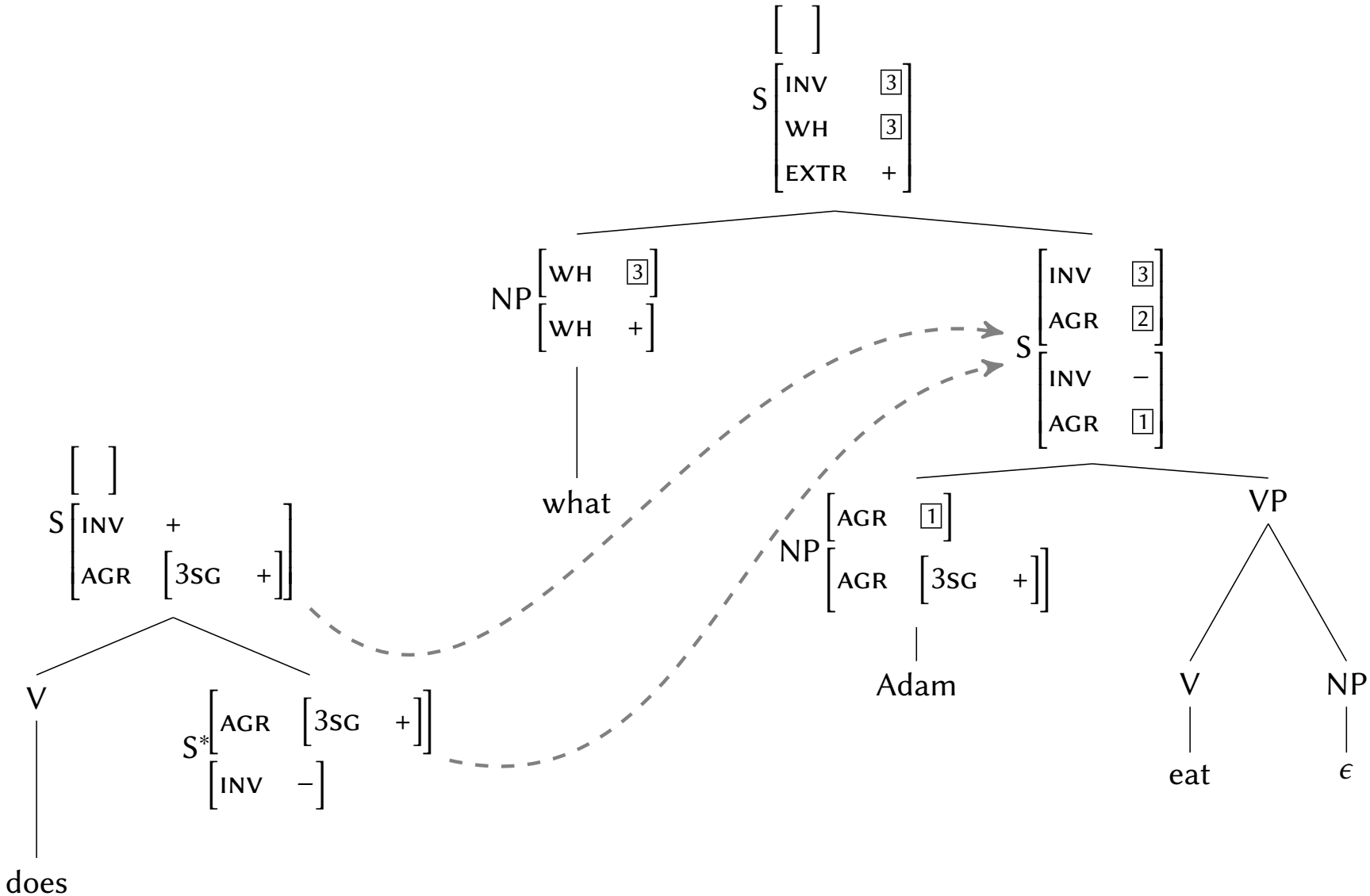


Analyses

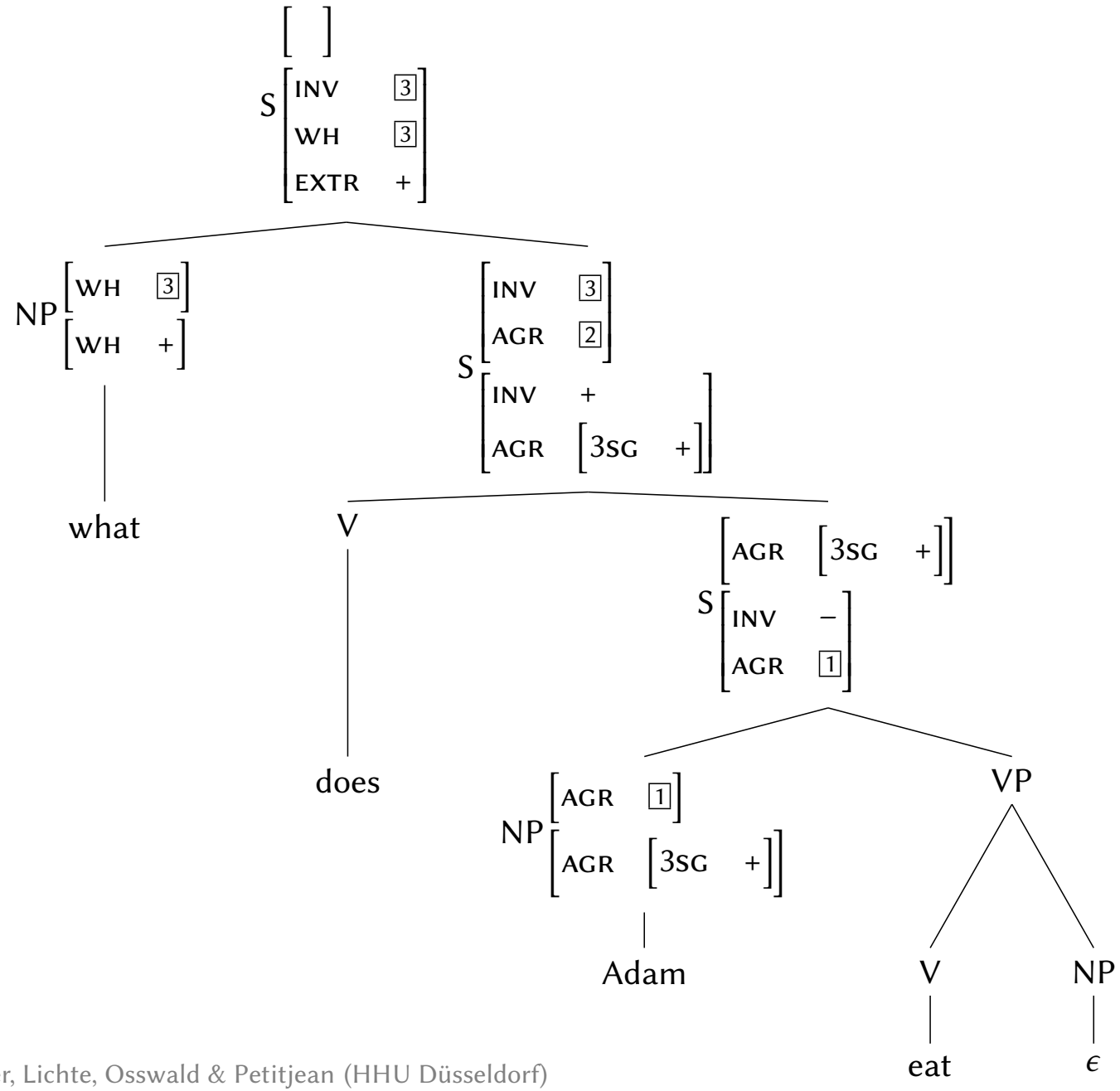
(9) What does Adam eat?



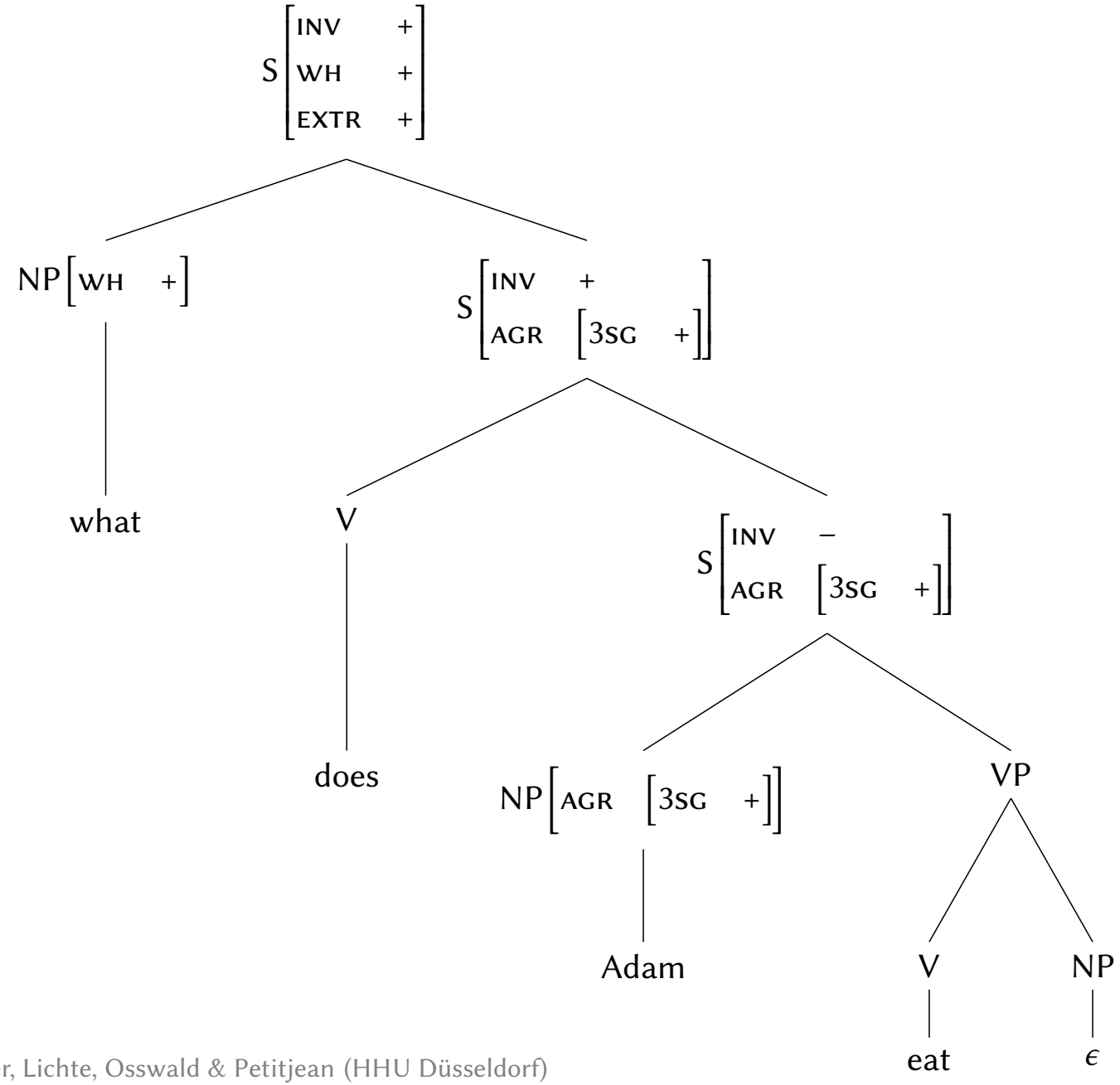
Analyses


[ ]
S [INV 3] [WH 3] [EXTR +]
NP [WH 3] [WH +]
what
S [INV 3] [AGR 2] [INV −] [AGR 1]
[ ]
S [INV +] [AGR [3SG +]]
NP [AGR 1] [AGR [3SG +]]
VP
V
S* [AGR [3SG +]] [INV −]
Adam
V
NP
does
eat
ϵ

Analyses


[ ]
S [INV 3, WH 3, EXTR +]
NP [WH 3, WH +]
what
S [INV 3, AGR 2]
[INV +, AGR [3SG +]]
V
does
[AGR [3SG +]]
S [INV −, AGR 1]
NP [AGR 1, AGR [3SG +]]
Adam
VP
V
eat
NP
ϵ

Analyses


S [INV +, WH +, EXTR +]
NP [WH +]
what
S [INV +, AGR [3SG +]]
V
does
S [INV −, AGR [3SG +]]
NP [AGR [3SG +]]
Adam
VP
V
eat
NP
ϵ

Outline of today's course

1. Extraction phenomena in LTAG
2. Generalization and factorization within the elementary trees
   - Tree families
   - LTAG & metagrammar specification
3. LTAG semantics
   - Synchronous TAGs for semantics
   - Unification-based LTAG semantics with predicate logic
   - Unification-based LTAG semantics with frames
4. Summary & outlook

Tree families

Lexical anchoring

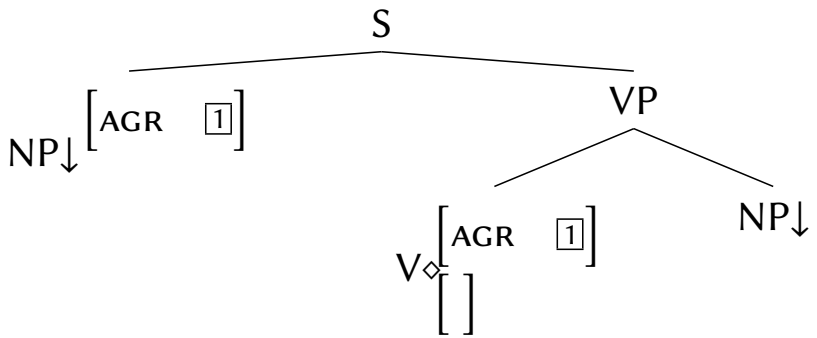
Unanchored trees and their lexical anchors are specified separately.
⇒ the contribution of the lexical anchor is factored out.

Tree families

Lexical anchoring

Unanchored trees and their lexical anchors are specified separately.
⇒ the contribution of the lexical anchor is factored out.

Lexical anchoring example:

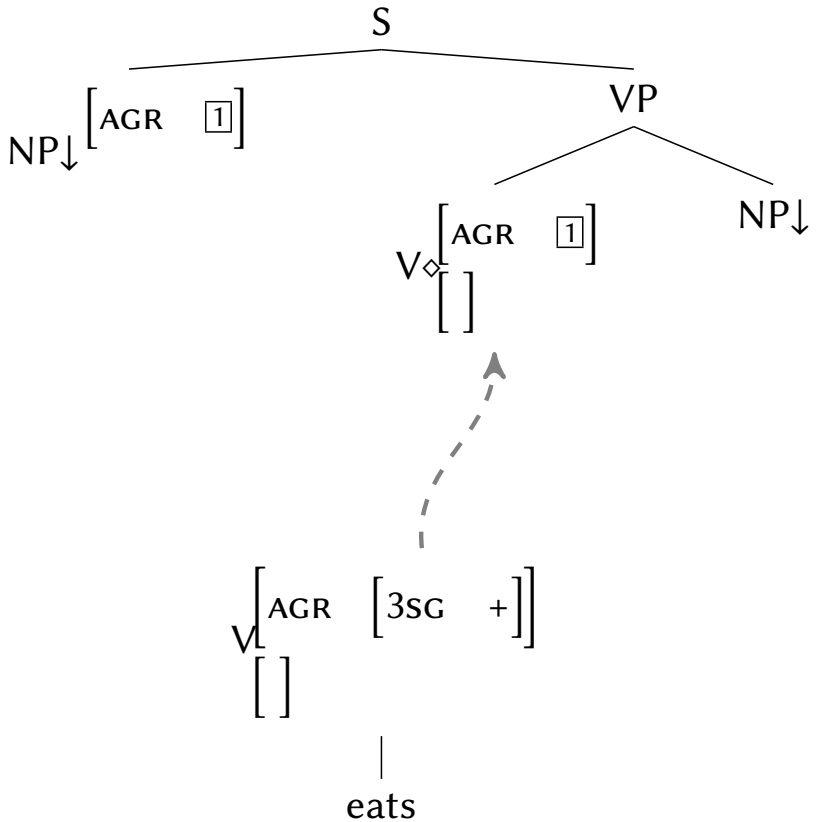


Tree families

Lexical anchoring

Unanchored trees and their lexical anchors are specified separately.
⇒ the contribution of the lexical anchor is factored out.

Lexical anchoring example:

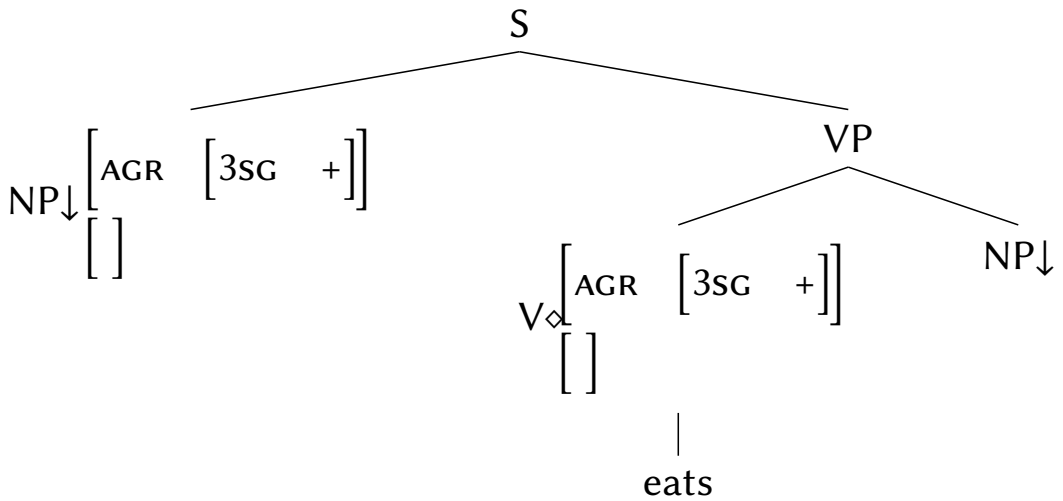


Tree families

Lexical anchoring

Unanchored trees and their lexical anchors are specified separately.
⇒ the contribution of the lexical anchor is factored out.

Lexical anchoring example:



Tree families

Unanchored elementary trees are organized in tree families, which capture variations in the (syntactic) subcategorization frames.

Tree families

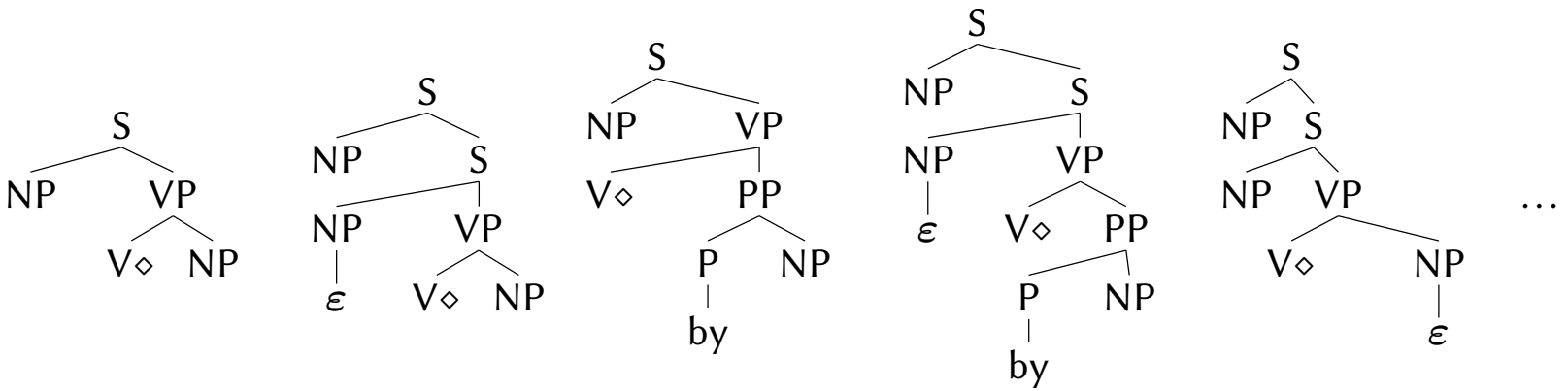
Unanchored elementary trees are organized in tree families, which capture variations in the (syntactic) subcategorization frames.

Example:

(10) a. Adam eats apples.
b. Who eats apples?
c. Apples are eaten by Adam.
d. What was eaten by Adam?
e. Apples, Adam eats.
f. ...

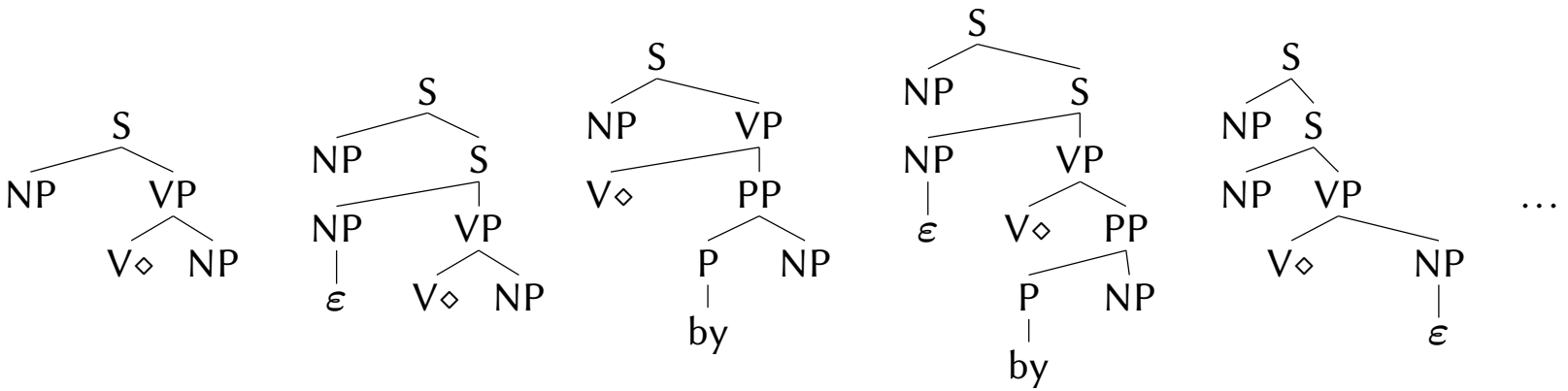
Tree families

Unanchored tree family for transitive verbs:



Tree families

Unanchored tree family for transitive verbs:



Options for the specification/generation of tree families:

- Transformation rules applied to base trees (e.g., metarules in XTAG)
- Classes of tree constraints (“metagrammar”, XMG system)

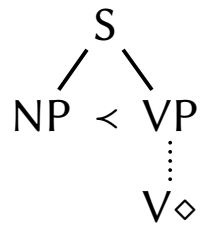
LTAG & metagrammar specification

eXtensible MetaGrammar (XMG, Crabbé et al. 2013)

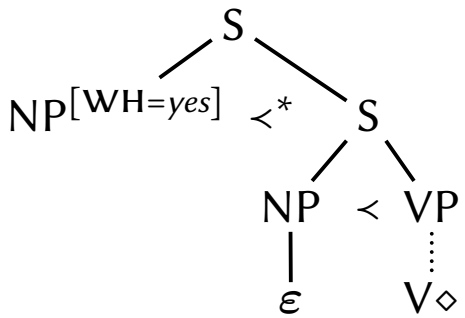
- A framework for specifying (the elementary structures of) tree based grammars by means of a **declarative** language (e.g., by dominance and precedence constraints)
- The specifications are organized into **classes** that can be **reused** ("imported") by other classes.
- Classes may contain descriptions from different **dimensions**, and the XMG system can be extended in this respect, e.g., by a dimension of **frame** descriptions.
- An XMG **compiler** generates the elementary structures of a grammar from a metagrammar.

LTAG & metagrammar specification: Example

Class *CanSubj*



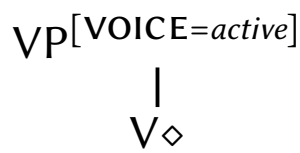
Class *ExtrSubj*



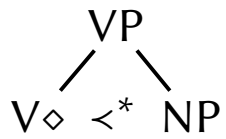
Class *Subj*

$CanSubj \vee ExtSubj$

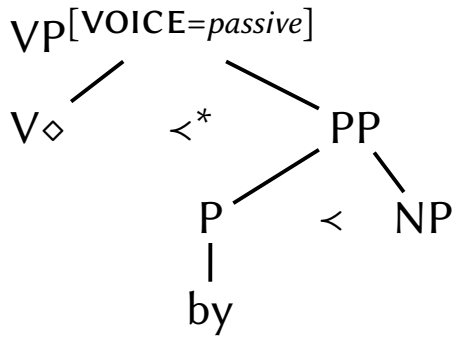
Class *ActV*



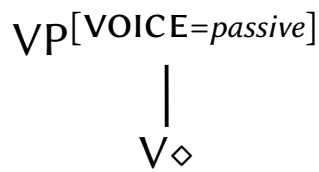
Class *DirObj*



Class *ByObj*



Class *PassV*

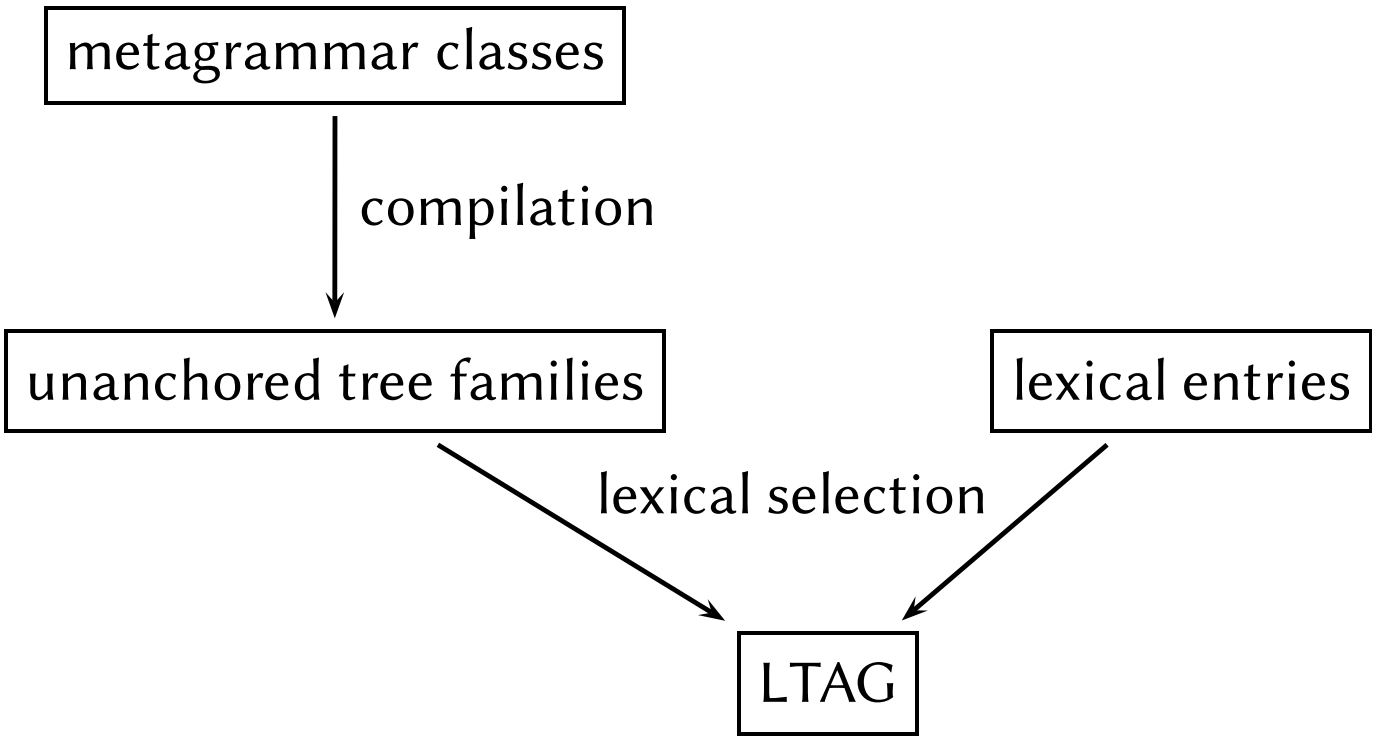


Class *Transitive*

$((Subj \wedge ActV) \vee ByObj \vee PassV) \wedge ((DirObj \wedge ActV) \vee (Subj \wedge PassV))$

LTAG & metagrammar specification

Overall architecture



Next step:

Add (frame) semantics to all components and link syntax to semantics.

Outline of today's course

1. Extraction phenomena in LTAG
2. Generalization and factorization within the elementary trees
   - Tree families
   - LTAG & metagrammar specification
3. LTAG semantics
   - Synchronous TAGs for semantics
   - Unification-based LTAG semantics with predicate logic
   - Unification-based LTAG semantics with frames
4. Summary & outlook

LTAG semantics: overview

Goal: an LTAG architecture of the syntax-semantics interface that

- is compositional: the meaning of a complex expression can be computed from the meaning of its subparts and its composition operation.
- pairs entire elementary trees with meaning components.

LTAG semantics: overview

Three principal approaches:

1. LTAG semantics with synchronous TAG (STAG)
   (Shieber 1994; Nesson & Shieber 2006; 2008)
2. Unification based LTAG semantics with predicate logic
   (Kallmeyer & Joshi 2003; Gardent & Kallmeyer 2003; Kallmeyer & Romero 2008)
3. Unification based LTAG semantics with frames
   (Kallmeyer & Osswald 2013; Kallmeyer et al. 2016)

We will use the third approach in this course and only briefly present the other two.

LTAG semantics: STAG

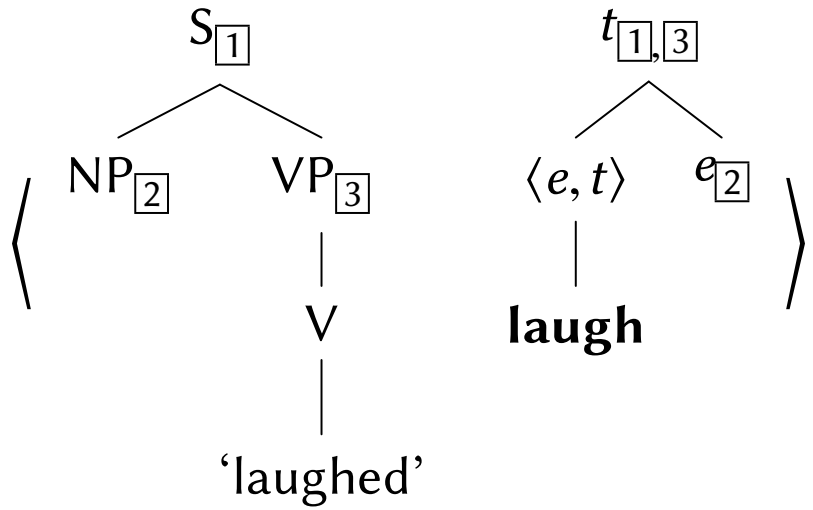
Idea:

- pair two TAGs, one for syntax and one for L(ogical) F(orm) (= typed predicate logic),
- and do derivations in parallel.

Formalism used for this: *synchronous TAG (STAG)* Shieber & Schabes (1990); Shieber (1994).

STAG = two TAGs G_1, G_2 whose trees are related to each other. More precisely, it contains pairs $\langle \gamma_1, \gamma_2, \mathit{link} \rangle$ where γ_1 is an elementary tree from G_1, γ_2 an elementary tree from G_2, and *link* is a set of pairs of node addresses from γ_1 and γ_2 respectively.

LTAG semantics: STAG



(The links are depicted with boxed numbers.)

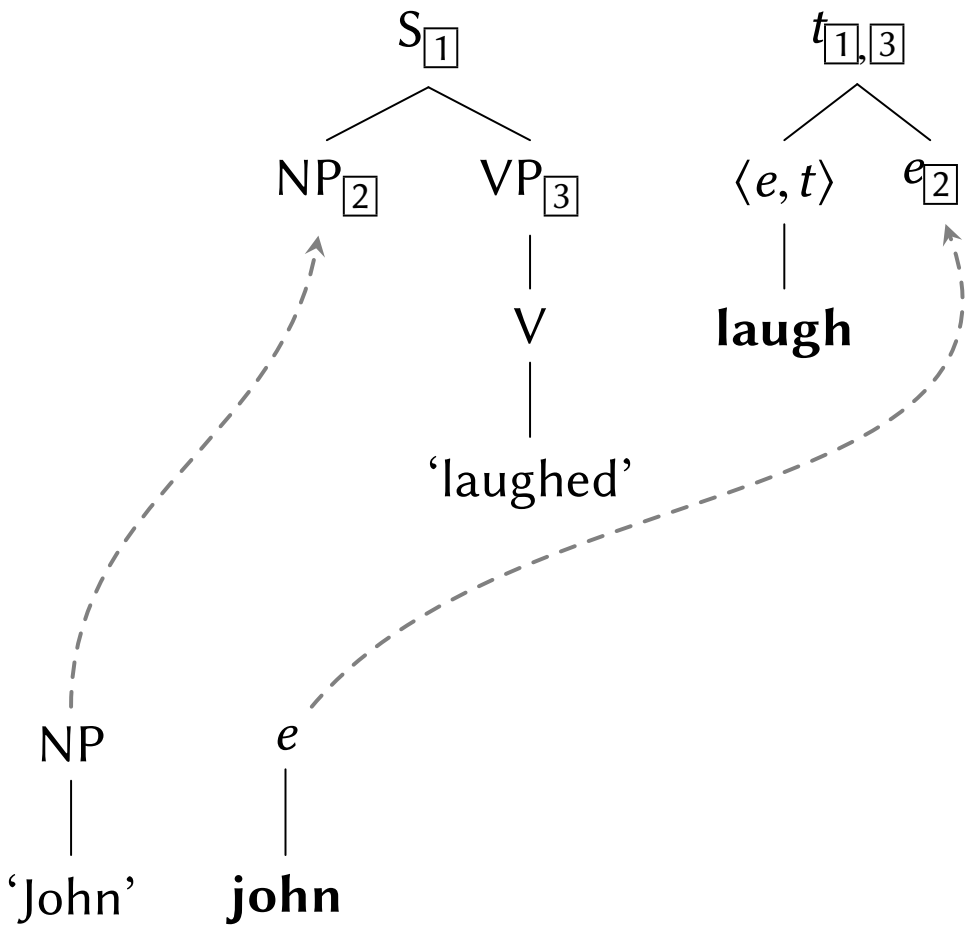
- The non-terminals of the semantic TAG are types $t, e, \langle e, t \rangle, \ldots$.
- The semantic TAG describes the syntactic structure of typed predicate logical formulas.
- The links in this example tell us, for instance, that the subject NP corresponds to the e argument of **laugh**.

LTAG semantics: STAG

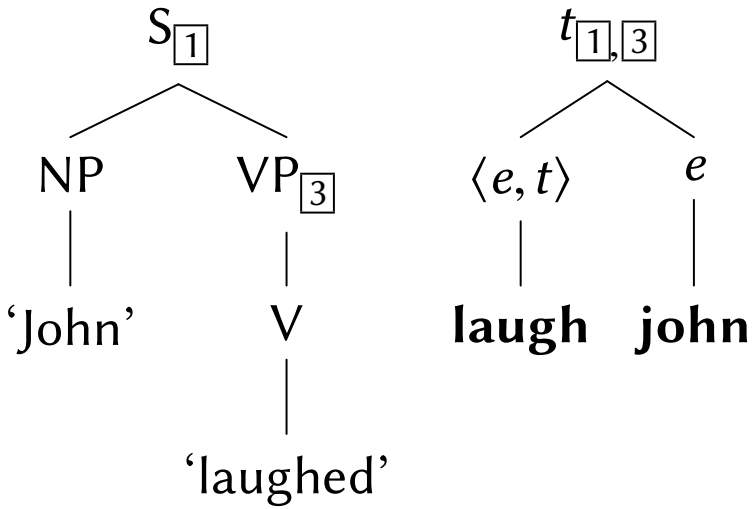
STAG derivation proceeds as in TAG, except that all operations must be paired: In every derivation step:

- A new elementary tree pair $\langle \gamma_1, \gamma_2 \rangle$ is picked.
- γ_1 is attached (substituted or adjoined) to the syntactic tree while γ_2 is attached to the semantic tree.
- The nodes that the two trees attach to must be linked.
- The link that is used in this derivation step disappears while all other links involving the attachment sites are inherited by the root of the attaching tree.

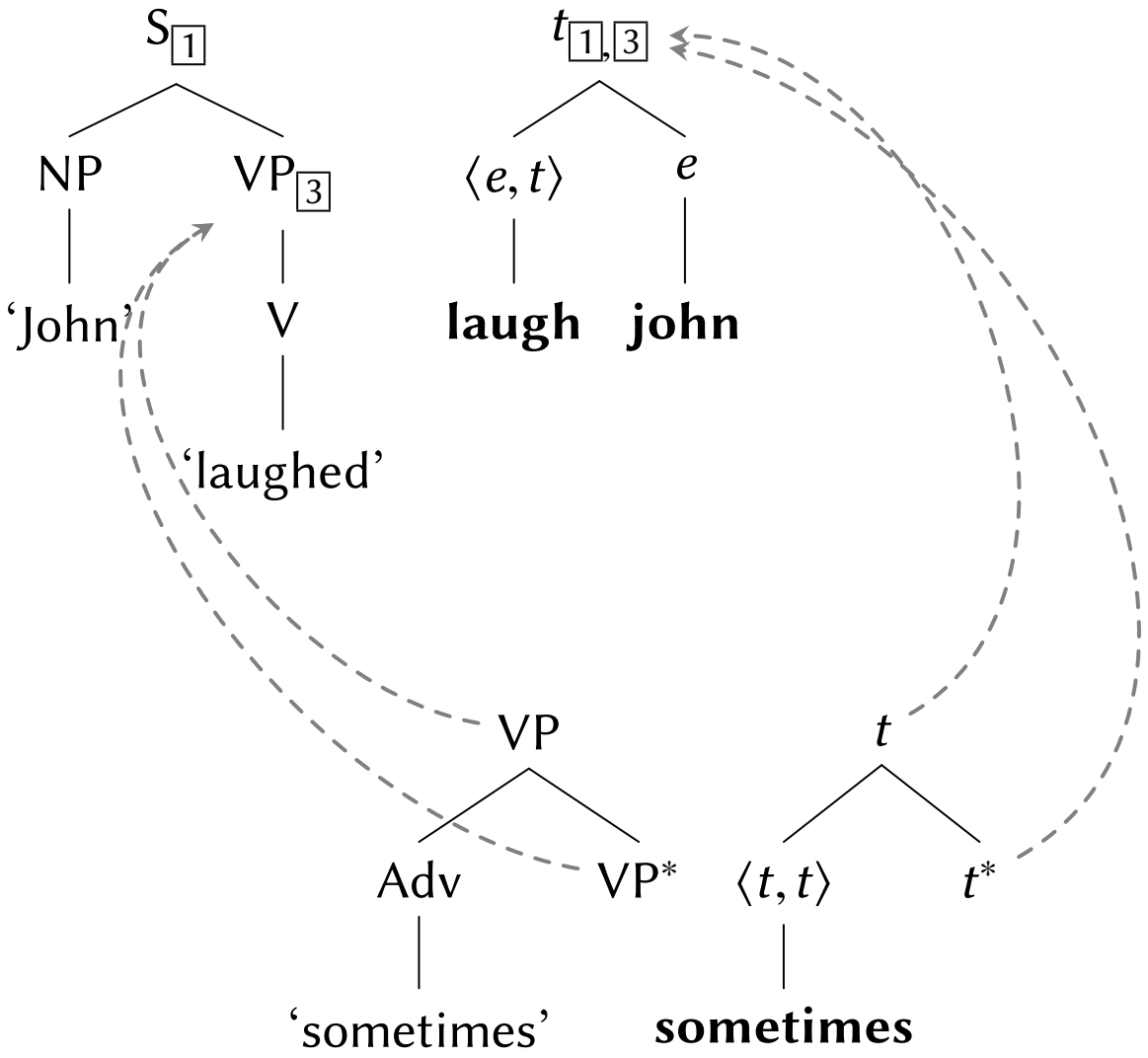
LTAG semantics: STAG



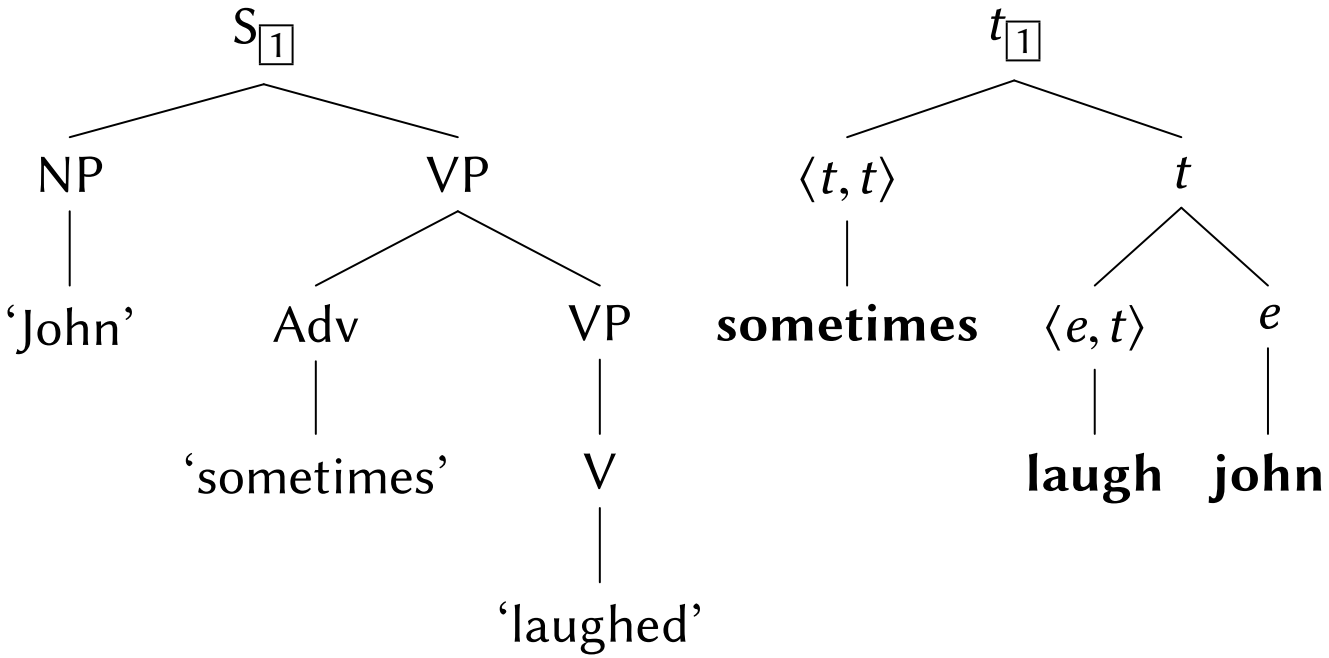
LTAG semantics: STAG



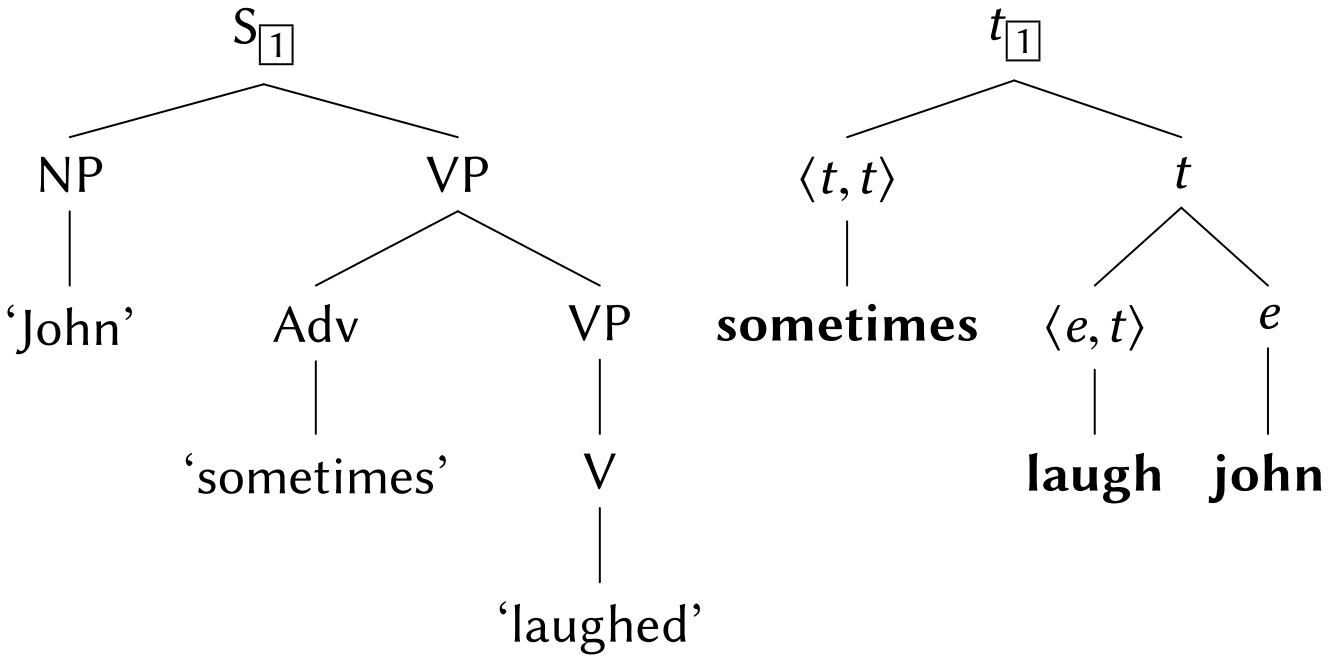
LTAG semantics: STAG



LTAG semantics: STAG



LTAG semantics: STAG



Logical form: **sometimes**(**laugh**(**john**))

Unification-based LTAG semantics with predicate logic

Kallmeyer & Romero (2008), Gardent & Kallmeyer (2003):
Syntax-Semantics Interface for LTAG

Idea: Each elementary tree is paired with

Unification-based LTAG semantics with predicate logic

Kallmeyer & Romero (2008), Gardent & 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)

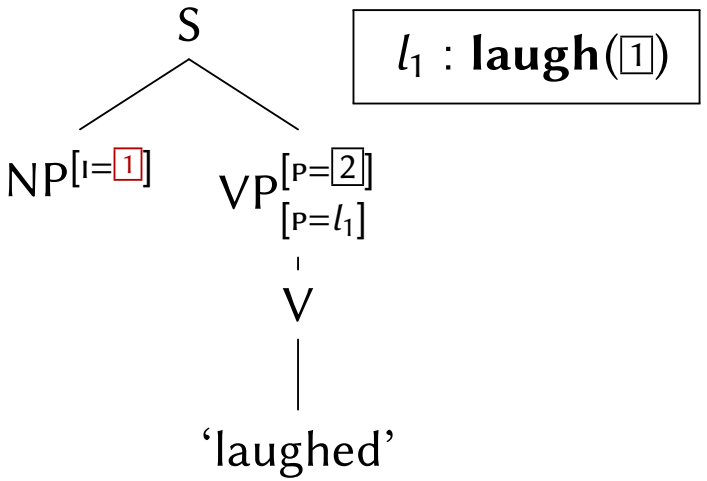
Unification-based LTAG semantics with predicate logic

Kallmeyer & Romero (2008), Gardent & Kallmeyer (2003):
Syntax-Semantics Interface for LTAG

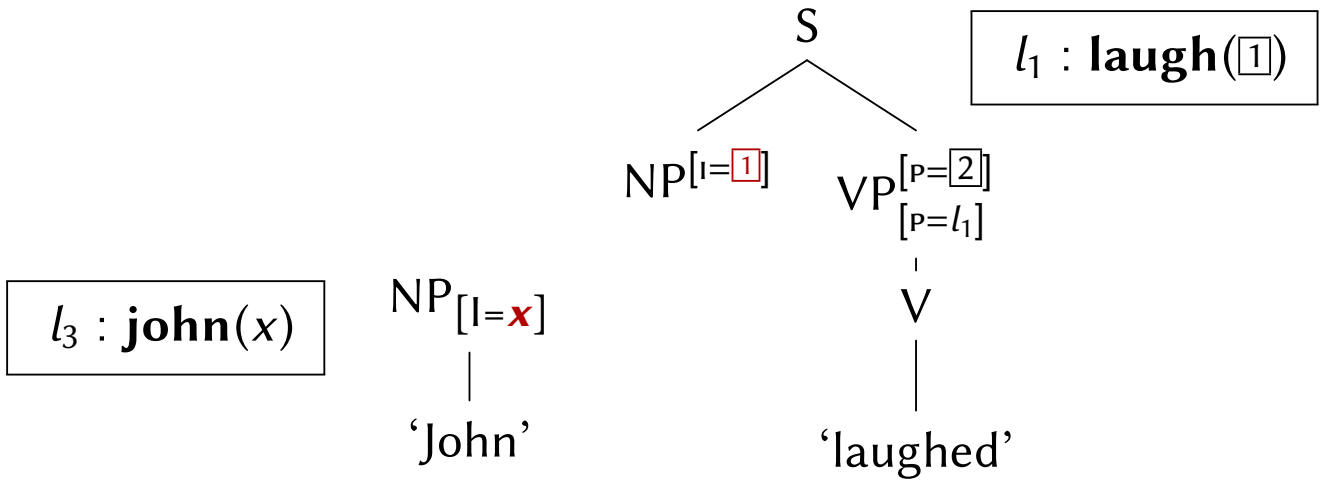
Idea: Each elementary tree is paired with

- A set of typed predicate logic expressions and of scope constraints (i.e., constraints on sub-term relations)
- interface features 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.

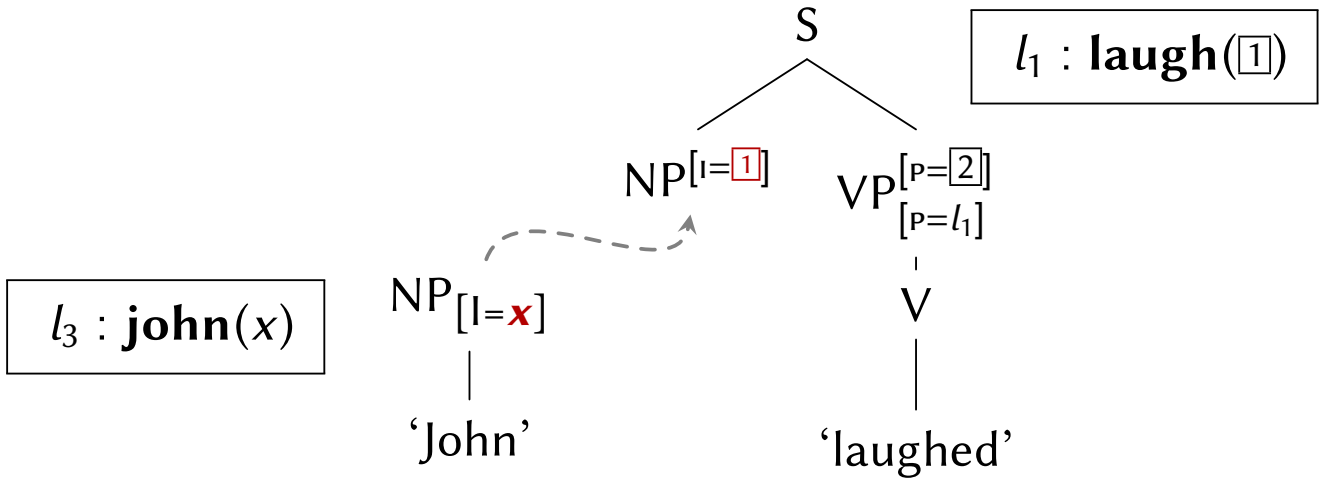
Unification-based LTAG semantics with predicate logic



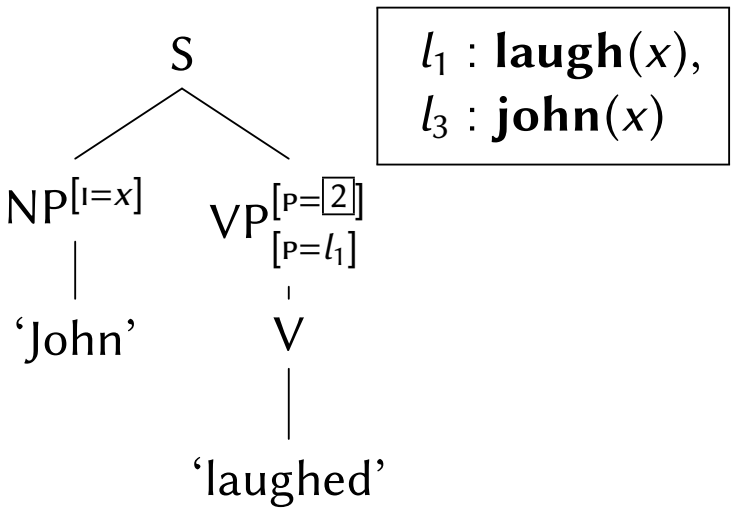
Unification-based LTAG semantics with predicate logic



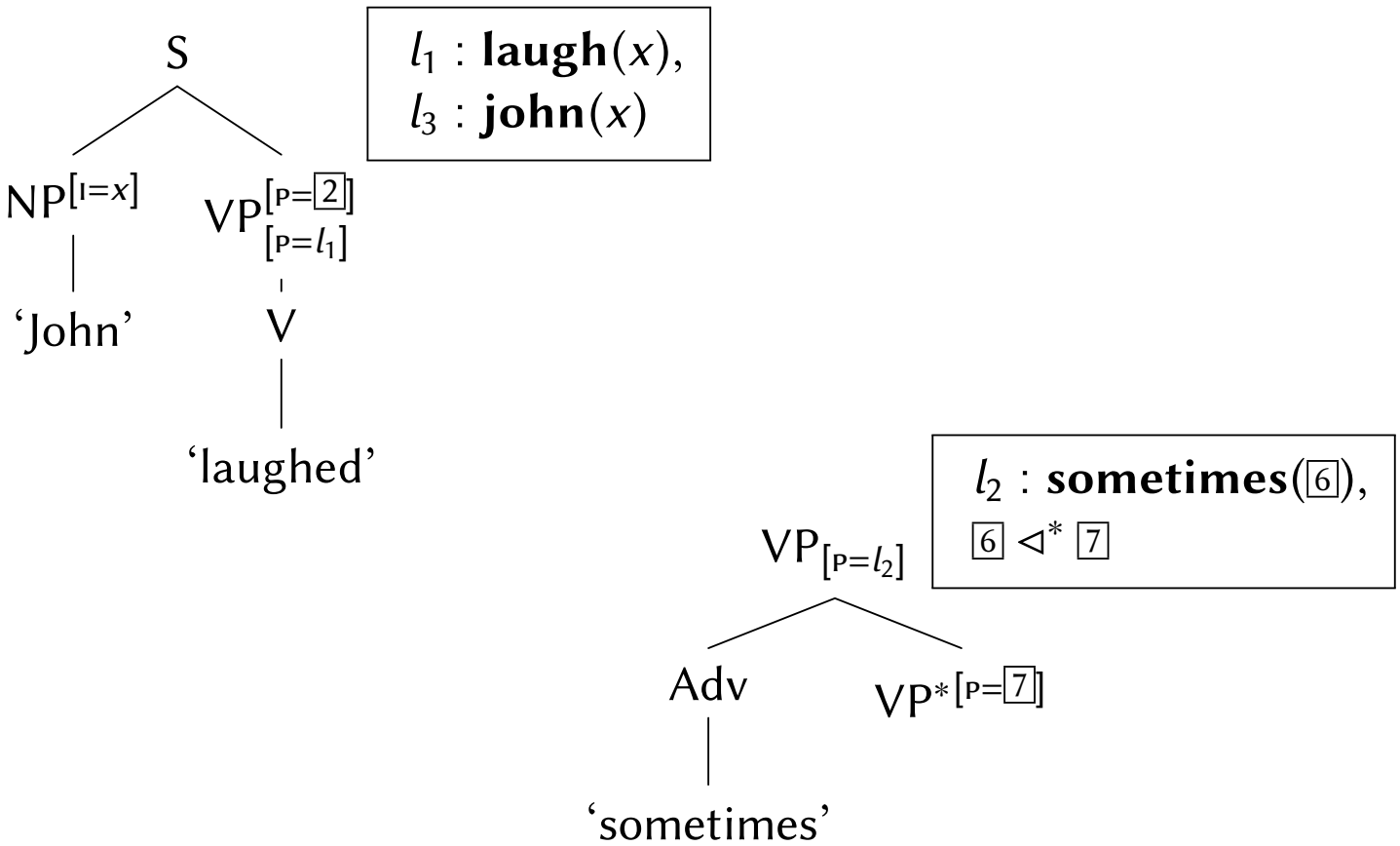
Unification-based LTAG semantics with predicate logic



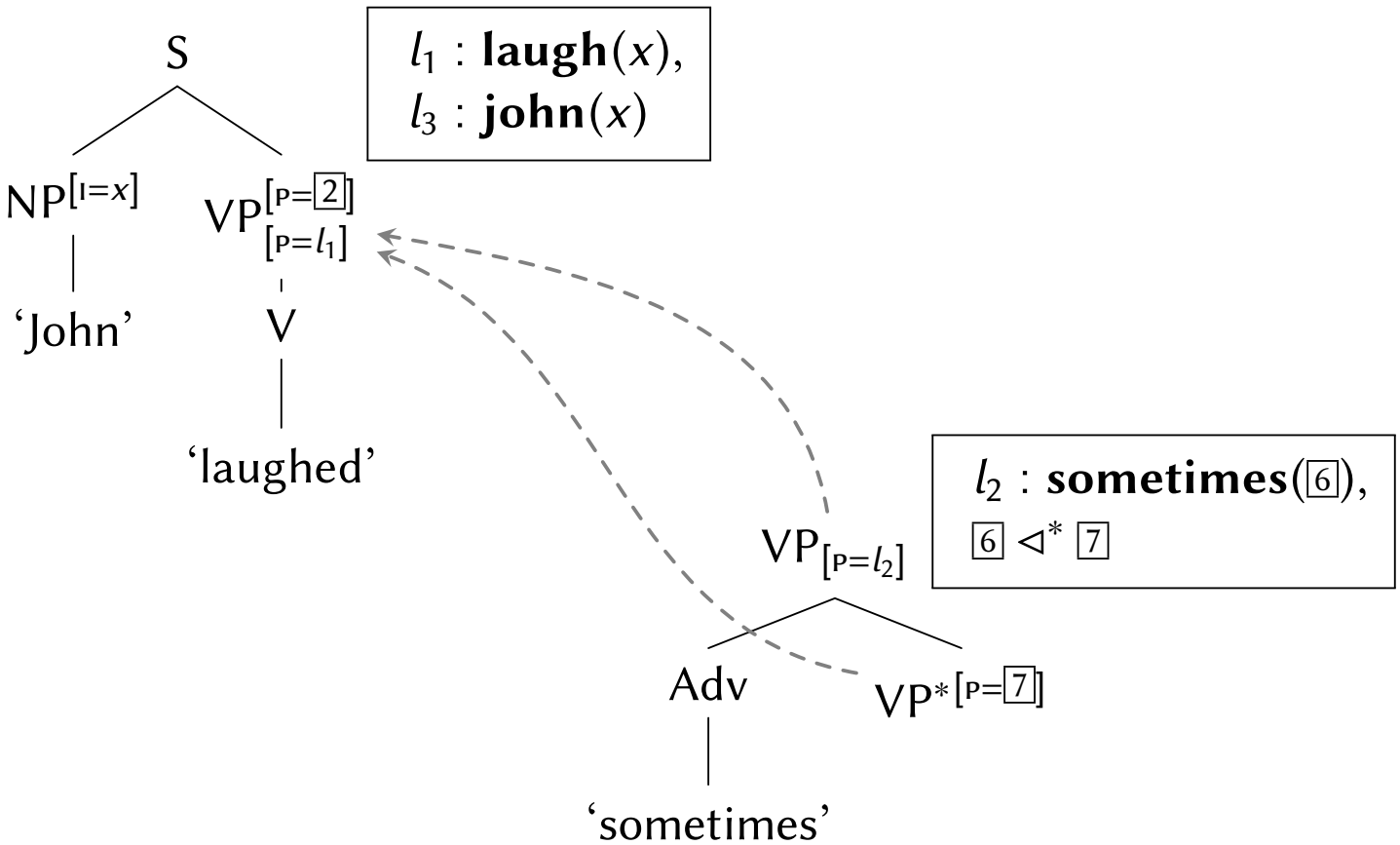
Unification-based LTAG semantics with predicate logic



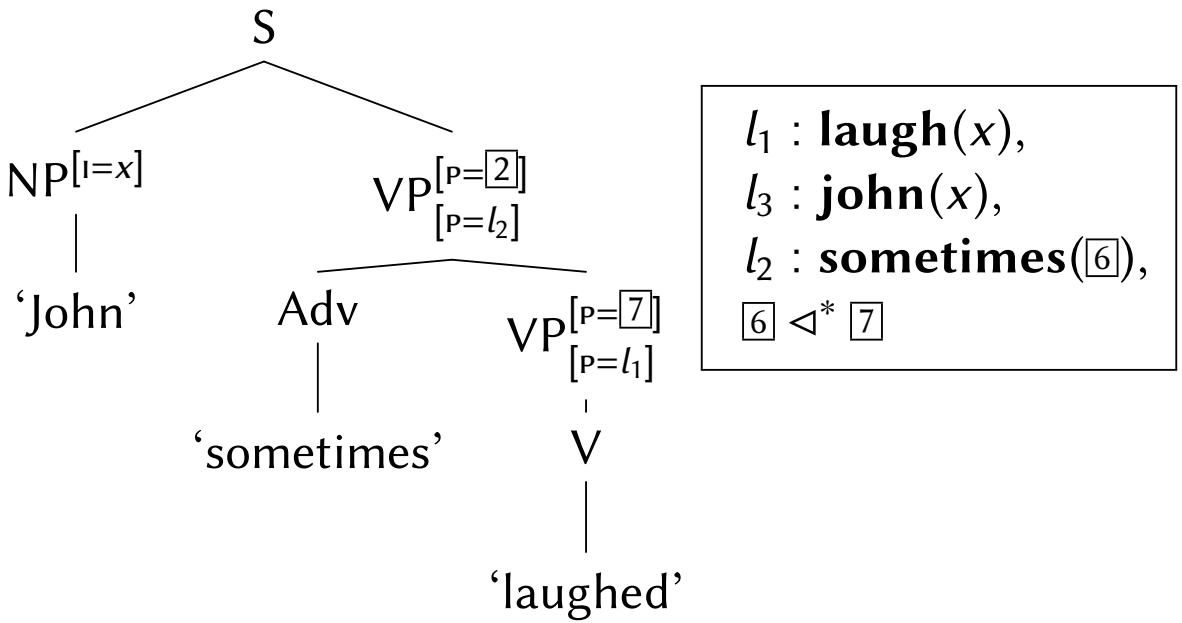
Unification-based LTAG semantics with predicate logic



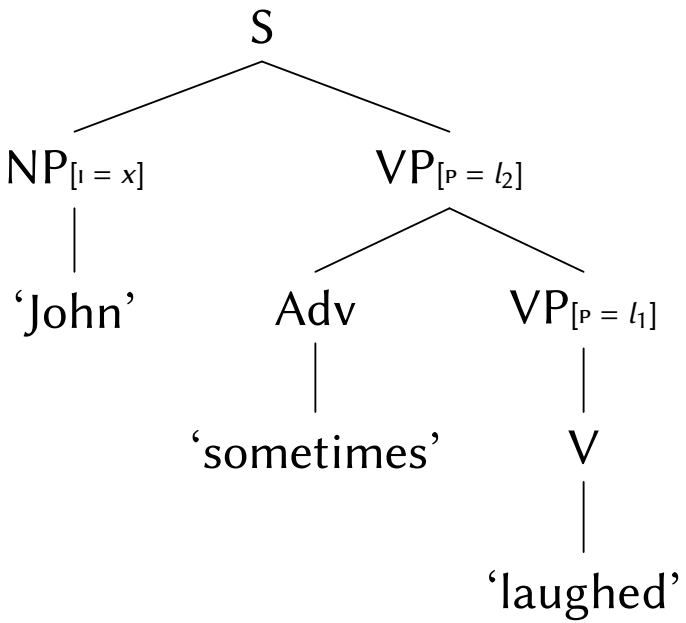
Unification-based LTAG semantics with predicate logic



Unification-based LTAG semantics with predicate logic

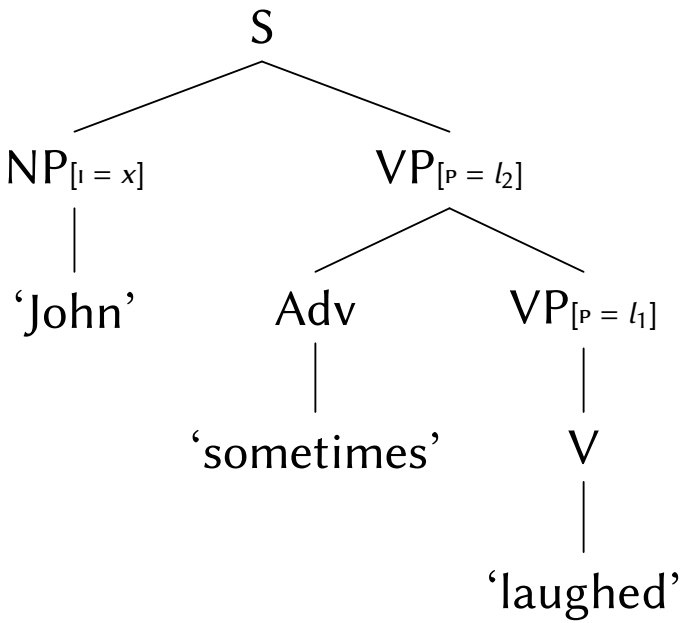


Unification-based LTAG semantics with predicate logic



$l_1 : \mathbf{laugh}(x)$,
$l_3 : \mathbf{john}(x)$,
$l_2 : \mathbf{sometimes}(\boxed{6})$,
$\boxed{6} \lhd^* l_1$

Unification-based LTAG semantics with predicate logic



$l_1 : \mathbf{laugh}(x)$,
$l_3 : \mathbf{john}(x)$,
$l_2 : \mathbf{sometimes}(\boxed{6})$,
$\boxed{6} \lhd^* l_1$

$\boxed{6} \lhd^* l_1$ signifies that the formula labeled l_1 is a subformula of the formula that has to be placed in the hole $\boxed{6}$.

Disambiguation leads to $\mathbf{john}(x) \wedge \mathbf{sometimes}(\mathbf{laugh}(x))$

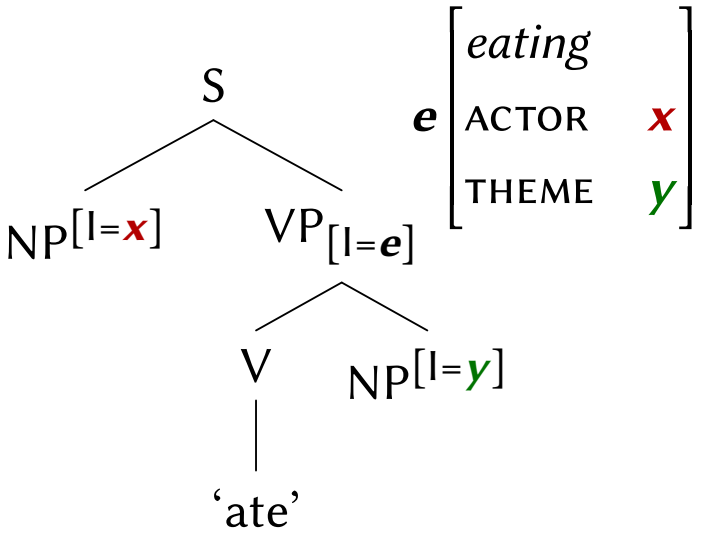
Unification-based LTAG semantics with frames

- Semantic representations are linked to entire elementary trees (as in the previous approaches).
- Semantic representations: frames, expressed as typed feature structures.
- Interface features relate nodes in the syntactic tree to nodes in the frame graph.
- Frame composition by unification, triggered by the unifications on the interface features that are in turn triggered by substitution, adjunction and final top-bottom unification on the derived tree.

Unification-based LTAG semantics with frames

Reminder:

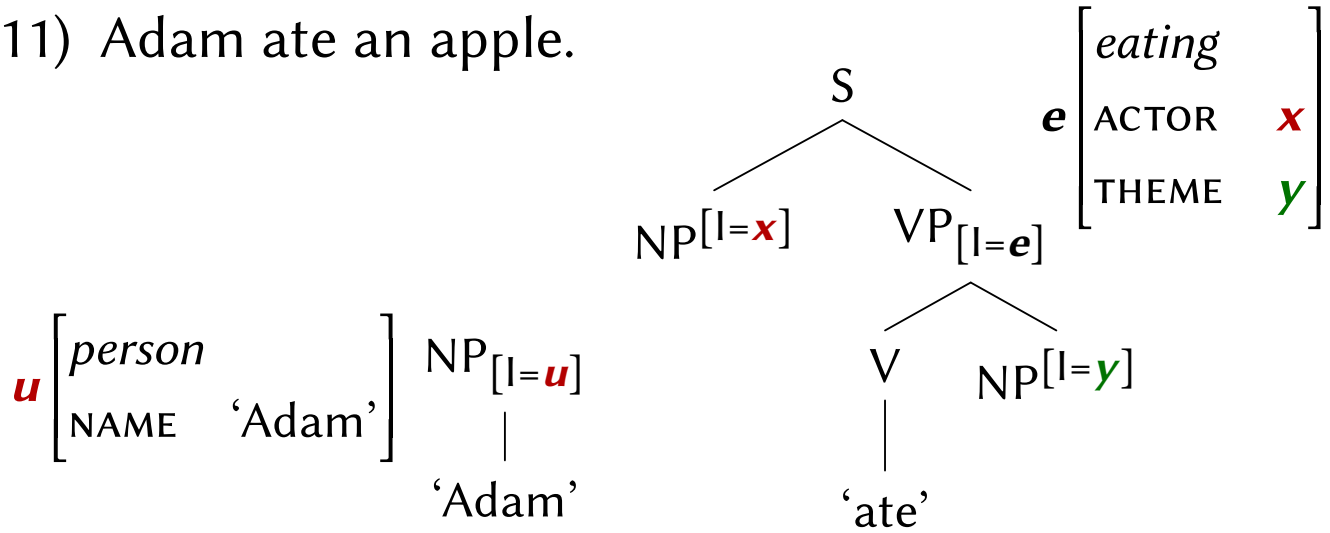
(11) Adam ate an apple.



Unification-based LTAG semantics with frames

Reminder:

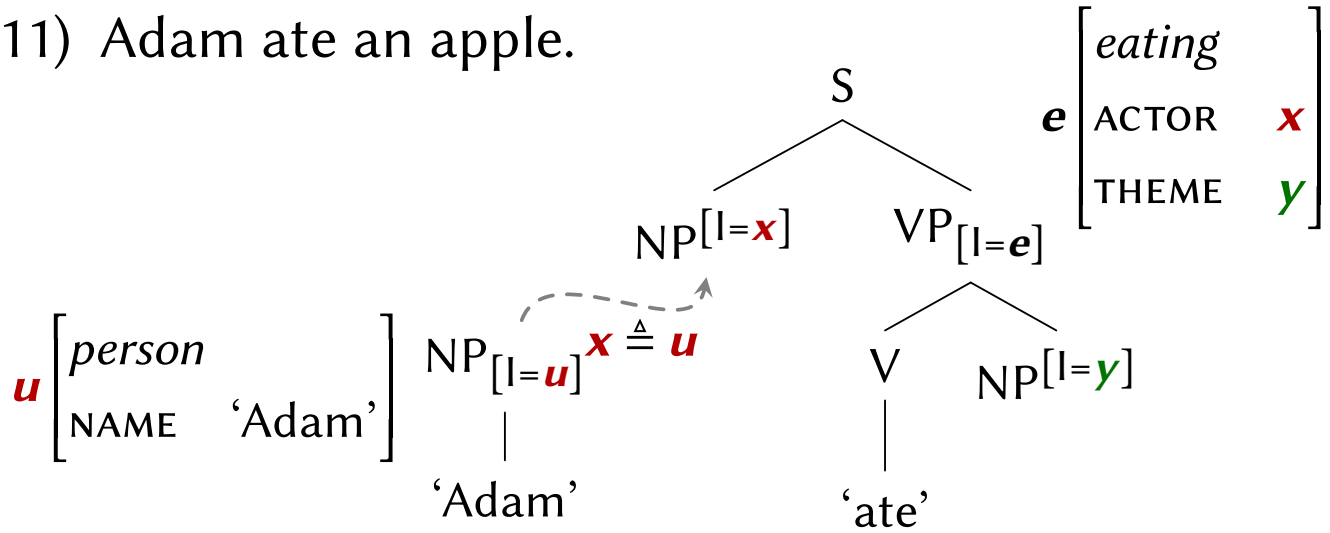
(11) Adam ate an apple.



Unification-based LTAG semantics with frames

Reminder:

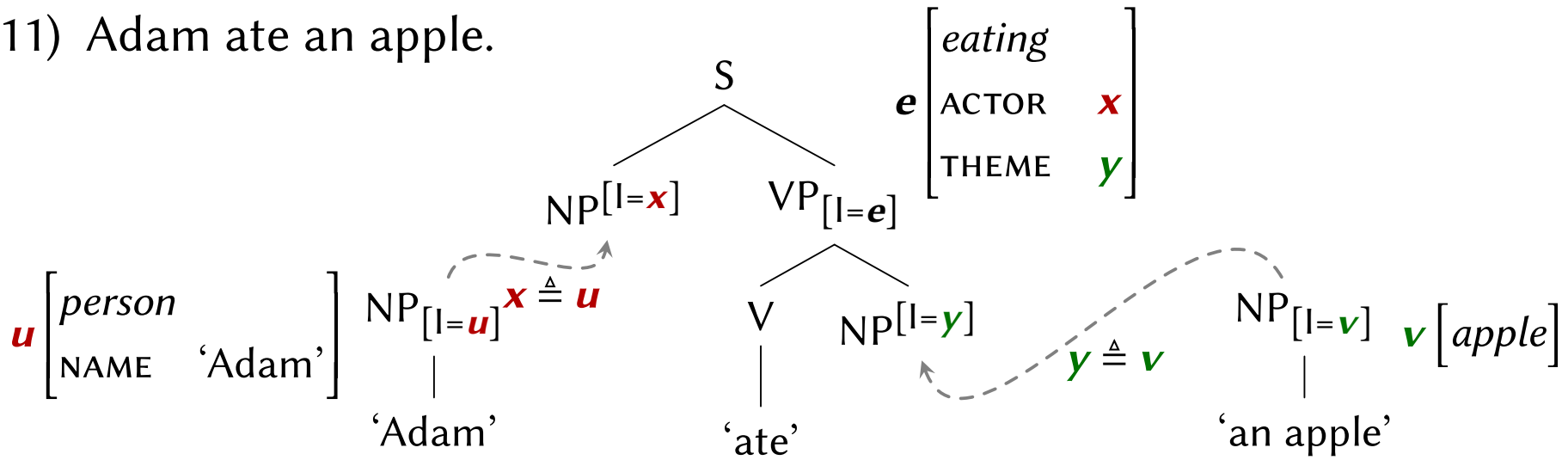
(11) Adam ate an apple.



Unification-based LTAG semantics with frames

Reminder:

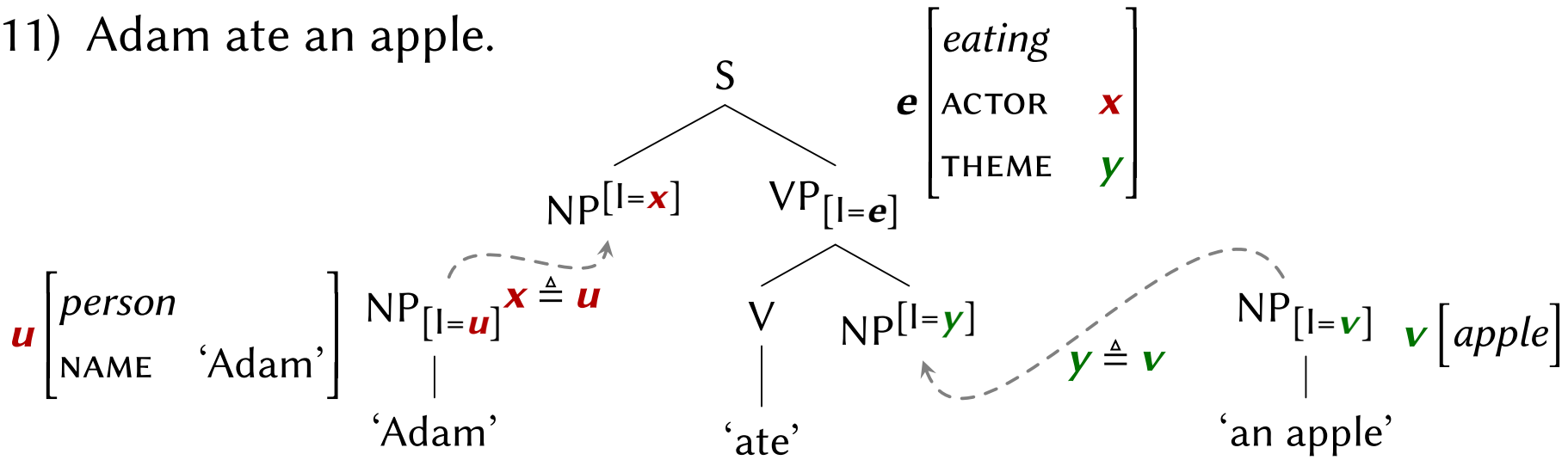
(11) Adam ate an apple.

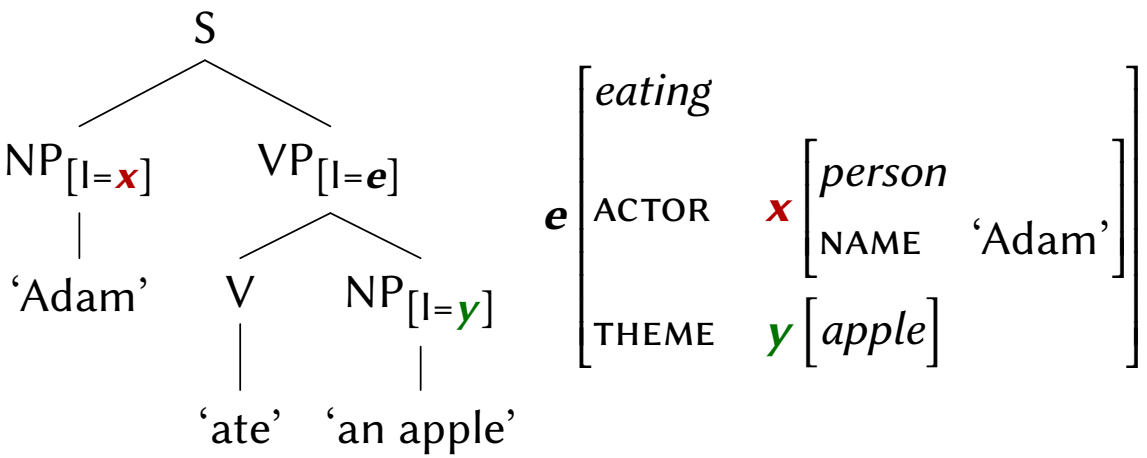


Unification-based LTAG semantics with frames

Reminder:

(11) Adam ate an apple.





Unification-based LTAG semantics with frames

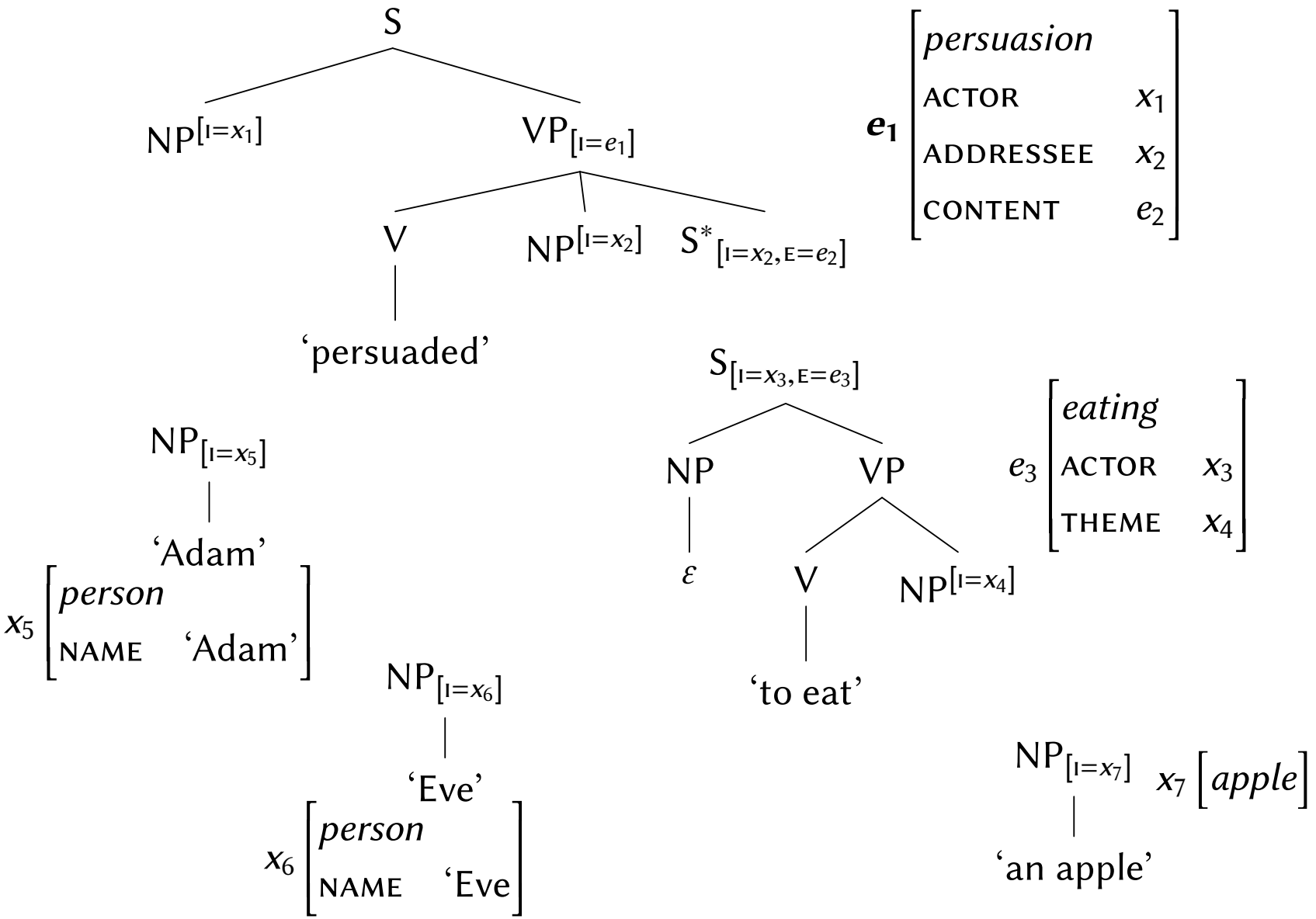
A more complex example that requires structure sharing in the frame:

(12) Adam persuaded Eve to eat an apple.

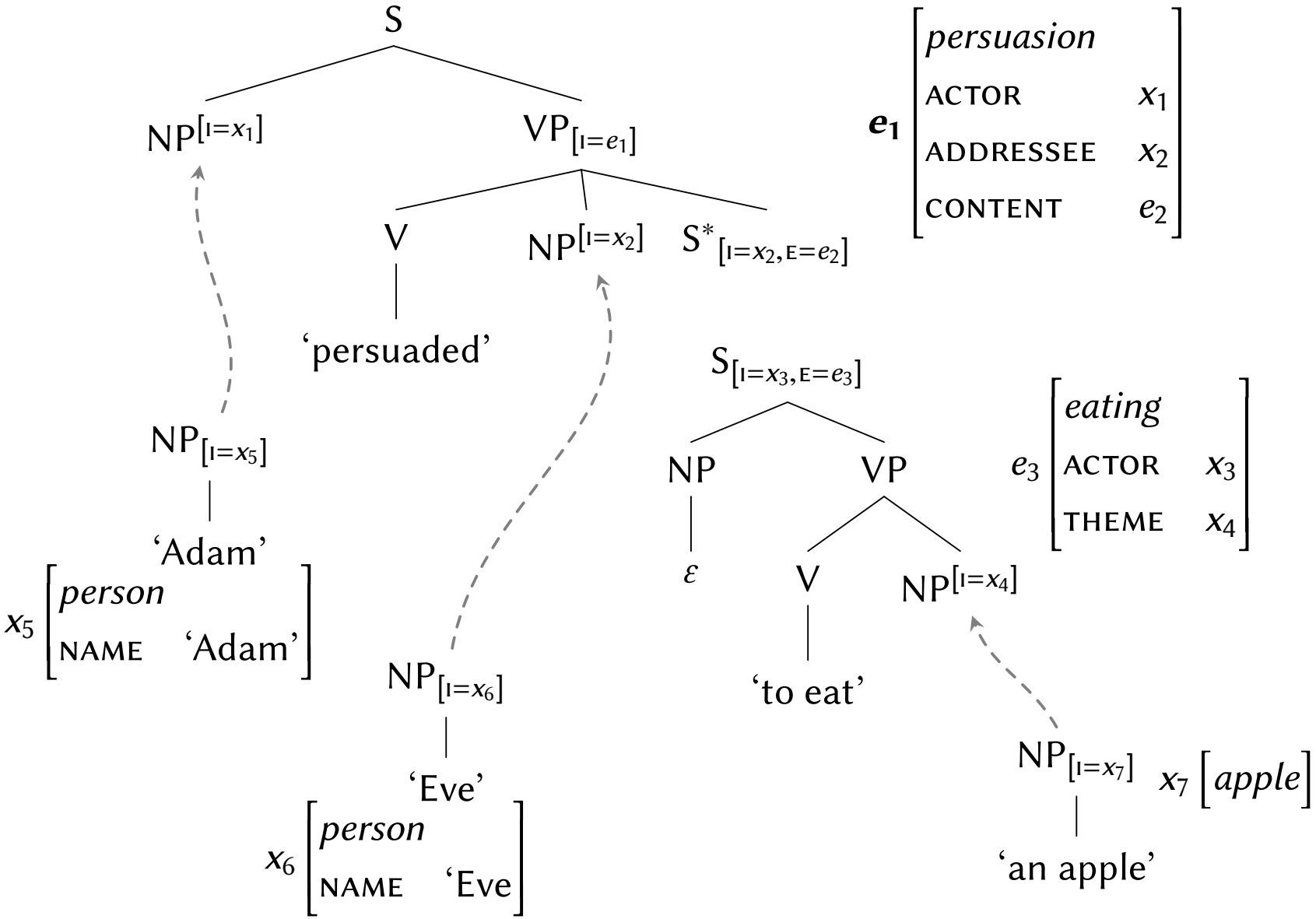
persuade is an object control verb: The object is not only an argument of *persuade* but also an implicit argument of the embedded infinitive. The semantic frame should (roughly) look like this:

$$\begin{bmatrix} \textit{persuasion} & \\ \textsc{actor} & \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{`Adam'} \end{bmatrix} \\ \textsc{addressee} & x_2 \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{`Eve'} \end{bmatrix} \\ \textsc{content} & \begin{bmatrix} \textit{eating} & \\ \textsc{actor} & x_2 \\ \textsc{theme} & \begin{bmatrix} \textit{apple} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

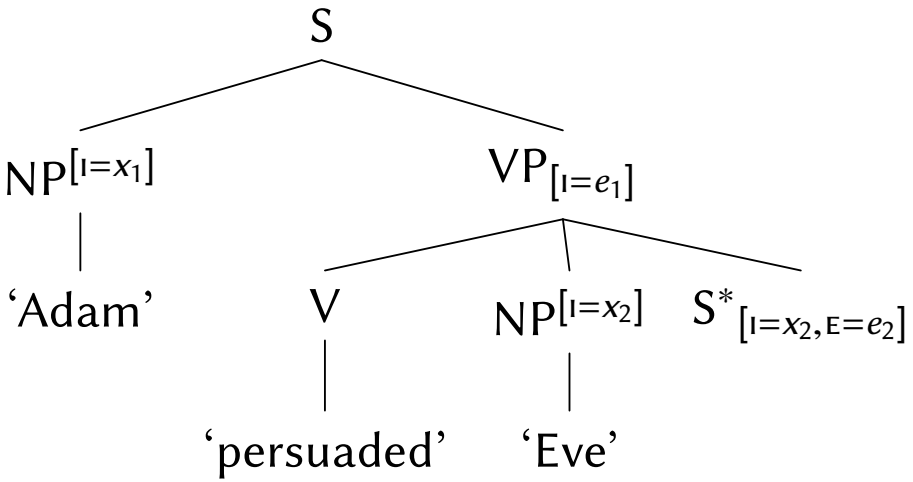
Unification-based LTAG semantics with frames



Unification-based LTAG semantics with frames

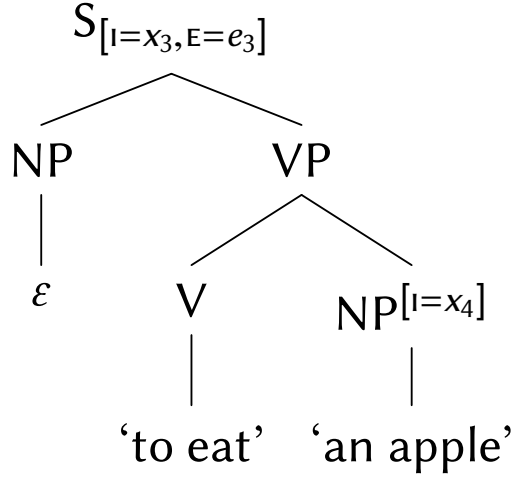


Unification-based LTAG semantics with frames

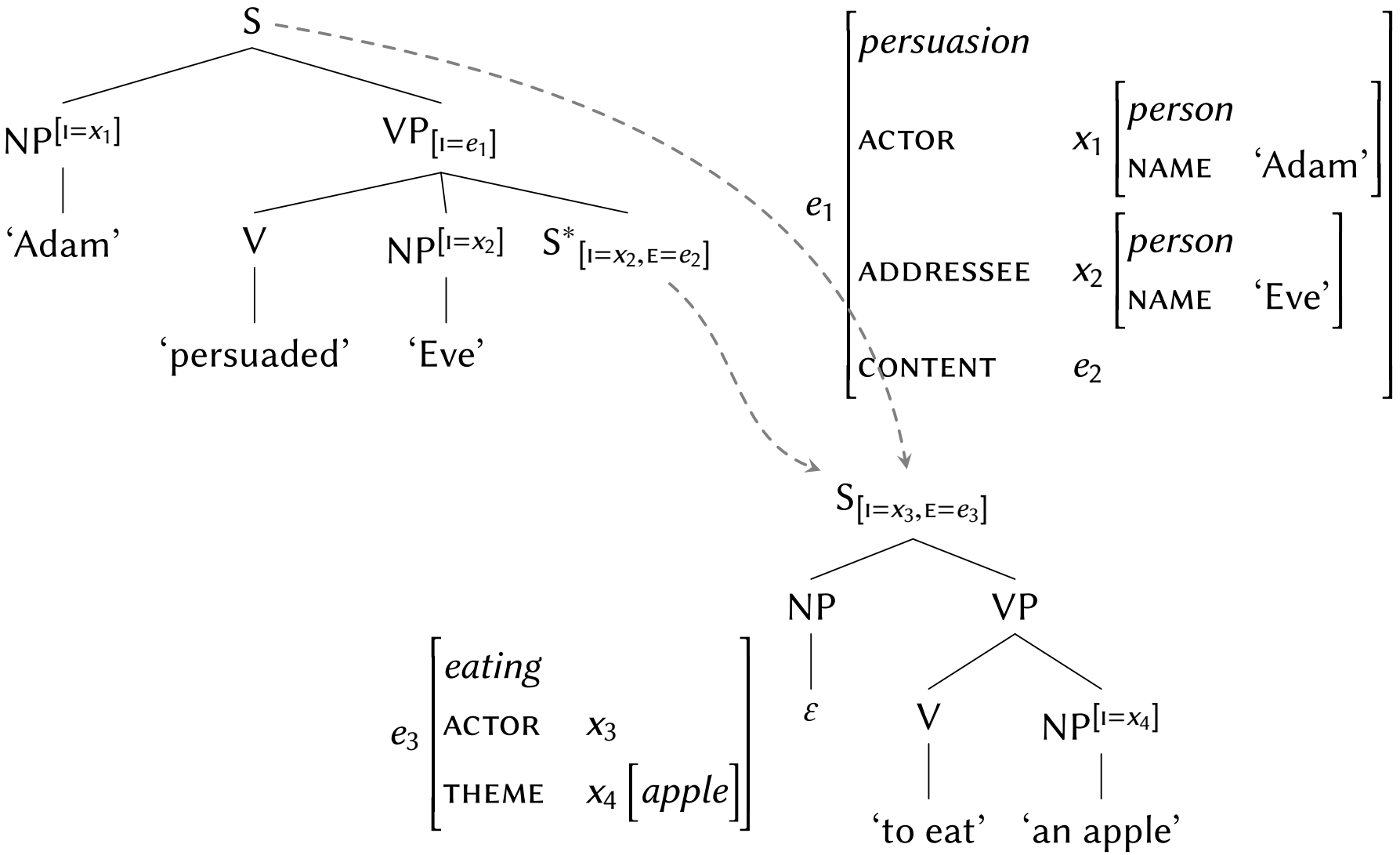


$$e_1 \begin{bmatrix} \textit{persuasion} & \\ \textsc{actor} & x_1 \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{'Adam'} \end{bmatrix} \\ \textsc{addressee} & x_2 \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{'Eve'} \end{bmatrix} \\ \textsc{content} & e_2 \end{bmatrix}$$

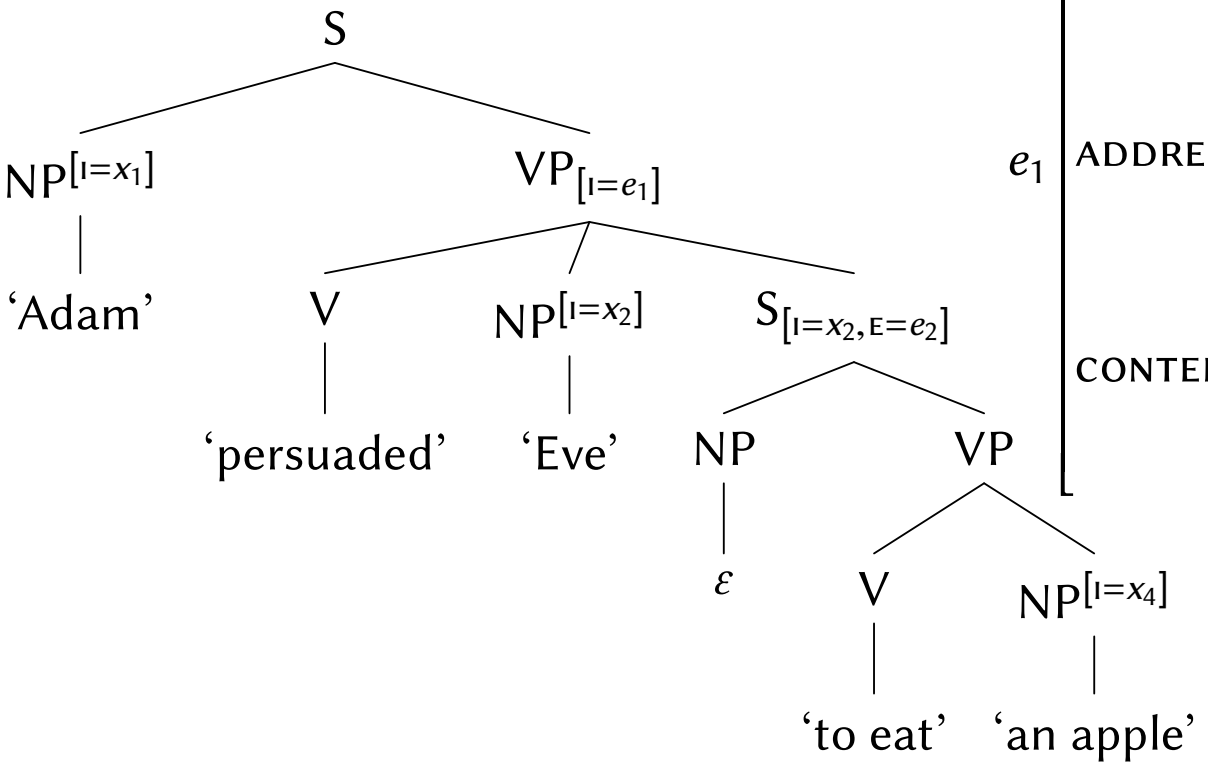
$$e_3 \begin{bmatrix} \textit{eating} & \\ \textsc{actor} & x_3 \\ \textsc{theme} & x_4 \begin{bmatrix} \textit{apple} \end{bmatrix} \end{bmatrix}$$



Unification-based LTAG semantics with frames



Unification-based LTAG semantics with frames



$$e_1 \begin{bmatrix} \textit{persuasion} & & \\ \textsc{actor} & x_1 & \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{'Adam'} \end{bmatrix} \\ \textsc{addressee} & x_2 & \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{'Eve'} \end{bmatrix} \\ \textsc{content} & e_2 & \begin{bmatrix} \textit{eating} & \\ \textsc{actor} & x_2 \\ \textsc{theme} & x_4 \begin{bmatrix} \textit{apple} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Outline of today's course

1. Extraction phenomena in LTAG
2. Generalization and factorization within the elementary trees
   - Tree families
   - LTAG & metagrammar specification
3. LTAG semantics
   - Synchronous TAGs for semantics
   - Unification-based LTAG semantics with predicate logic
   - Unification-based LTAG semantics with frames
4. Summary & outlook

Summary & outlook

Summary

- LTAG features for extraction phenomena
- grammar factorization
- LTAG approaches for the syntax-semantics interface

Summary & outlook

Summary

- LTAG features for extraction phenomena
- grammar factorization
- LTAG approaches for the syntax-semantics interface

Tomorrow

- introduction to frame semantics
- formalization using attribute value logic
- type constraints, type hierarchy
- extensions

References

Crabbé, Benoit, Denys Duchier, Claire Gardent, Joseph Le Roux & Yannick Parmentier. 2013. XMG: eXtensible MetaGrammar. **Computational Linguistics** 39(3). 1–66. http://hal.archives-ouvertes.fr/hal-00768224/en/.

Gardent, Claire & Laura Kallmeyer. 2003. Semantic Construction in FTAG. In **Proceedings of eacl 2003**, 123–130. Budapest.

Kallmeyer, Laura & Aravind K. Joshi. 2003. Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. **Research on Language and Computation** 1(1–2). 3–58.

Kallmeyer, Laura & Rainer Osswald. 2013. Syntax-driven semantic frame composition in Lexicalized Tree Adjoining Grammars. **Journal of Language Modelling** 1(2). 267–330.

Kallmeyer, Laura, Rainer Osswald & Sylvain Pogodalla. 2016. *For*-adverbials and aspectual interpretation: An LTAG analysis using hybrid logic and frame semantics. In Christopher Piñón (ed.), **Empirical issues in syntax and semantics EISS**, vol. 11, .

Kallmeyer, Laura & Maribel Romero. 2008. Scope and situation binding in LTAG using semantic unification. **Research on Language and Computation** 6(1). 3–52.

Nesson, Rebecca & Stuart M. Shieber. 2006. Simpler TAG semantics through synchronization. In **Proceedings of the 11th conference on formal grammar**, Malaga, Spain. http://www.eecs.harvard.edu/~shieber/Biblio/Papers/Nesson-2006-SSS.pdf.

Nesson, Rebecca & Stuart M. Shieber. 2008. Synchronous vector tag for syntax and semantics: Control verbs, relative clauses, and inverse linking. In **Proceedings of the ninth international workshop on tree adjoining grammars and related formalisms (tag+ 9)**, Tübingen, Germany.

References (cont.)

Shieber, Stuart M. 1994. Restricting the weak-generative capacity of synchronous Tree-Adjoining Grammars. **Computational Intelligence** 10(4). 271–385.

Shieber, Stuart M. & Yves Schabes. 1990. Synchronous Tree-Adjoining Grammars. In **Proceedings of coling**, 253–258.

XTAG Research Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Tech. rep. Institute for Research in Cognitive Science, University of Pennsylvania Philadelphia, PA.