Towards a formalization of Role & Reference Grammar

Laura Kallmeyer
(joint work with Rainer Osswald)

Heinrich-Heine-Universität Düsseldorf

27. 03. 2018, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier LIRMM

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

erc European Research Council

SFB 991

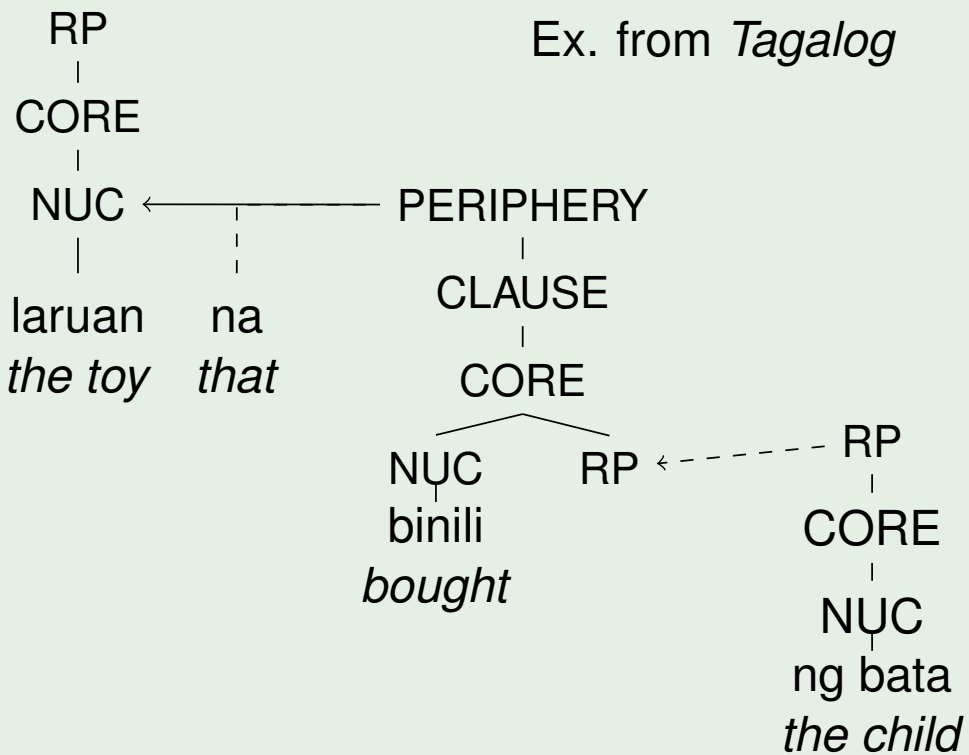
Outline

Introduction

Role and Reference Grammar (RRG; Van Valin & LaPolla 1997; Van Valin 2005) is a typologically rich grammar theory.

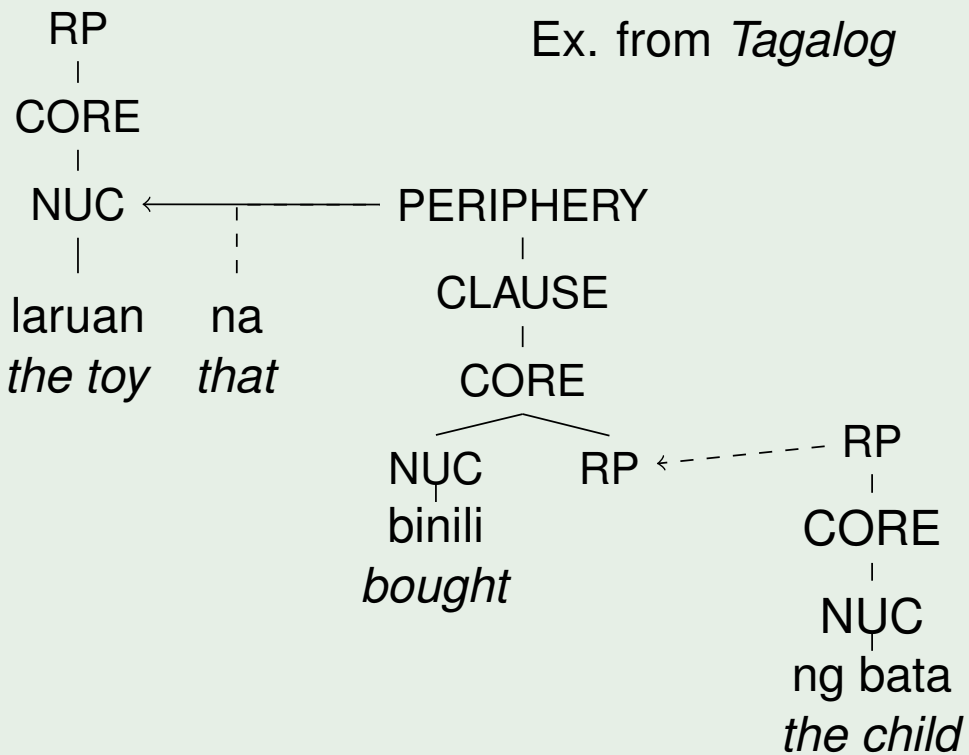
Introduction

Role and Reference Grammar (RRG; Van Valin & LaPolla 1997; Van Valin 2005) is a typologically rich grammar theory.



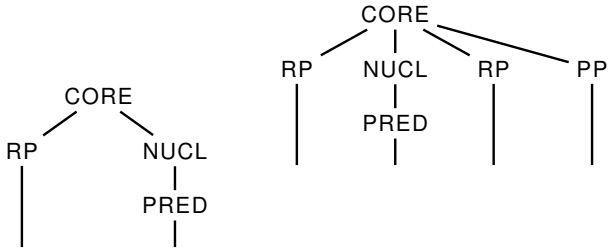
Introduction

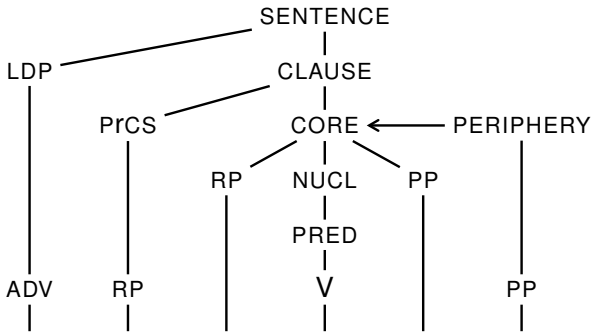
Role and Reference Grammar (RRG; Van Valin & LaPolla 1997; Van Valin 2005) is a typologically rich grammar theory.

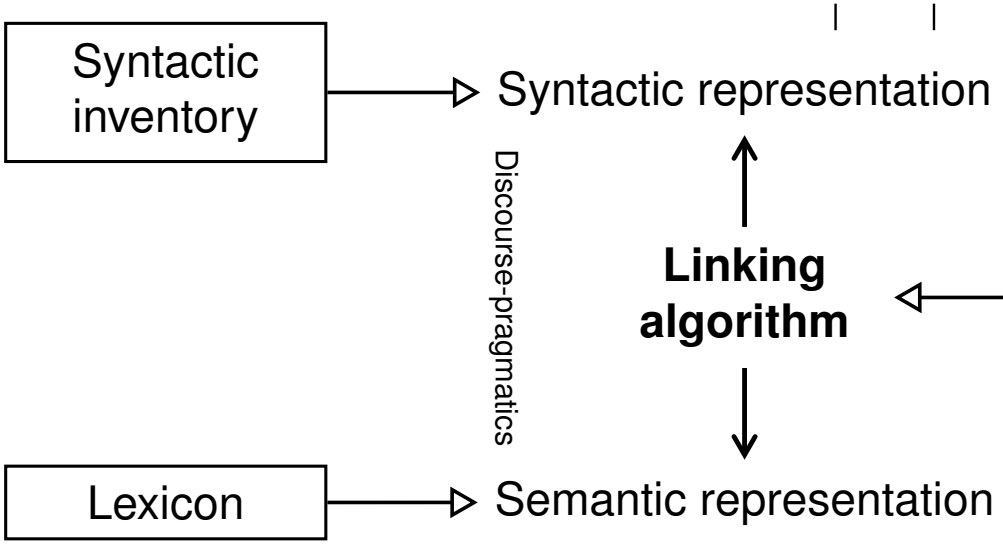


But: not fully formalized, no implementation framework.

The architecture of Role & Reference Grammar (RRG)



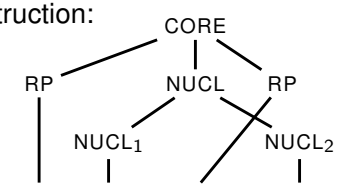




$[\textbf{do}'(x, \varnothing)]$ CAUSE [INGR **shattered**$'(y)$]

$\langle_{\text{IF}}$ *INT* $\langle_{\text{TNS}}$ *PRES* $\langle_{\text{ASP}}$ *PERF PROG* $\langle$**do**$'$(Kim, [**cry**$'$(Kim)]$\rangle\rangle\rangle\rangle$

MORPHOLOGY —
SYNTAX Juncture: nuclear
Nexus: cosubordination
Construction:



Linking: default
SEMANTICS [SEM$_{\text{NUCL}_1}$] CAUSE [SEM$_{\text{NUCL}_2}$]
PRAGMATICS unspecified

Introduction

Why is a formal perspective on RRG useful (and for whom)?

- Is a formalization relevant for the working typologist?

Introduction

Why is a formal perspective on RRG useful (and for whom)?

- Is a formalization relevant for the working typologist?

  Maybe not, but it can help to eliminate **inconsistencies** and **gaps** of the theory.

Introduction

Why is a formal perspective on RRG useful (and for whom)?

- Is a formalization relevant for the working typologist?

  Maybe not, but it can help to eliminate **inconsistencies** and **gaps** of the theory.

- Doesn't RRG already come with a lot of formal elements?

Introduction

Why is a formal perspective on RRG useful (and for whom)?

- Is a formalization relevant for the working typologist?

  Maybe not, but it can help to eliminate **inconsistencies** and **gaps** of the theory.

- Doesn't RRG already come with a lot of formal elements?

  Sure, but these elements are not defined with **logical** and **mathematical** rigor.

Introduction

Why is a formal perspective on RRG useful (and for whom)?

- Is a formalization relevant for the working typologist?

  Maybe not, but it can help to eliminate **inconsistencies** and **gaps** of the theory.

- Doesn't RRG already come with a lot of formal elements?

  Sure, but these elements are not defined with **logical** and **mathematical** rigor.

- Further advantages:

  A formalization can serve as a basis (in fact, is a requirement) for a **computational treatment** of RRG.
  It allows us to study the **generative power** of RRG and the **complexity issues** related to processing RRG-based grammars.
  Moreover, the formalization should make it easier to **extend** and **modify** the theory.

General plan of the formalization

- Take **all explanatory components** of RRG into account.
- Develop a **declarative**, constraint-based formulation.

Introduction

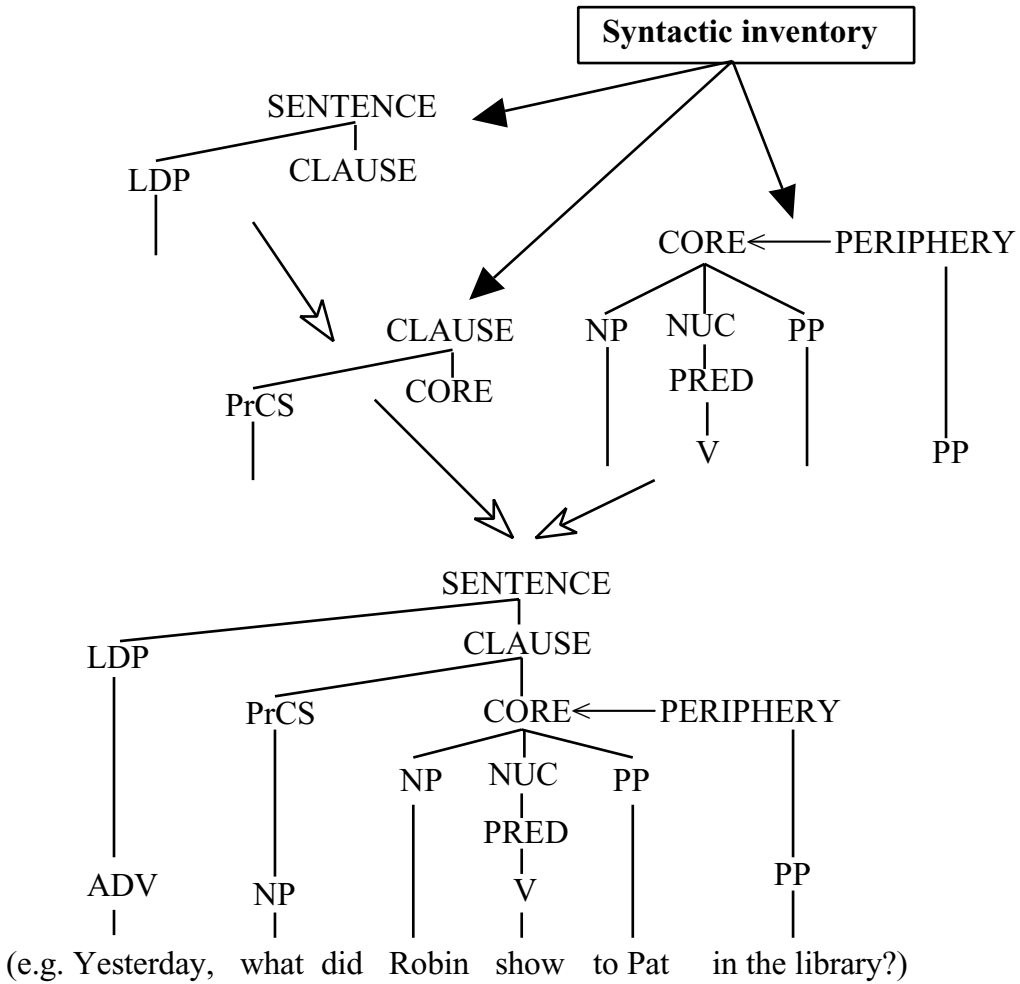
General plan of the formalization

- Take **all explanatory components** of RRG into account.
- Develop a **declarative**, constraint-based formulation.

Some of the tasks

- **Syntactic** representation
  Formal specification of the syntactic inventory and of the compositional operations on trees
- **Semantic** representation
  Clarification of the logical (and model-theoretic) aspects of RRG's logical structures
- **Linking** algorithm
  Non-procedural, inherently bidirectional description as a system of constraints

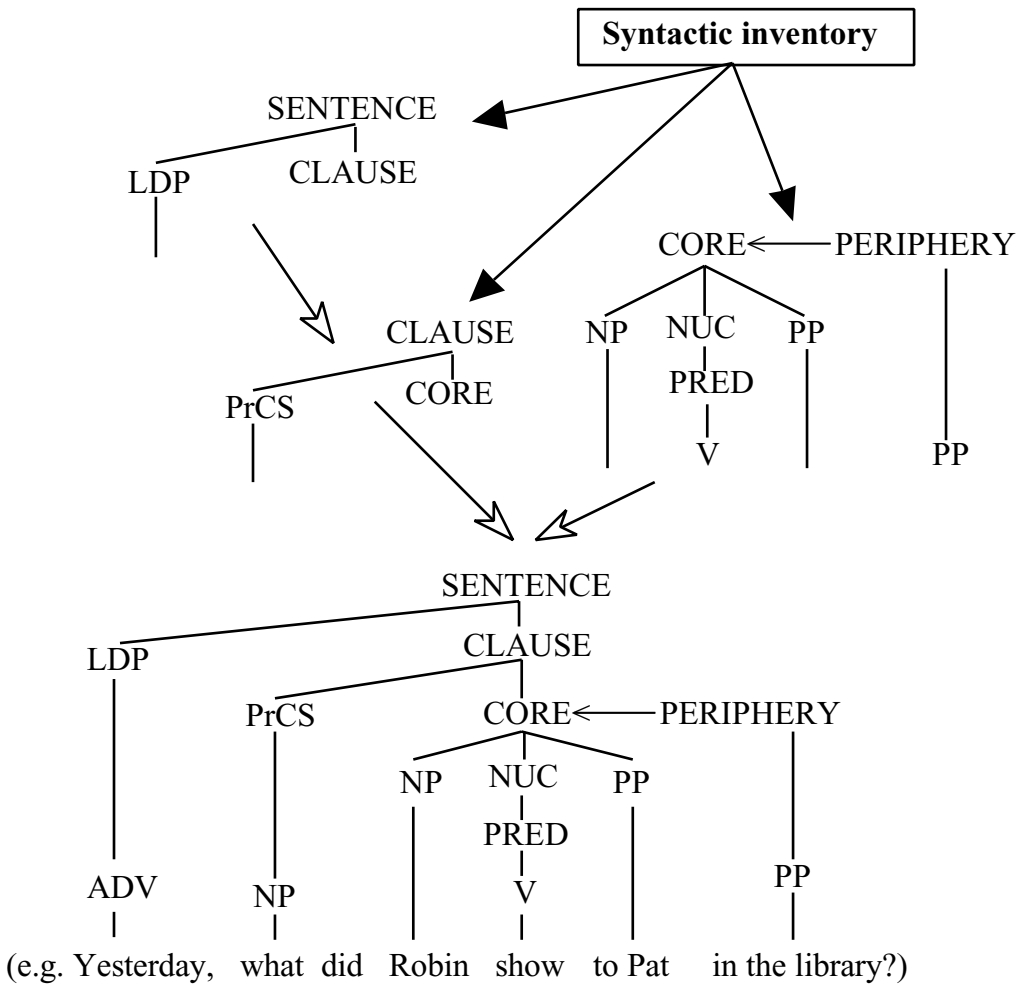
The inventory of syntactic templates



[Van Valin 2005, p. 15]

Syntactic representation

The inventory of syntactic templates



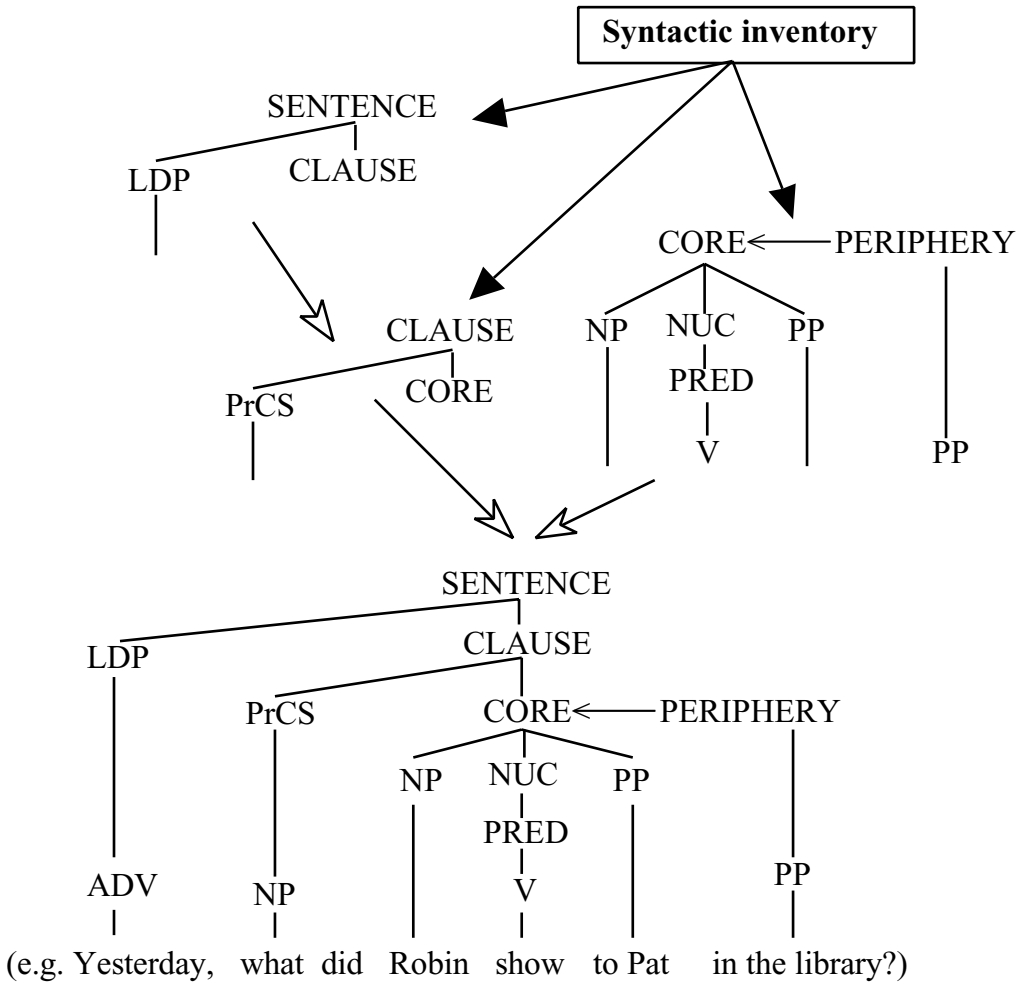
[Van Valin 2005, p. 15]

Issues

- How are syntactic templates defined?
- How do they combine?

Syntactic representation

The inventory of syntactic templates



[Van Valin 2005, p. 15]

Issues

- How are syntactic templates defined?
- How do they combine?

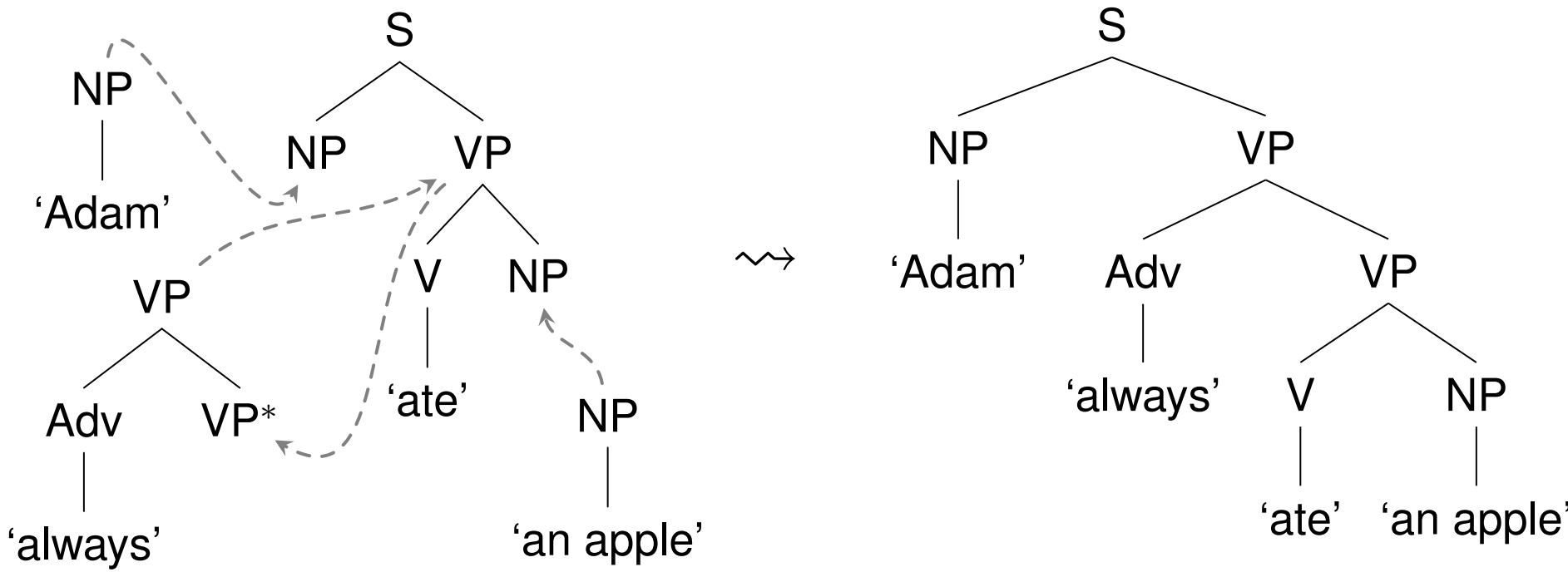
Proposal

- Use concepts from (Lexicalized) Tree Adjoining Grammars (LTAG)
- Adapt the LTAG formalism to the syntactic dimension of RRG

Background: LTAG

Lexicalized Tree Adjoining Grammars (LTAG)

- Tree-rewriting system
- Finite set of (**lexicalized**) **elementary trees**.
- Two operations: **substitution** (replacing a leaf with a new tree) and **adjunction** (replacing an internal node with a new tree).



Background: LTAG

Two key properties of the LTAG formalism

- **Extended domain of locality**

  The full argument projection of a lexical item can be represented by a single elementary tree.

  Elementary trees can have a complex constituent structure.

- **Factoring recursion from the domain of dependencies**

  Constructions related to iteration and recursion are modeled by adjunction.

  Through adjunction, the local dependencies encoded by elementary trees can become long-distance dependencies in the derived trees.

Background: LTAG

Two key properties of the LTAG formalism

- **Extended domain of locality**

  The full argument projection of a lexical item can be represented by a single elementary tree.

  Elementary trees can have a complex constituent structure.

- **Factoring recursion from the domain of dependencies**

  Constructions related to iteration and recursion are modeled by adjunction.

  Through adjunction, the local dependencies encoded by elementary trees can become long-distance dependencies in the derived trees.

Slogan: "**Complicate locally**, **simplify globally**" [Bangalore/Joshi 2010]

Background: LTAG

"**Simplify globally**"

- The composition of elementary trees can be expressed by two general operations: substitution and adjunction.

  (Since basically all linguistic constraints are specified over the local domains represented by elementary trees.)

"**Complicate locally**"

- Elementary trees can have complex semantic representations which are not necessarily derived compositionally (in the syntax) from smaller parts of the trees.

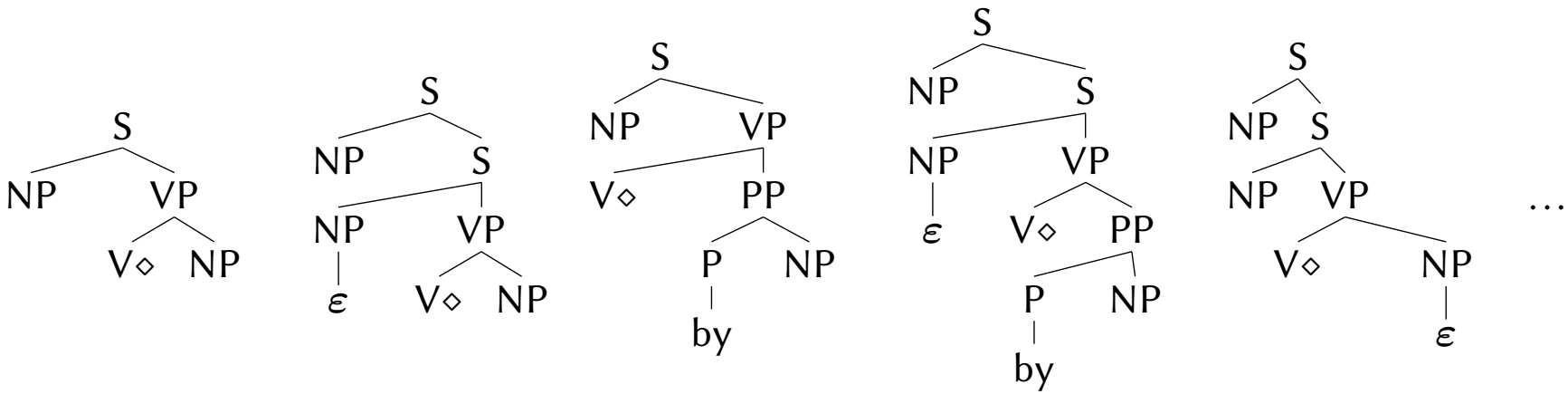
  In particular, there is no need to reproduce the internal structure of an elementary syntactic tree within its associated semantic representation. [Kallmeyer/Joshi 2003]

Background: LTAG

Tree families

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

Example unanchored family for transitive verbs

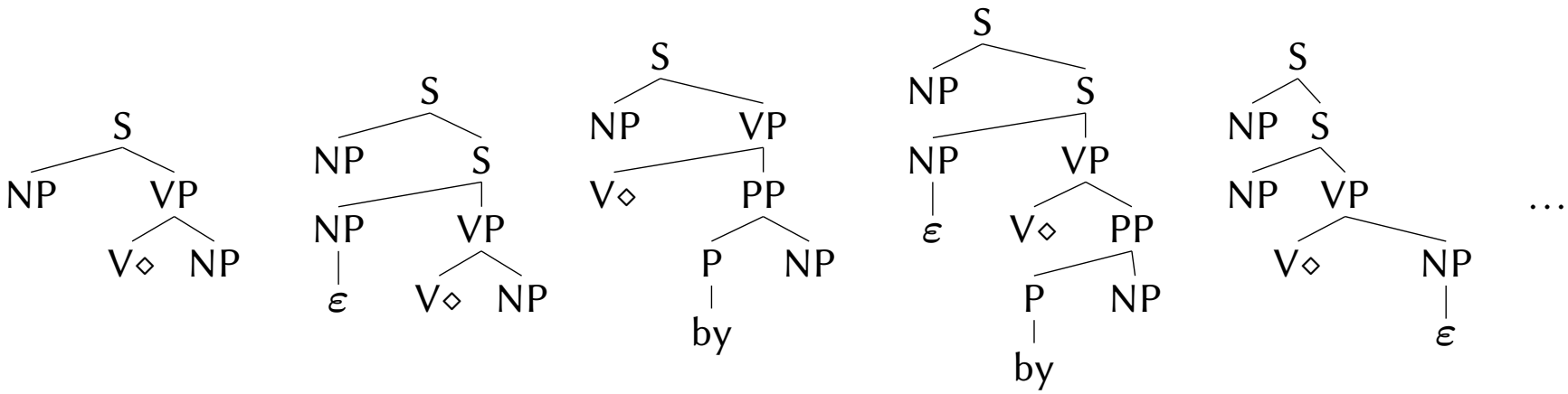


Background: LTAG

Tree families

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

Example unanchored family for transitive verbs

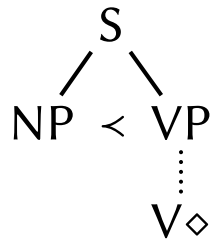


Metagrammar

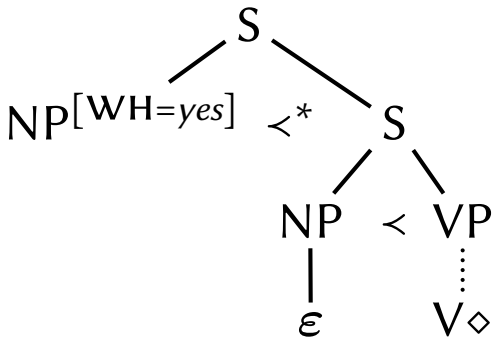
Modular characterization of elementary trees by a system of tree descriptions.

Decomposition/factorization in the metagrammar

Class *CanSubj*



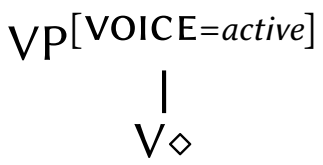
Class *ExtrSubj*



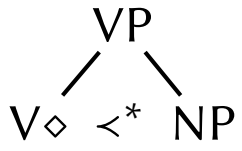
Class *Subj*

CanSubj ∨ *ExtSubj*

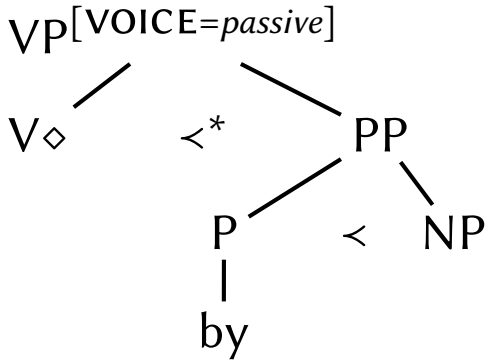
Class *ActV*



Class *DirObj*



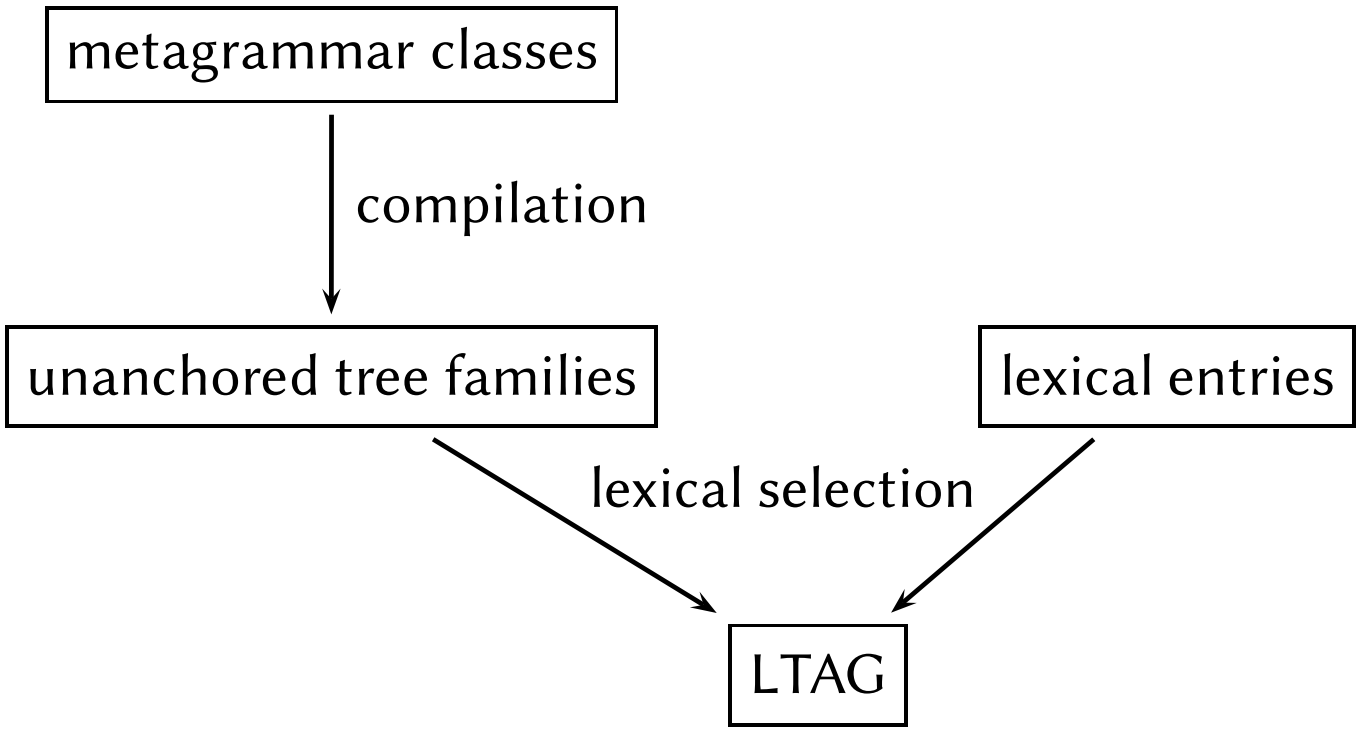
Class *ByObj*



Class *PassV*

Background: LTAG

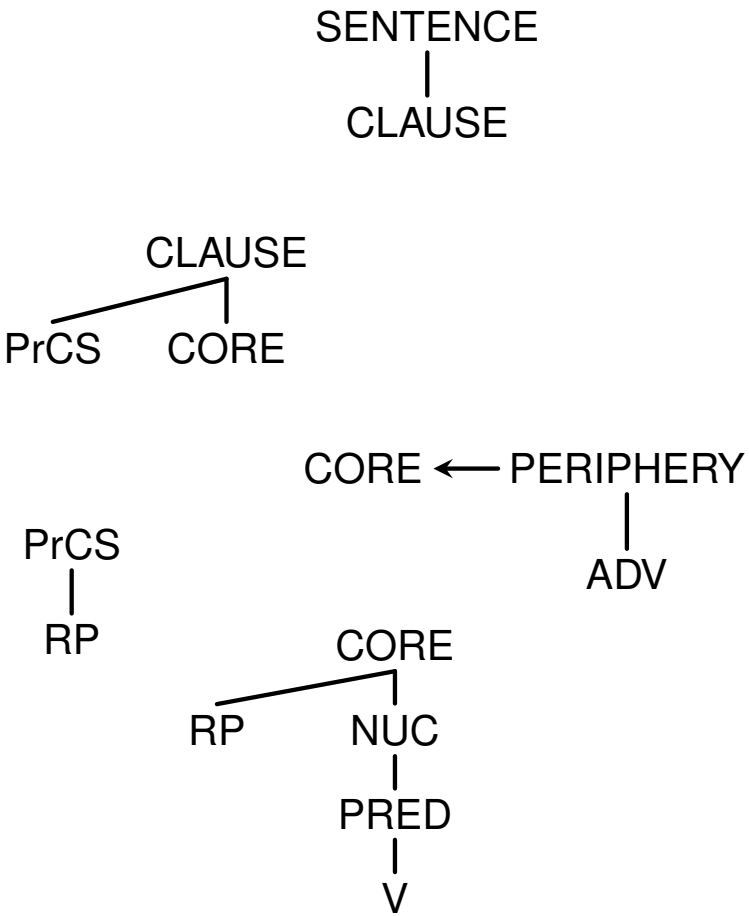
Decomposition/factorization in the metagrammar



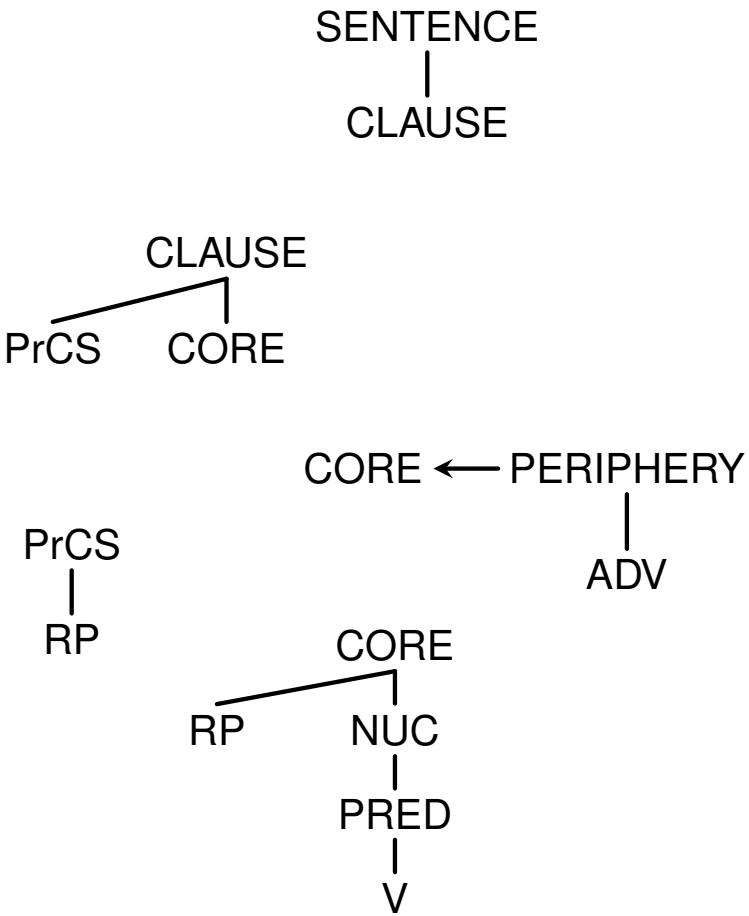
Advantage

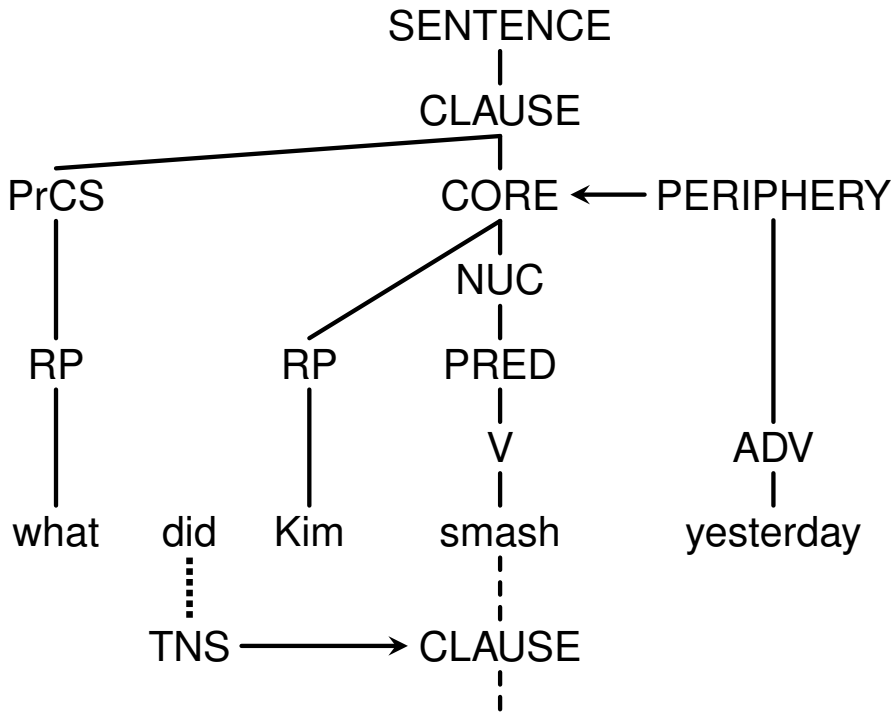
The metagrammar allows one to express and implement lexical and constructional generalizations.

Syntactic templates in RRG

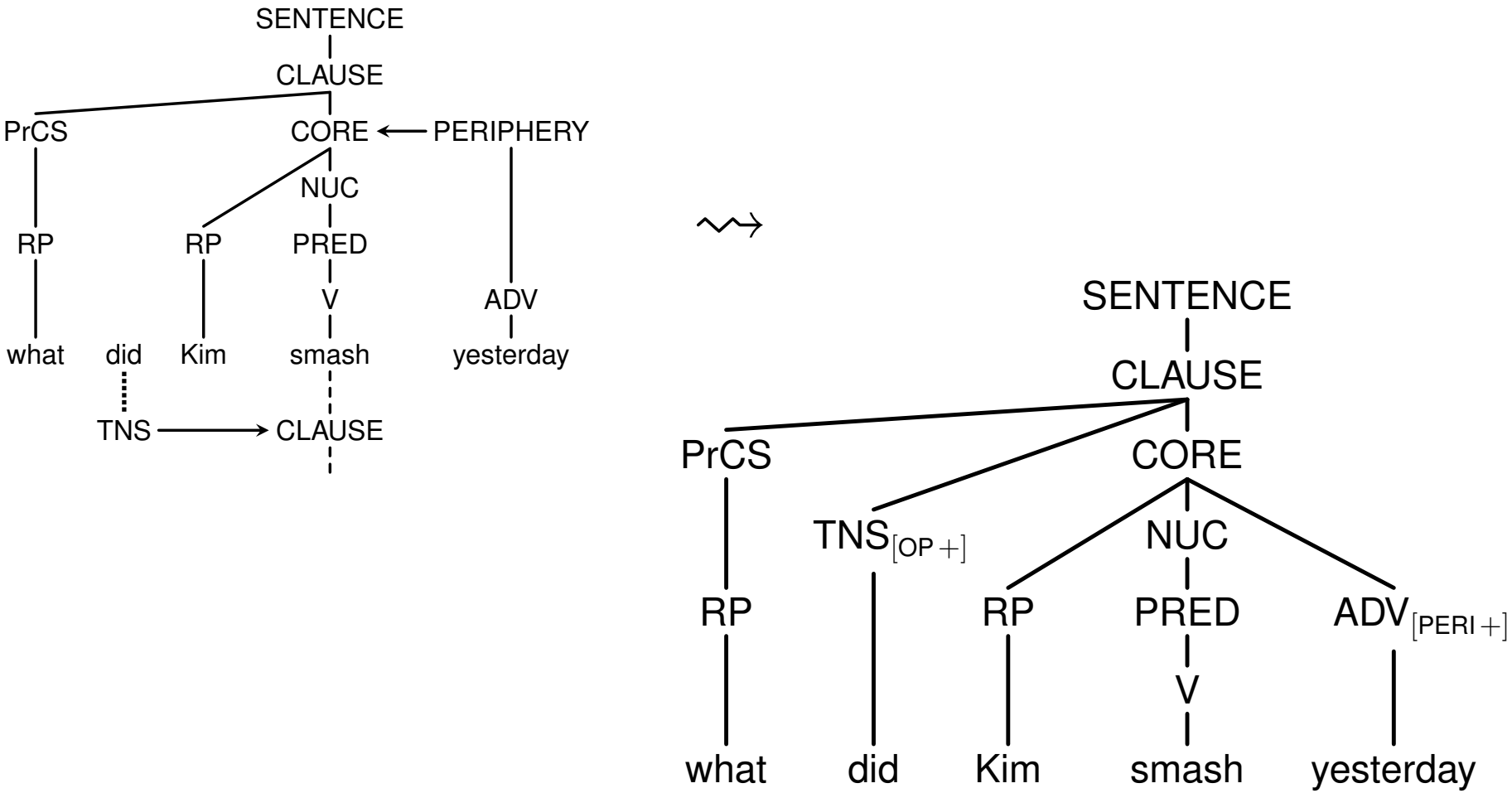


Syntactic templates in RRG





Modified representation



Syntactic representation

Application of the LTAG formalism to RRG

- What are the **elementary trees** of RRG?
- What are their **modes of composition**?
- How can they be characterized as minimal models of **metagrammatical specifications**?

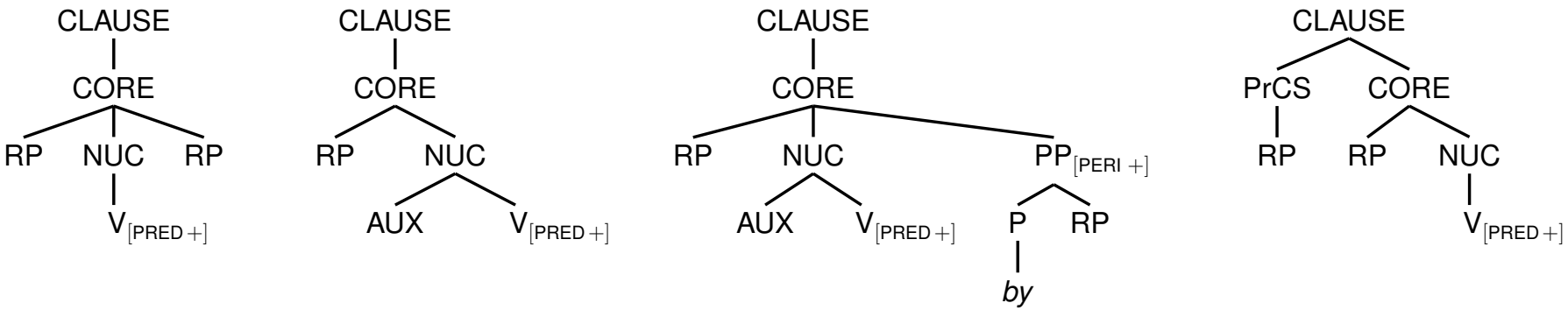
Syntactic representation

Application of the LTAG formalism to RRG

- What are the **elementary trees** of RRG?
- What are their **modes of composition**?
- How can they be characterized as minimal models of **metagrammatical specifications**?

Possible candidates for elementary trees in RRG

- Basic predication templates and their variants, e.g.

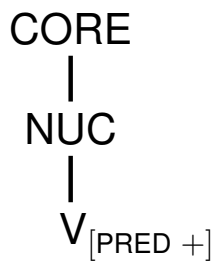


- Constructional schemas (strictly speaking, their syntactic dimension)

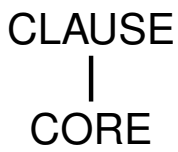
Syntactic representation

Metagrammar sketches

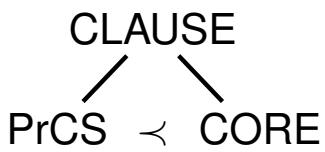
core-spine



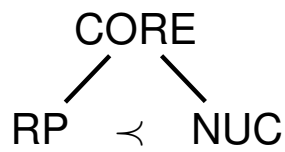
core-clause



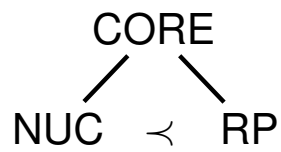
precore-slot



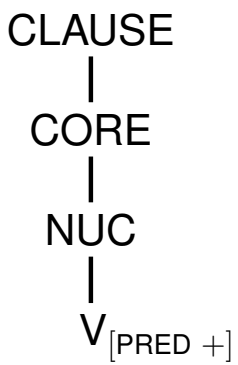
prenuc-rp



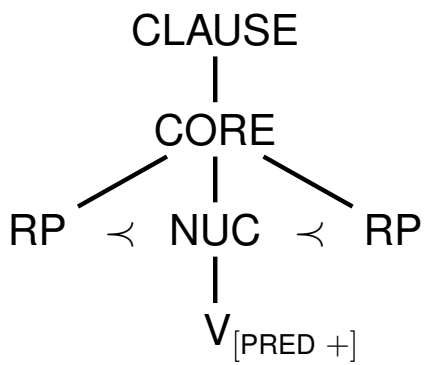
postnuc-rp



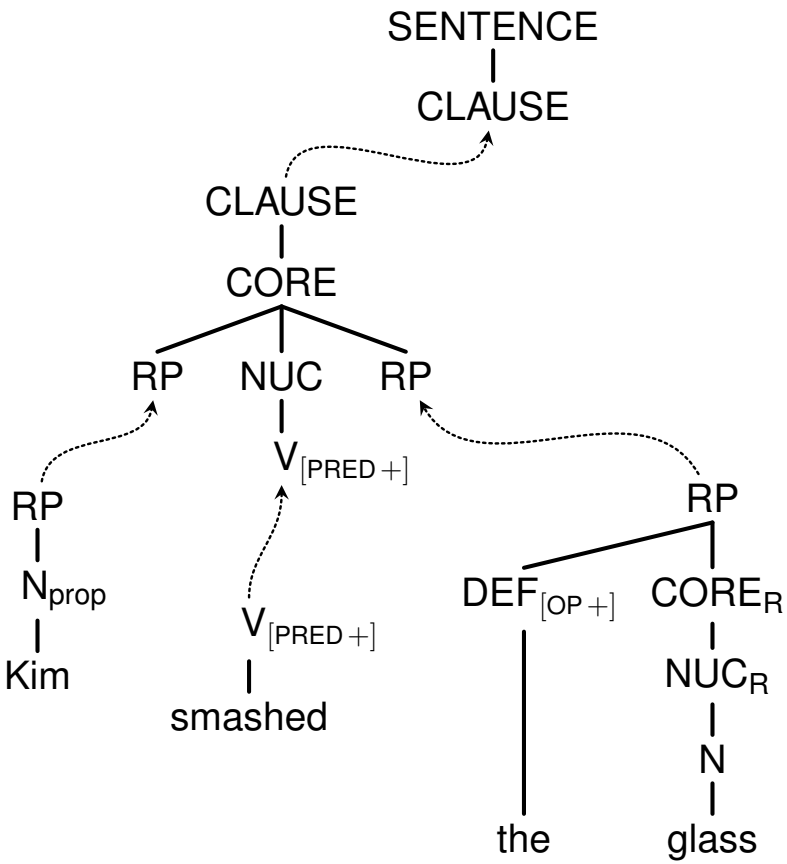
clause-spine :=
core-spine ∧ *core-clause*

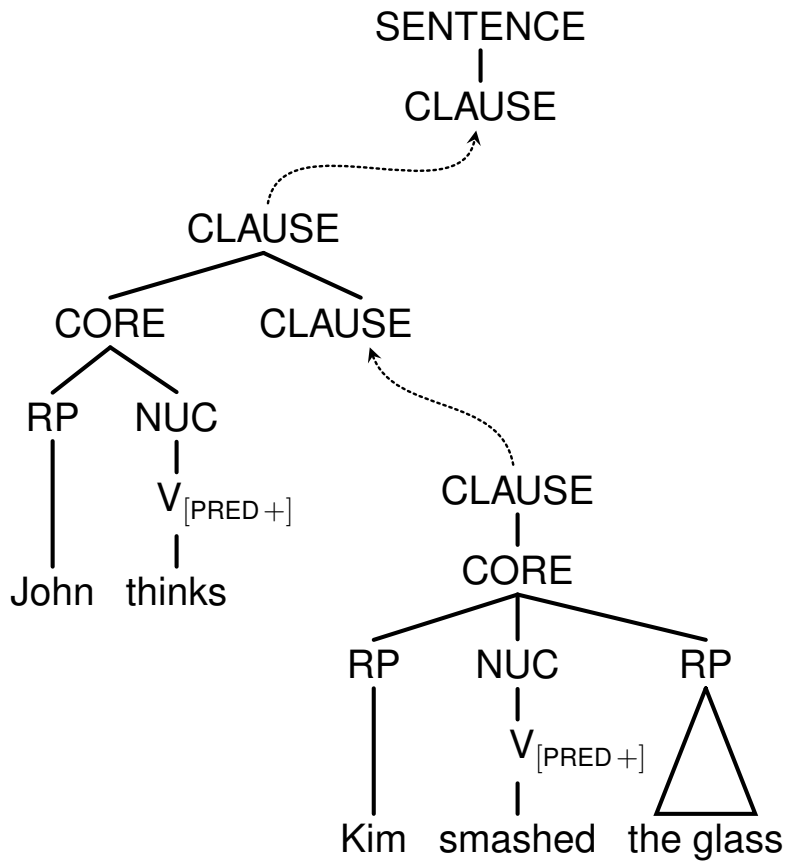


base-transitive :=
clause-spine ∧ *prenuc-rp* ∧ *postnuc-rp*

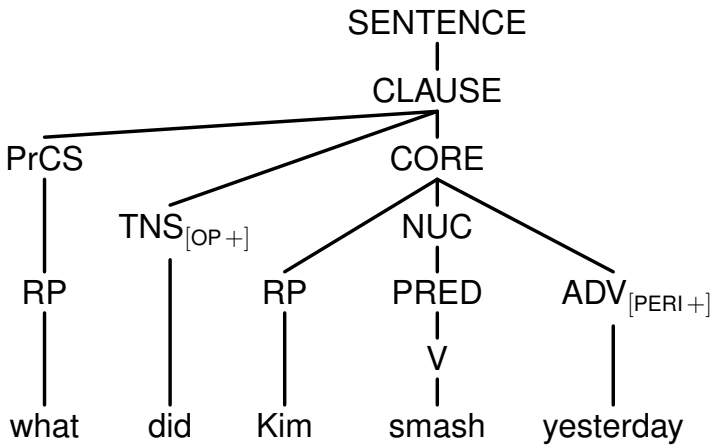


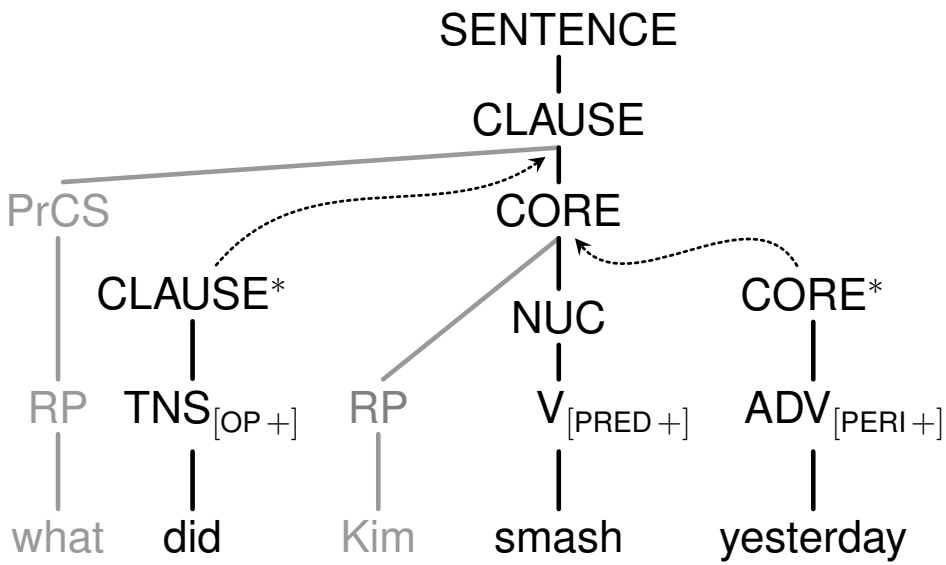
Mode of composition I: (simple) substitution



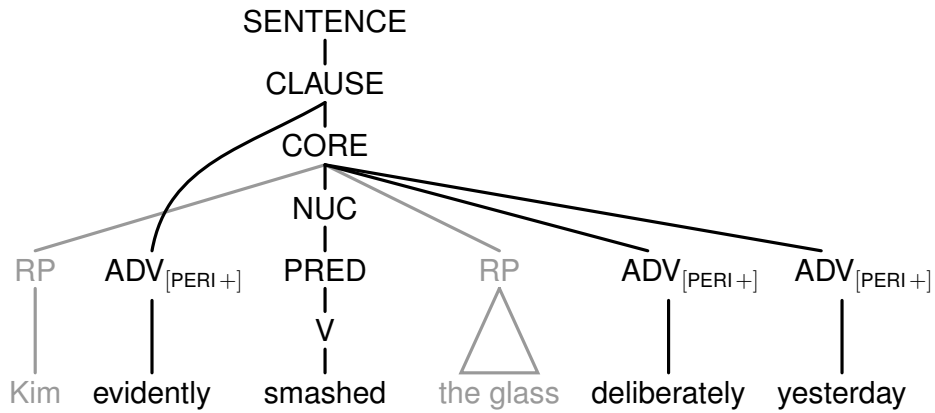


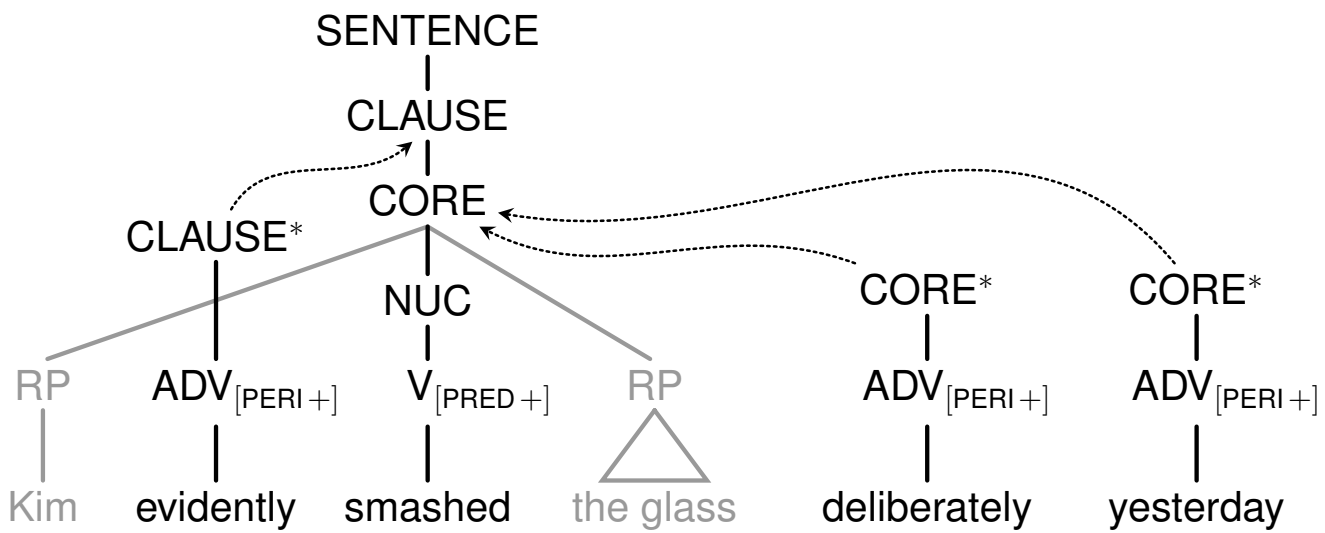
Mode of composition II: (sister) adjunction





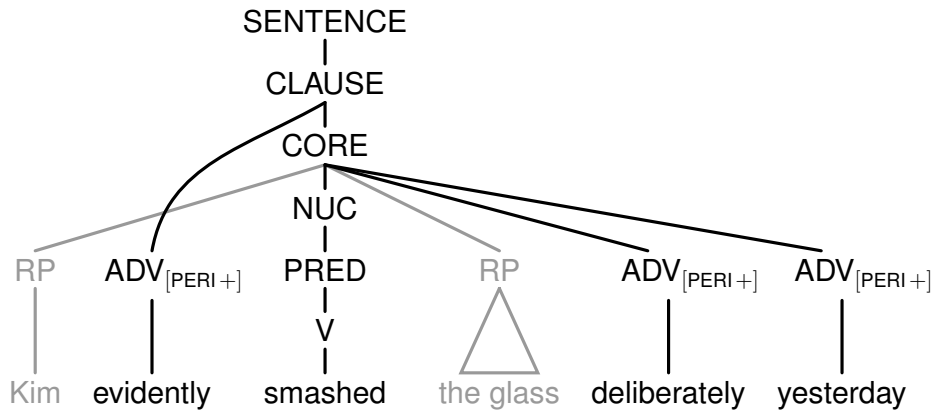
Mode of composition II: (sister) adjunction

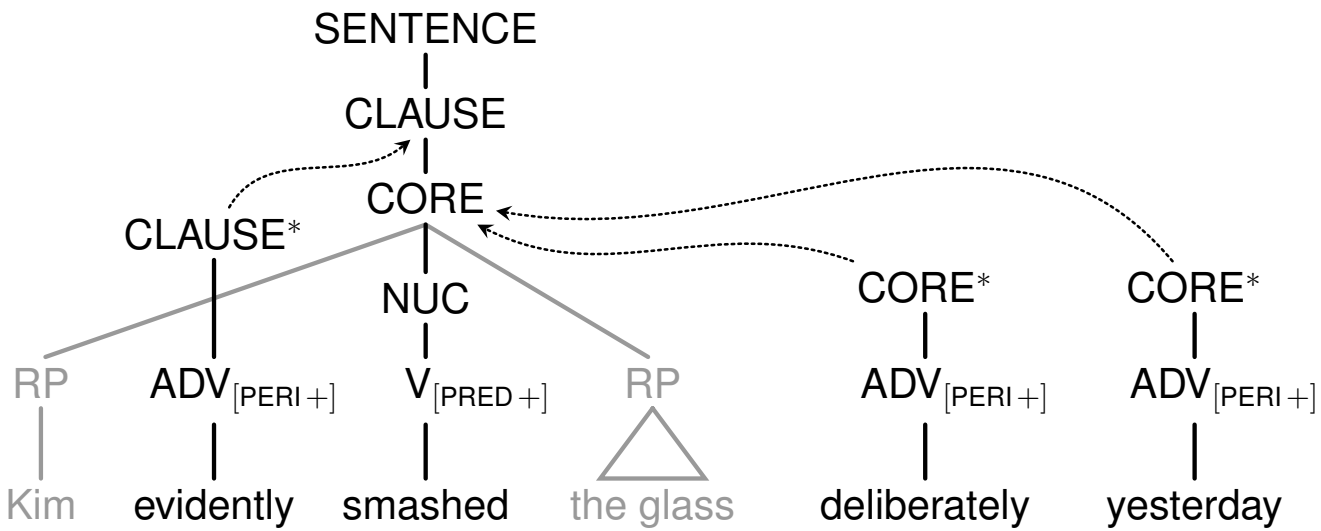




Syntactic composition

Mode of composition II: (sister) adjunction



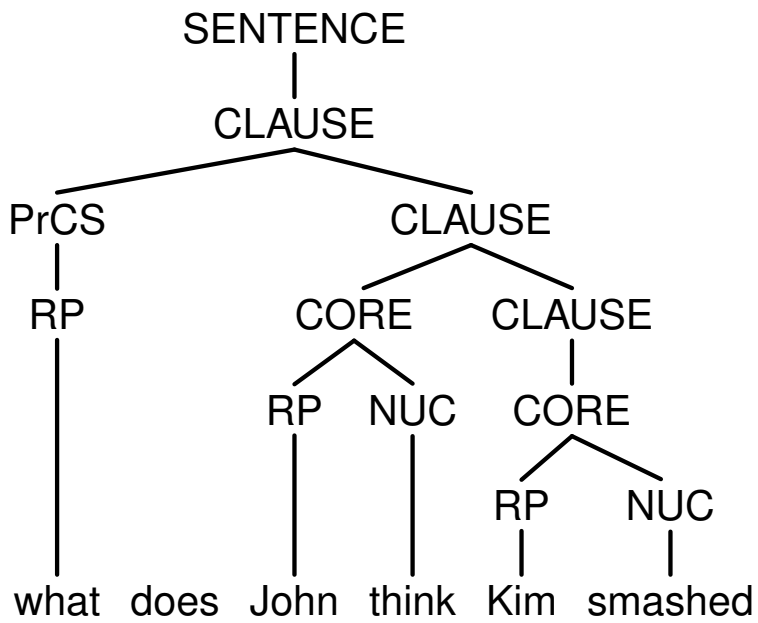


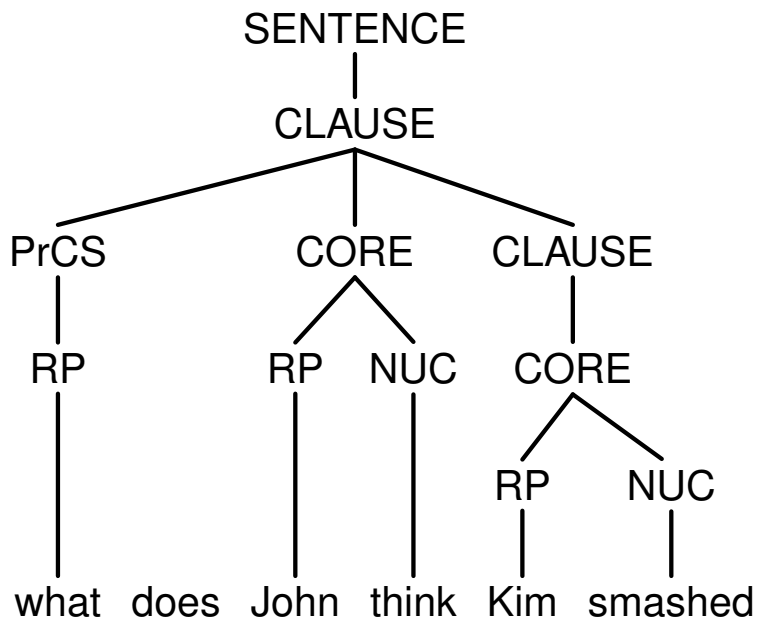
Issue: Crossing branches (more about this later)

Wh-extraction

(1) **What** does John think **Kim smashed**?

Possible analyses of (1):



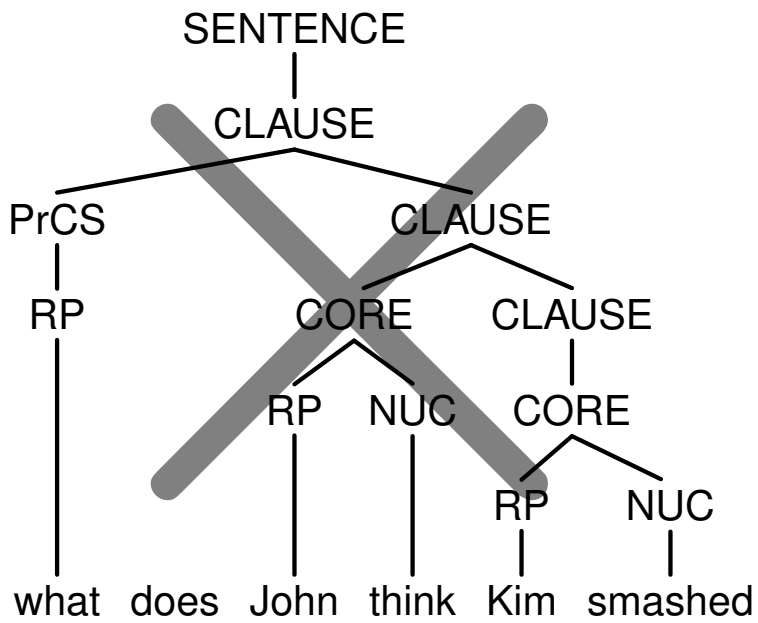


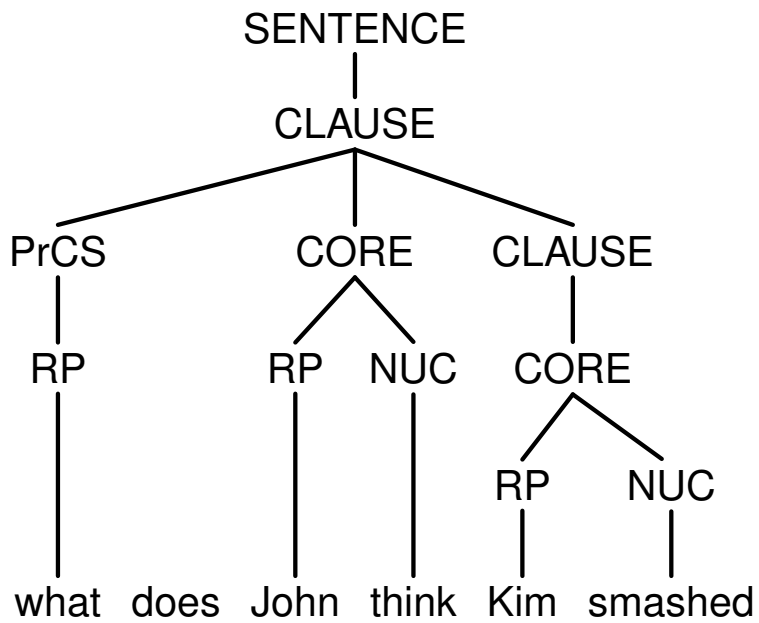
Syntactic composition

Wh-extraction

(1) **What** does John think **Kim smashed**?

Possible analyses of (1):



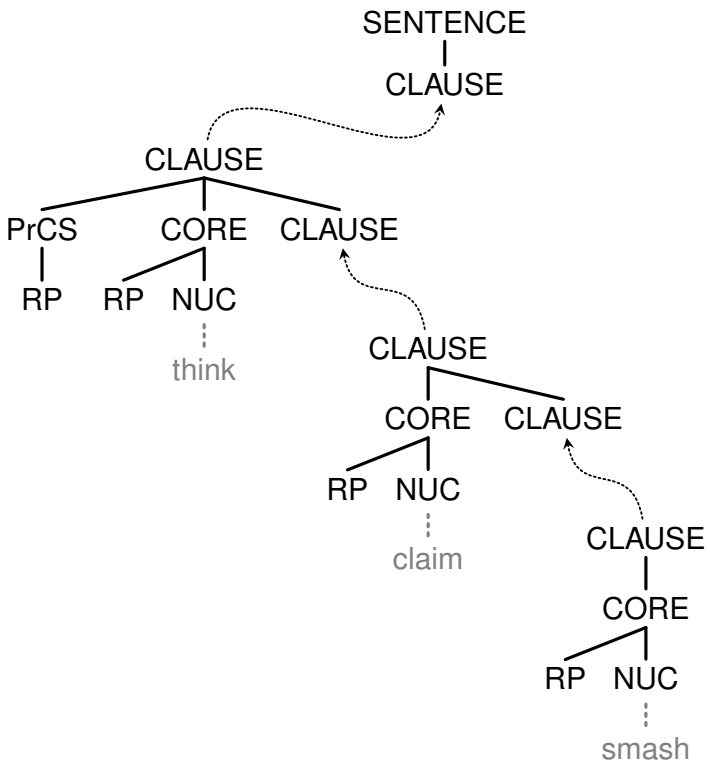


Syntactic composition

Wh-extraction

(2) **What** does John think Mary claimed **Kim smashed**?

Compositional derivation of (2):

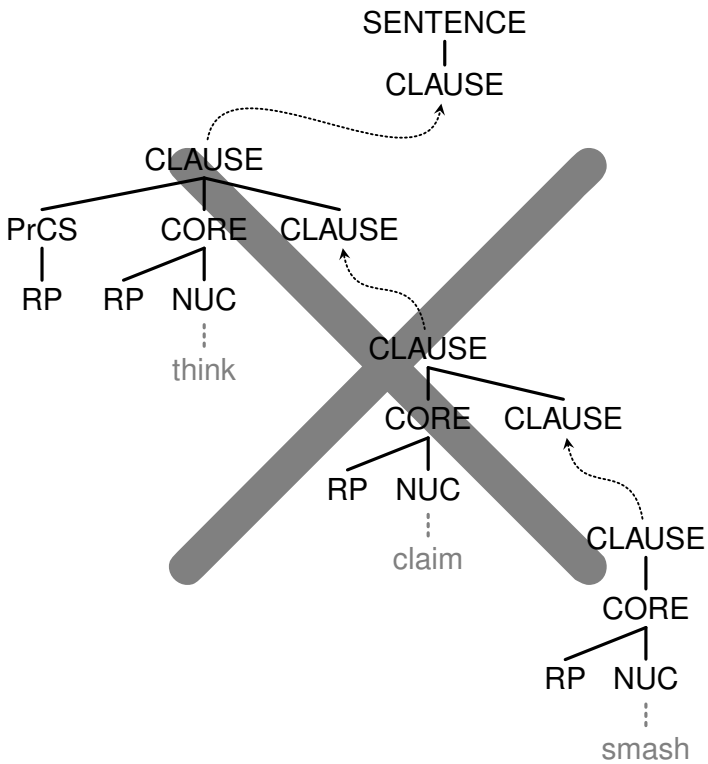


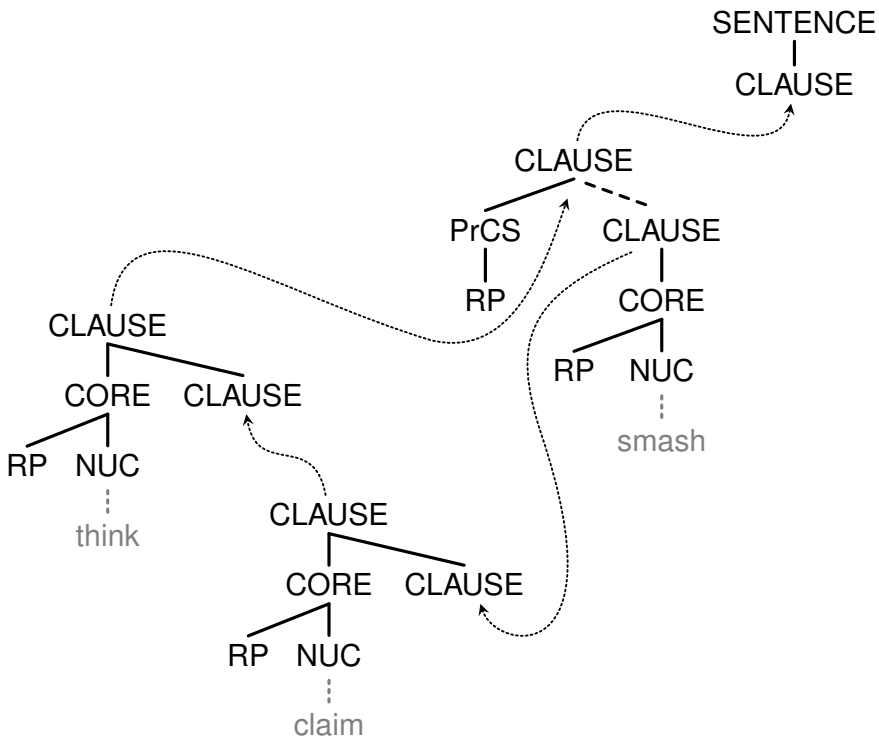
Syntactic composition

Wh-extraction

(2) **What** does John think Mary claimed **Kim smashed**?

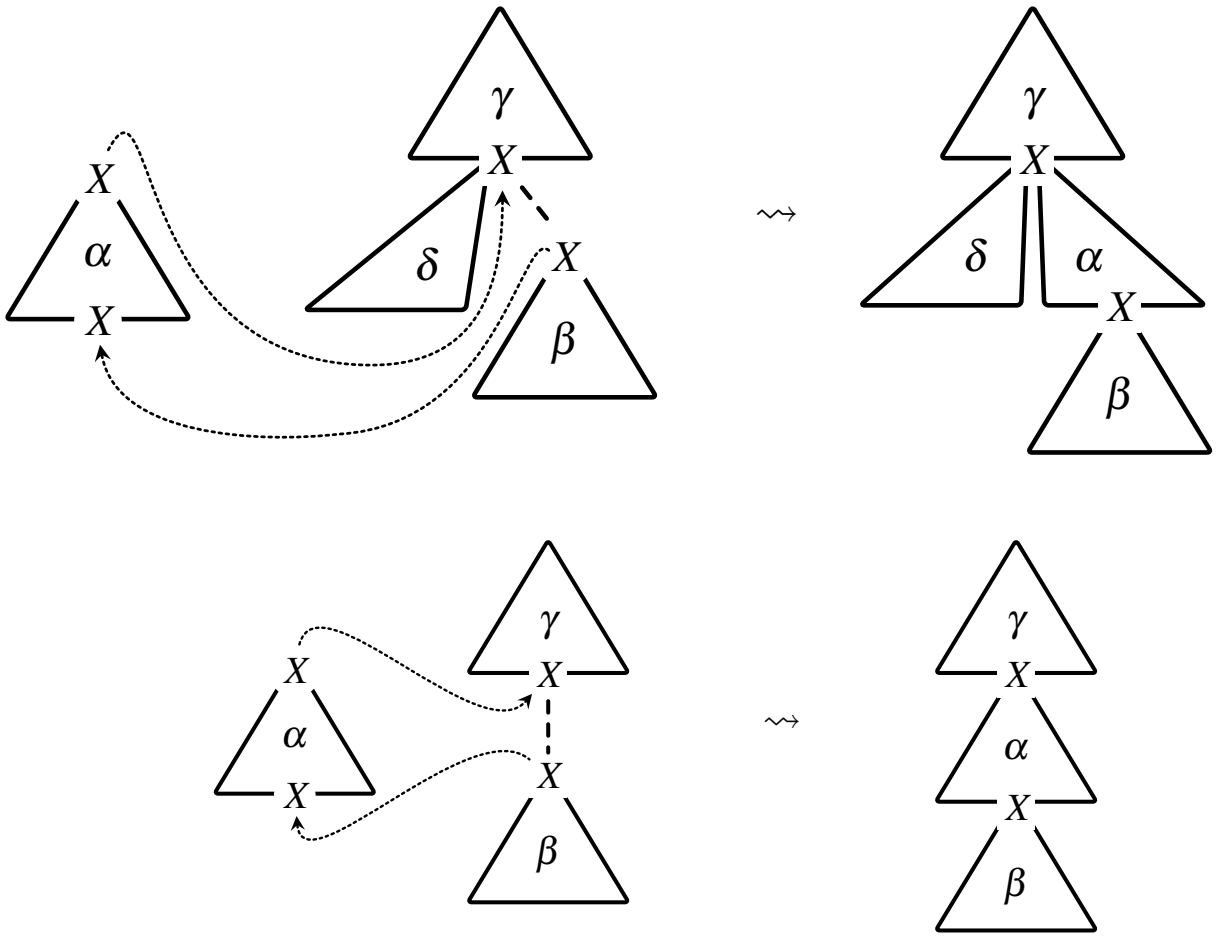
Compositional derivation of (2):





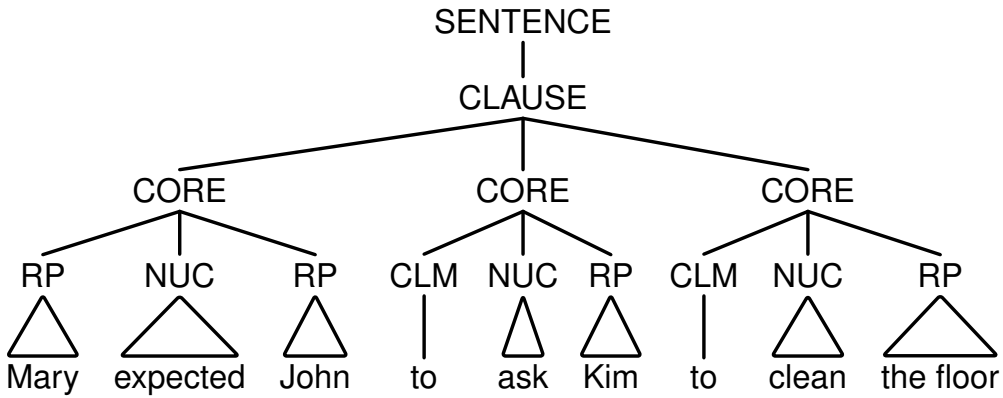
Syntactic composition

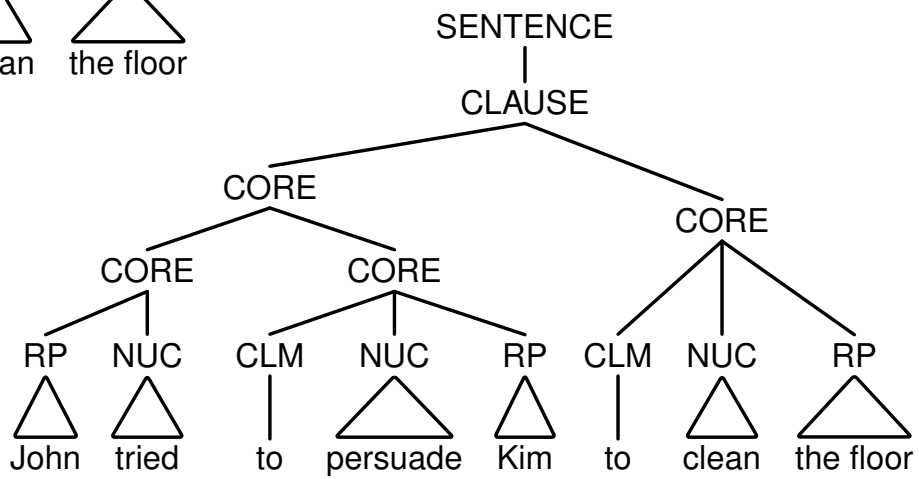
Mode of composition III: wrapping (substitution) (special versions)

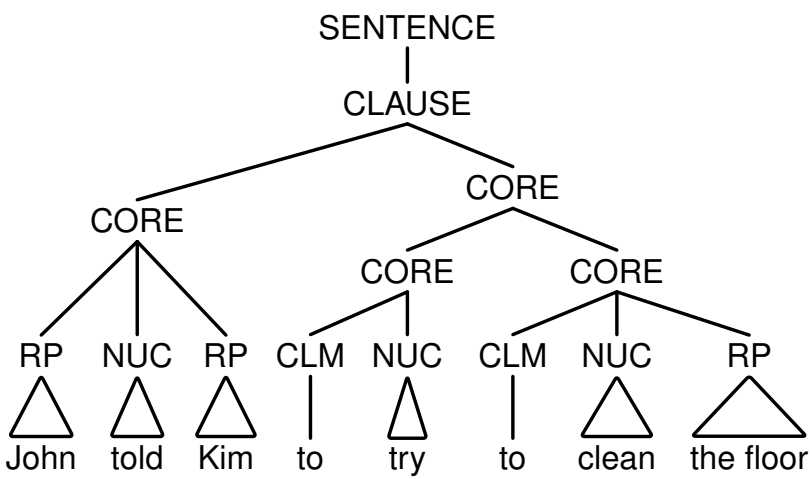


Syntactic composition

Control and matrix coding ($\approx$ raising)

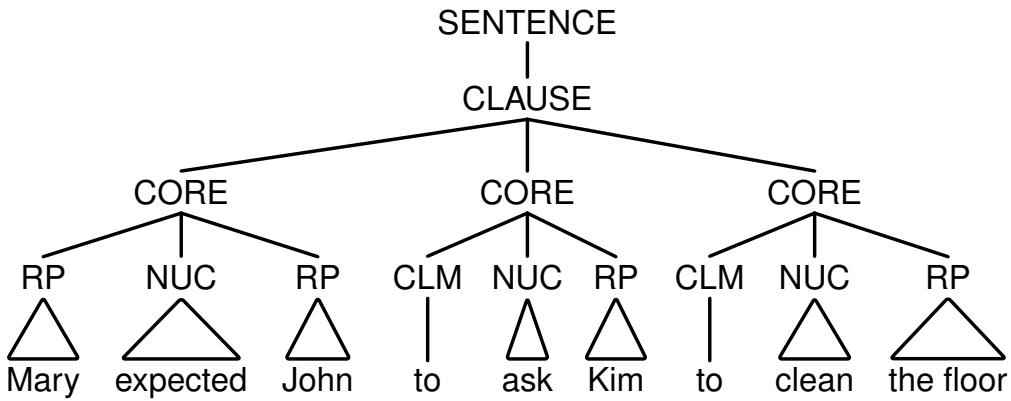


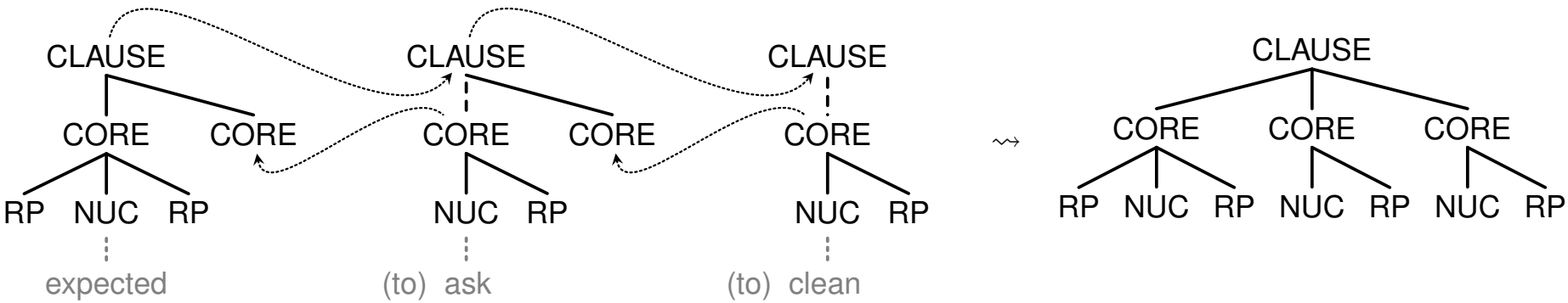




Syntactic composition

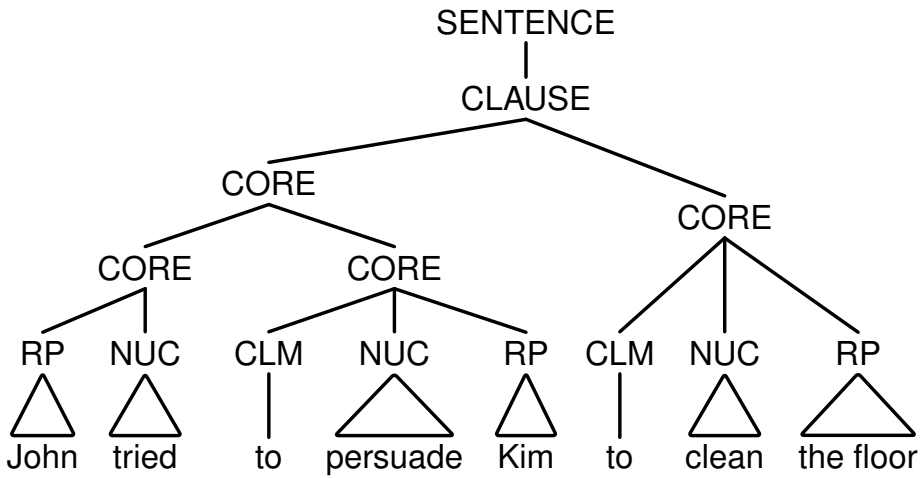
Control and matrix coding ($\approx$ raising)

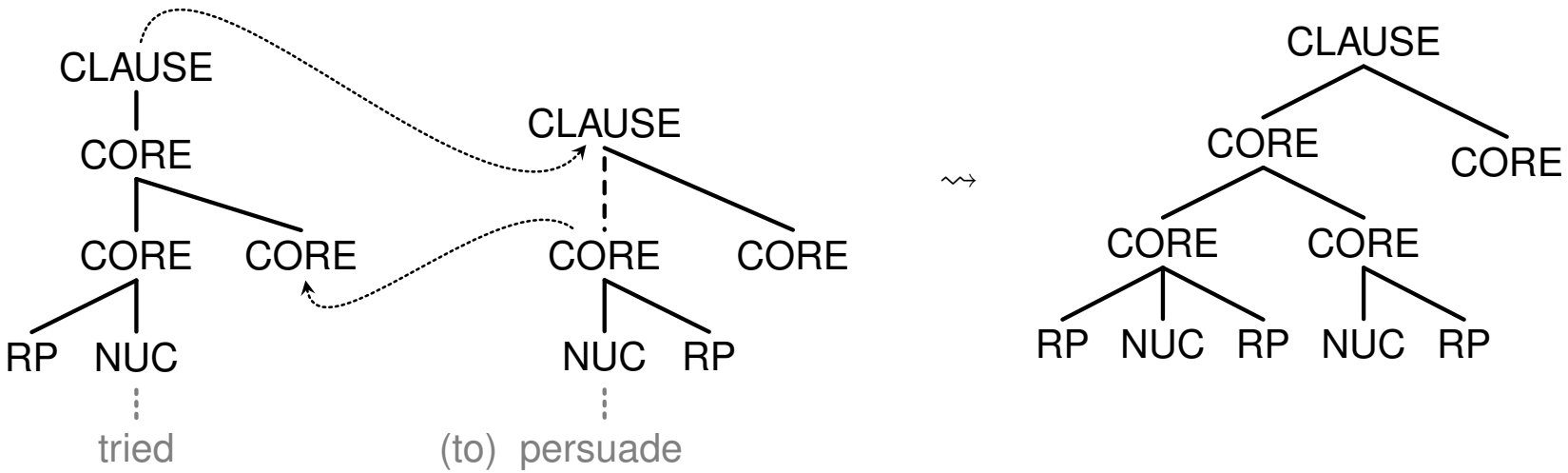




Syntactic composition

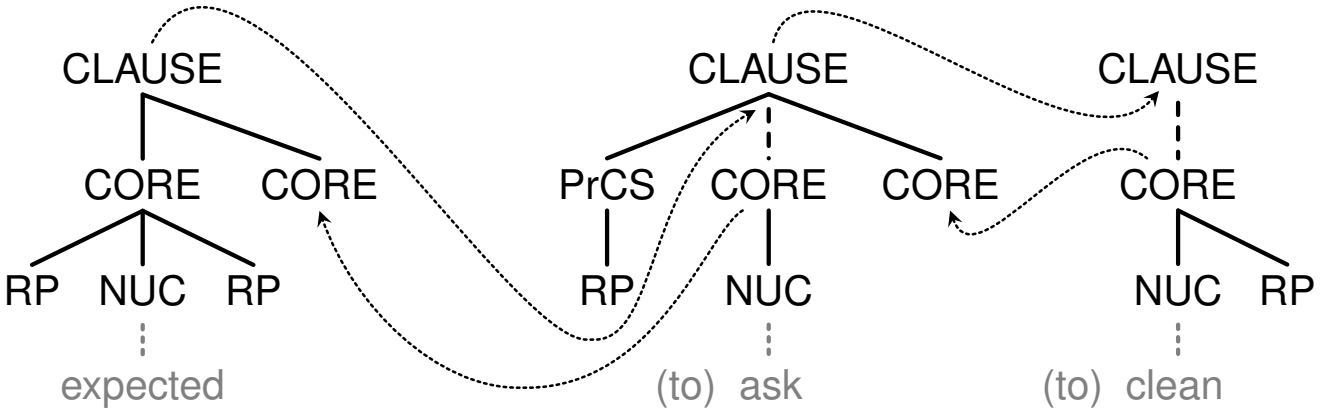
Control and matrix coding ($\approx$ raising)

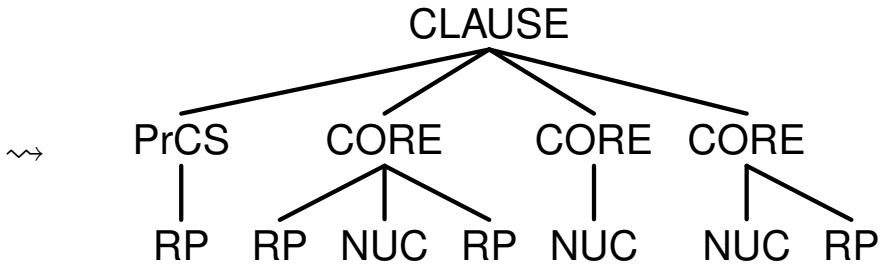




Control, matrix coding & wh-extraction

(3) **Whom** did Mary expect John to **ask** to clean the floor?

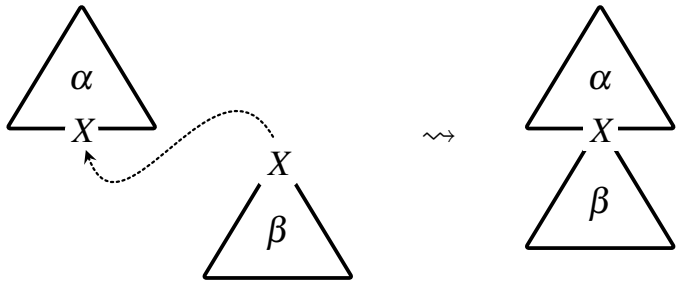




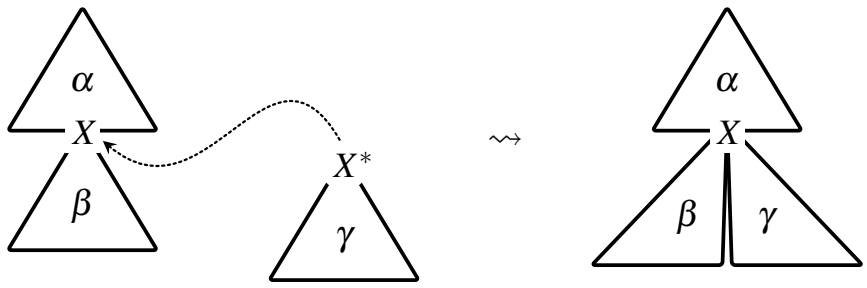
Syntactic composition

Modes of composition ($\rightsquigarrow$ **Tree Wrapping Grammar**; **TWG**)

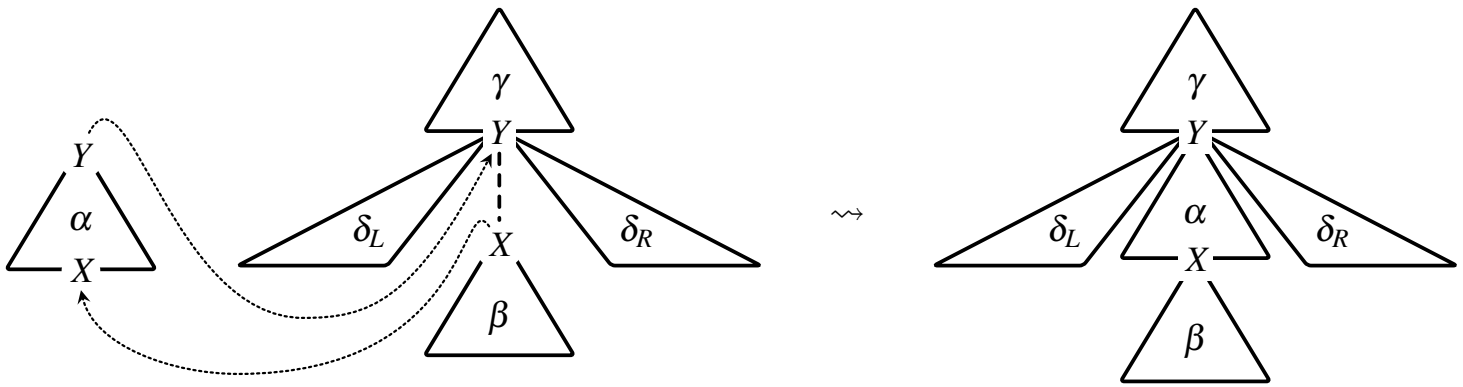
I. Simple substitution



II. Adjunction

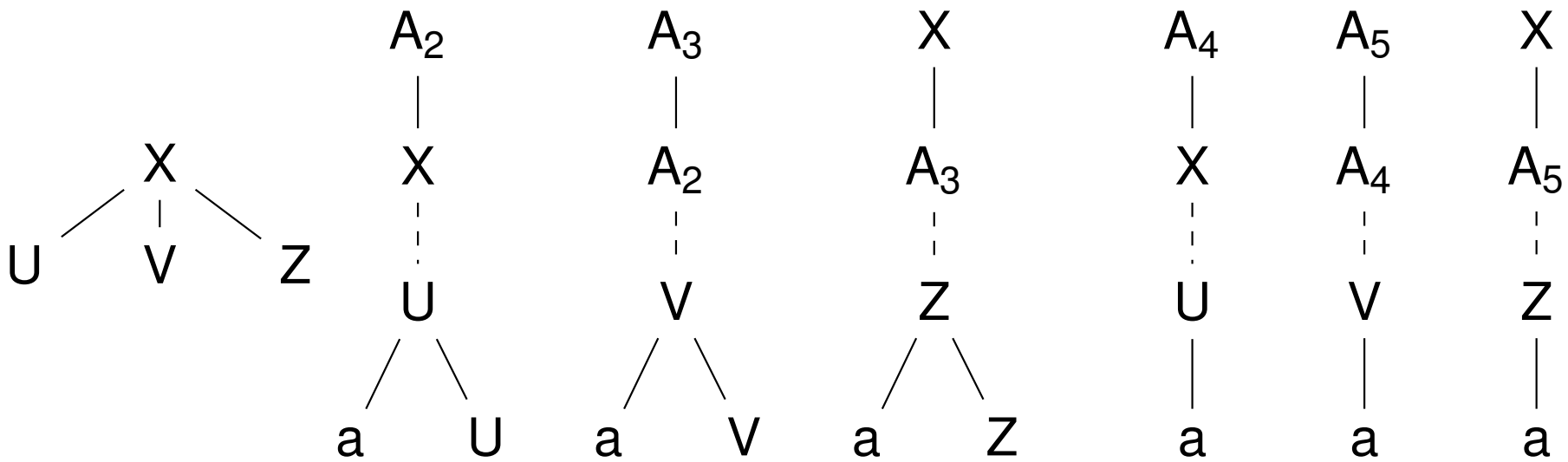


III. Wrapping substitution



Formal properties of TWGs

Example: TWG for $\{w^3 \mid w \in \{a, b\}^+\}$.



(+ same trees for b and $B_1, \ldots, B_5$ resp.)

k-TWG

Idea of k-TWG: limit the number of times a node can be part of a wrapping spine to k.

k-TWG

Idea of k-TWG: limit the number of times a node can be part of a wrapping spine to k.

We define the wrapping decoration of a specific derivation of some tree γ_d as the following set of node pairs $W(\gamma_d)$:

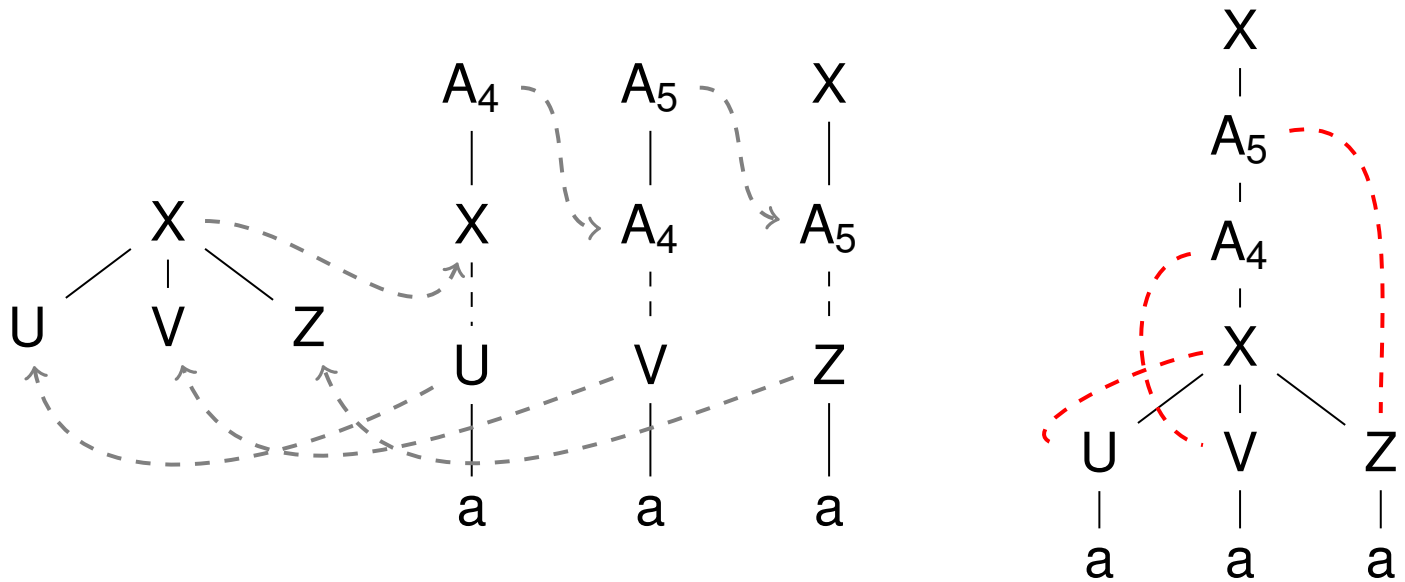
- In every wrapping substitution step with r the root and v the substitution node in the target tree, $\langle r, v \rangle \in W(\gamma_d)$.
- Nothing else is in $W(\gamma_d)$.

k-TWG

Idea of k-TWG: limit the number of times a node can be part of a wrapping spine to k.

We define the wrapping decoration of a specific derivation of some tree γ_d as the following set of node pairs $W(\gamma_d)$:

- In every wrapping substitution step with r the root and v the substitution node in the target tree, $\langle r, v \rangle \in W(\gamma_d)$.
- Nothing else is in $W(\gamma_d)$.

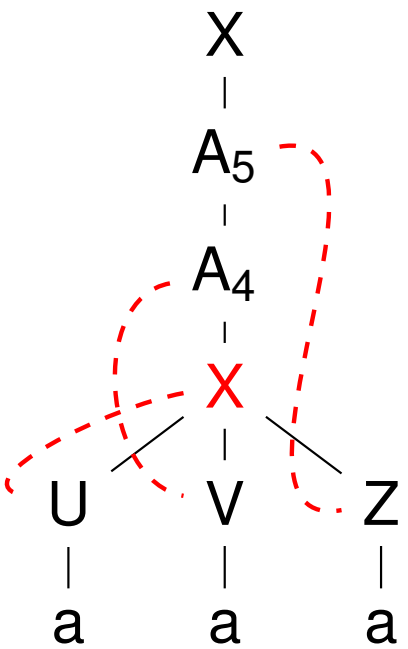


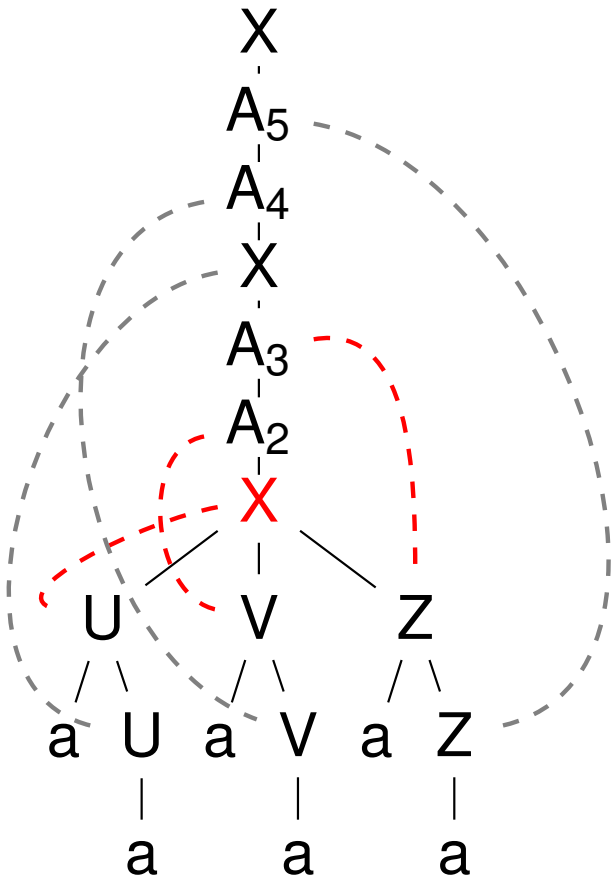
k-TWG

- For every node in a derived tree, the **gap degree** k gives the number of wrapping spines (dominance edges) stretching across that node (with respect to a specific derivation).
- If two such edges are nested, only the innermost counts.
- The maximal gap degree of the nodes gives the **wrapping degree** of the derivation.
- The minimal wrapping degree of all derivations for a given derived tree gives the wrapping degree of the derived tree.
- The k**-tree language** of a TWG is then the set of all its derived trees with wrapping degree $\leq k$.
- The tree language of a k**-TWG** G is defined as the k-tree language of the TWG G.

k-TWG

Example: TWG for $\{w^3 \mid w \in \{a, b\}^+\}$, derivations:





(only the red dominance edges count $\Rightarrow$ gap/wrapping degree $k = 3$)

k-TWG

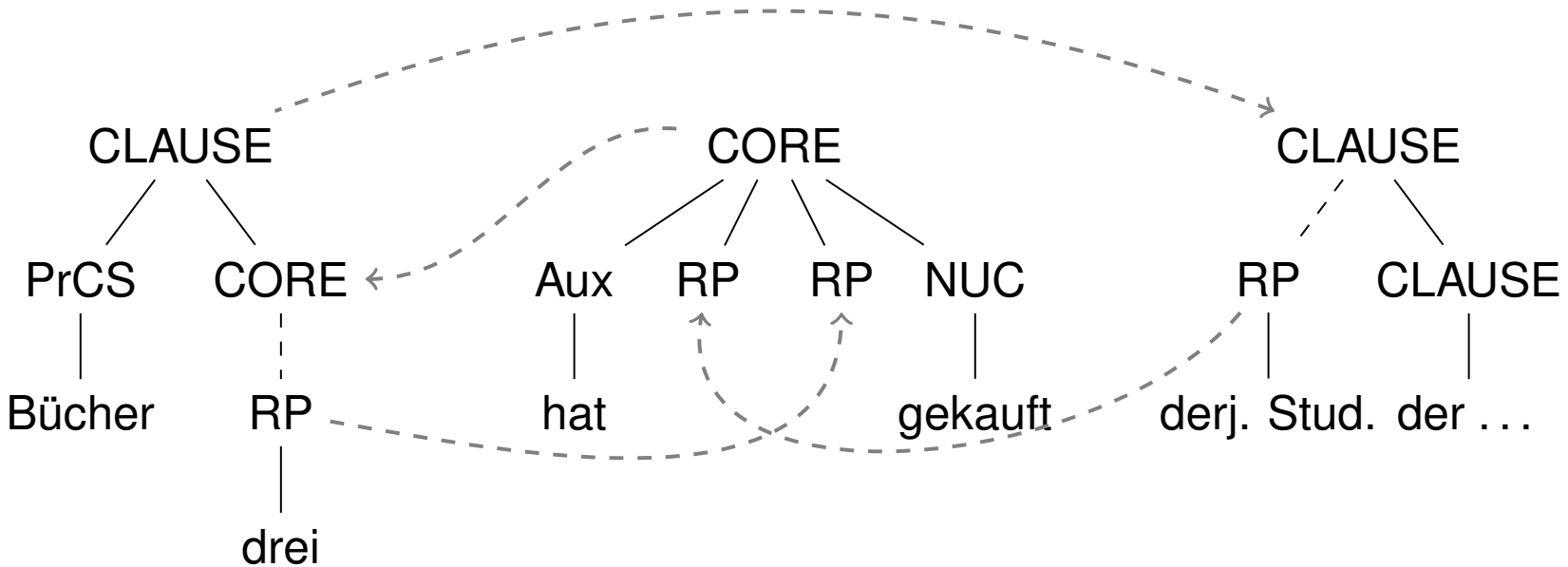
A $k > 1$ allows extraction out of several arguments

(4) Bücher hat derjenige Student drei gekauft der am meisten Geld
books has that student three bought who the most money
hatte
had
'the student with the most money bought three books'
(from Chen-Main & Joshi, 2012)

k-TWG

A $k > 1$ allows extraction out of several arguments

(4) Bücher hat derjenige Student drei gekauft der am meisten Geld
books has that student three bought who the most money
hatte
had
'the student with the most money bought three books'
(from Chen-Main & Joshi, 2012)



k-TWG and simple CFTG of rank k

- For every k-TWG, a simple Context-Free Tree Grammar (CFTG) of rank k can be constructed (Kallmeyer, 2016)
- This, in turn, is equivalent to a well-nested Linear Context-Free Rewriting System (LCFRS) of fan-out $k + 1$.
- Consequently, k-TWGs are in particular mildly context-sensitive.

k-TWG and simple CFTG of rank k

To show: for every k-TWG one can construct an equivalent simple context-free tree grammar of rank k.

k-TWG and simple CFTG of rank k

To show: for every k-TWG one can construct an equivalent simple context-free tree grammar of rank k.

A **simple context-free tree grammar** (CFTG, Rounds, 1970; Engelfriet & Schmidt, 1977) is a quadruple $G = \langle N, \Sigma, P, S \rangle$, where

1. N is a ranked alphabet of non-terminals,
2. Σ an unranked alphabet of terminals,
3. $S \in N$ is of rank 0, and
4. P is a finite set of productions of the form

$$A x_1 \dots x_n \to t[x_1, \dots, x_n]$$

where $A \in N^{(n)}$ and $t[x_1, \dots, x_n]$ is a tree over $N \cup \Sigma \cup \{x_1, \dots, x_n\}$ with each of the $x_1, \dots, x_n$ occurring exactly once as a leaf label.

The **rank** of G is the maximal rank of its non-terminals.

k-TWG and simple CFTG of rank k

Example: simple CFTG for $\{w^3 \mid w \in \{a,b\}^+\}$:

$N^0 = \{\overline{S}\}, N^{(3)} = \{\overline{X}\}, \Sigma = \{a,b,A\}$, S the start symbol.

P contains the following productions:

$\overline{S} \to \overline{X}aaa \mid \overline{X}bbb$
$\overline{X}x_1x_2x_2 \to \overline{X}(Aax_1)(Aax_2)(Aax_3) \mid \overline{X}(Abx_1)(Abx_2)(Abx_3) \mid Ax_1x_2x_3$

k-TWG and simple CFTG of rank k

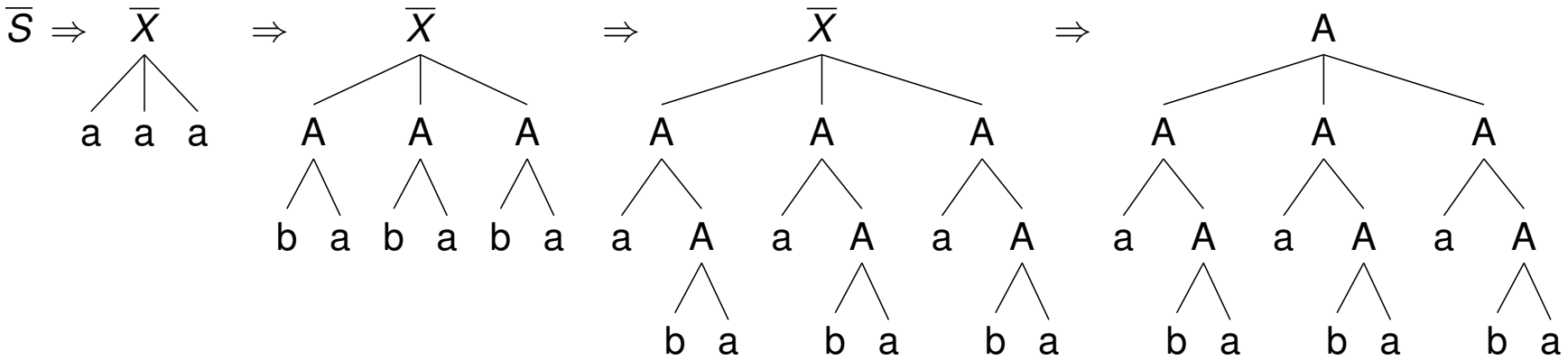
Example: simple CFTG for $\{w^3 \mid w \in \{a,b\}^+\}$:

$N^0 = \{\overline{S}\}, N^{(3)} = \{\overline{X}\}, \Sigma = \{a,b,A\}$, S the start symbol.

P contains the following productions:

$\overline{S} \to \overline{X}aaa \mid \overline{X}bbb$

$\overline{X}x_1x_2x_2 \to \overline{X}(Aax_1)(Aax_2)(Aax_3) \mid \overline{X}(Abx_1)(Abx_2)(Abx_3) \mid Ax_1x_2x_3$



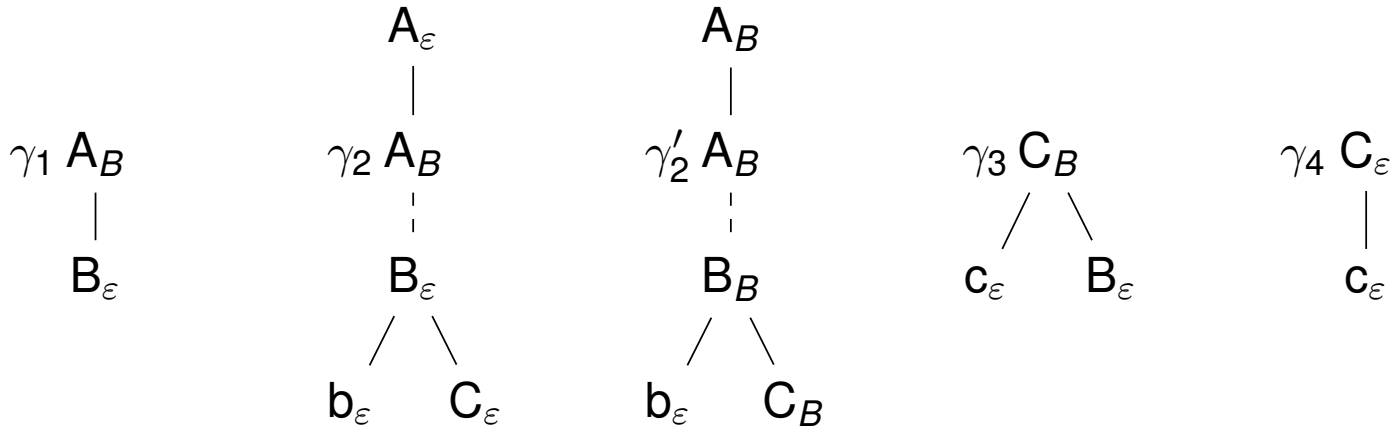
k-TWG and simple CFTG of rank k

Idea of the construction:

- The CFTG terminals comprise the terminals and non-terminals from the TWG.
- The CFTG non-terminals have the form $[A, A_1 A_2 \ldots A_n]$ where
  - A is the root category of the tree this nonterminal expands to and
  - $A_1 A_2 \ldots A_n$ are the categories of pending gaps from wrappings that stretch across this tree.
- I.e., the CFTG non-terminals encode possible gap sets of nodes in specific derivations.

k-TWG and simple CFTG of rank k

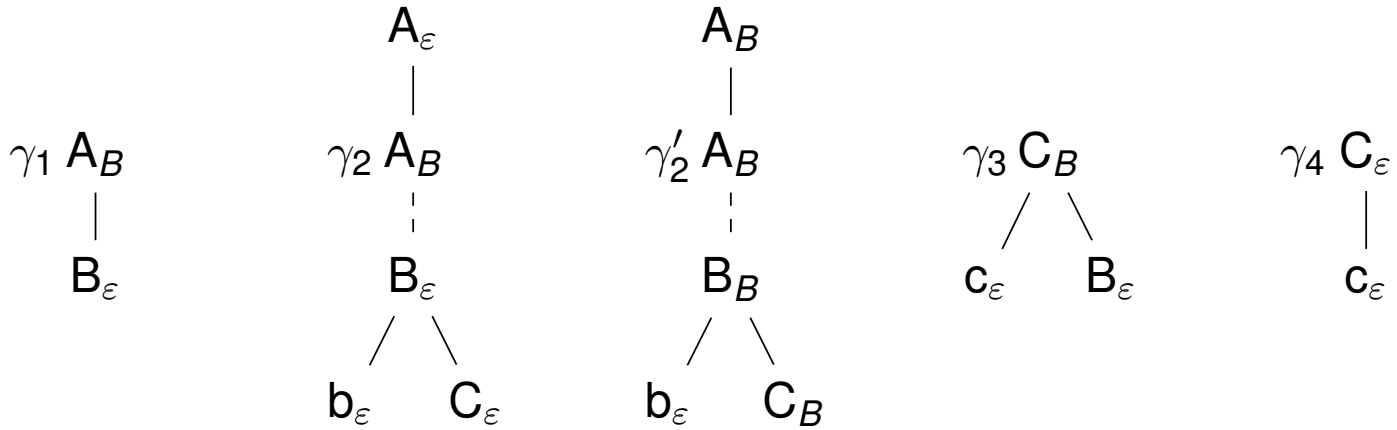
TWG for $\{(bc)^n \mid n \geq 1\} \cup \{c\}$:



(Decoration with possible gap categories.)

k-TWG and simple CFTG of rank k

TWG for $\{(bc)^n \mid n \geq 1\} \cup \{c\}$:



(Decoration with possible gap categories.)

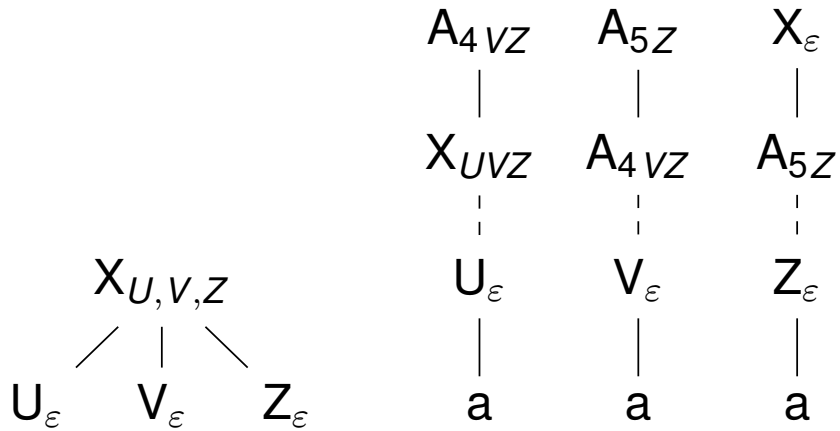
Equivalent simple CFTG:

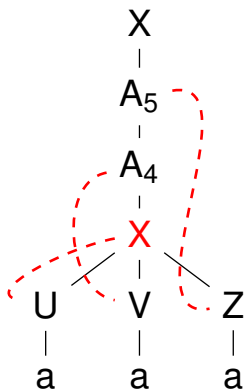
$$
\begin{array}{ll}
 & S \to [A],\ S \to [C] \\
\gamma_1\colon & [A, B]x_1 \to Ax_1 \\
\gamma_2\colon & [A] \to A([A, B](Bb[C])) \\
\gamma_2'\colon & [A, B]x_1 \to A([A, B](Bb([C, B]x_1))) \\
\gamma_3\colon & [C, B]x_1 \to Ccx_1 \\
\gamma_4\colon & [C] \to Cc
\end{array}
$$

k-TWG and simple CFTG of rank k

Example: 3-TWG for $\{w^3 \mid w \in \{a, b\}^+\}$

Guess possible gap sequences and construct CFTG rules accordingly:





Corresponding CFTG productions:

$S \to [X]$

$[X, UVZ]x_1x_2x_3 \to Xx_1x_2x_3$

$[A_4, VZ](x_2, x_3) \to A_4([X, UVZ](Ua, x_2, x_3))$

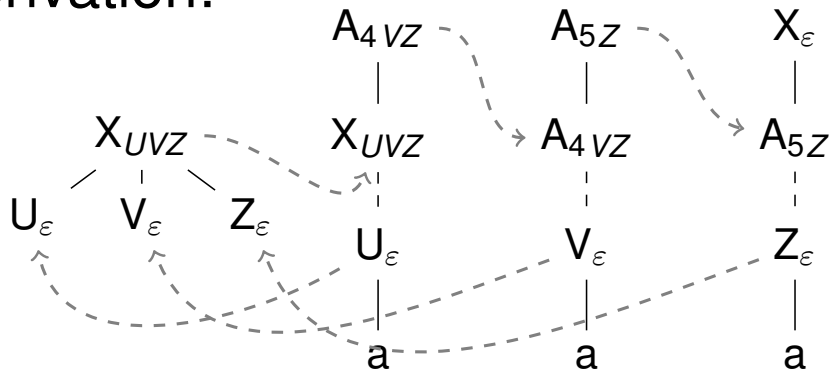
$[A_5, Z](x_3) \to A_5([A_4, VZ](Va, x_3))$

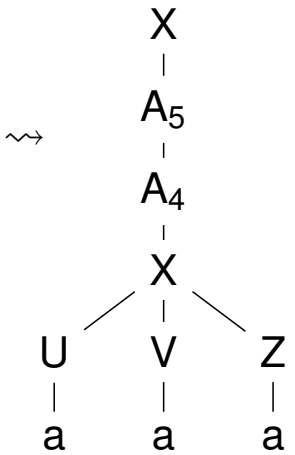
$[X] \to X([A_5, Z](Za))$

k-TWG and simple CFTG of rank k

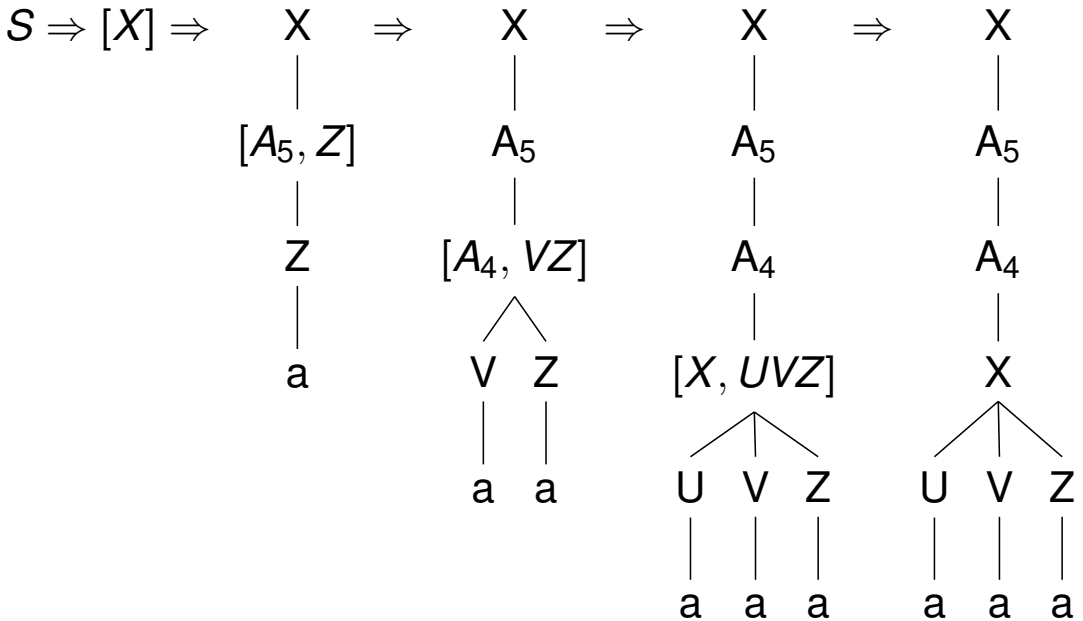
Example continued

TWG derivation:





CFTG derivation:



Syntax-semantics interface

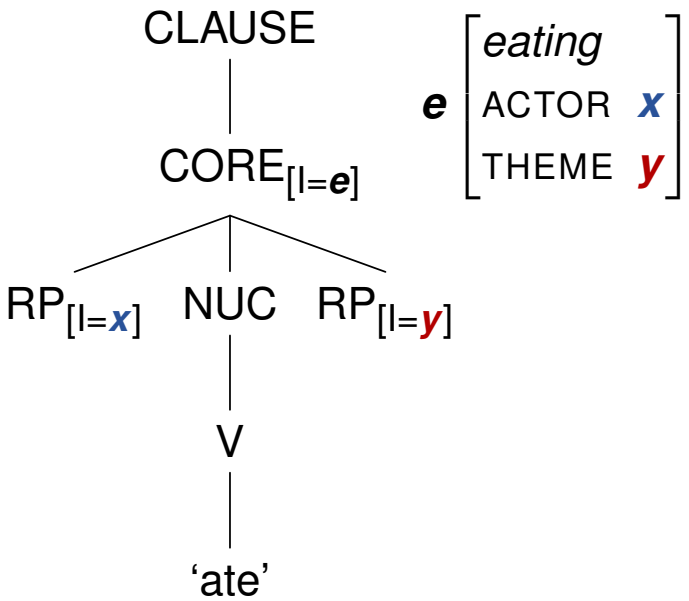
Example

(5) Adam ate an apple.

Syntax-semantics interface

Example

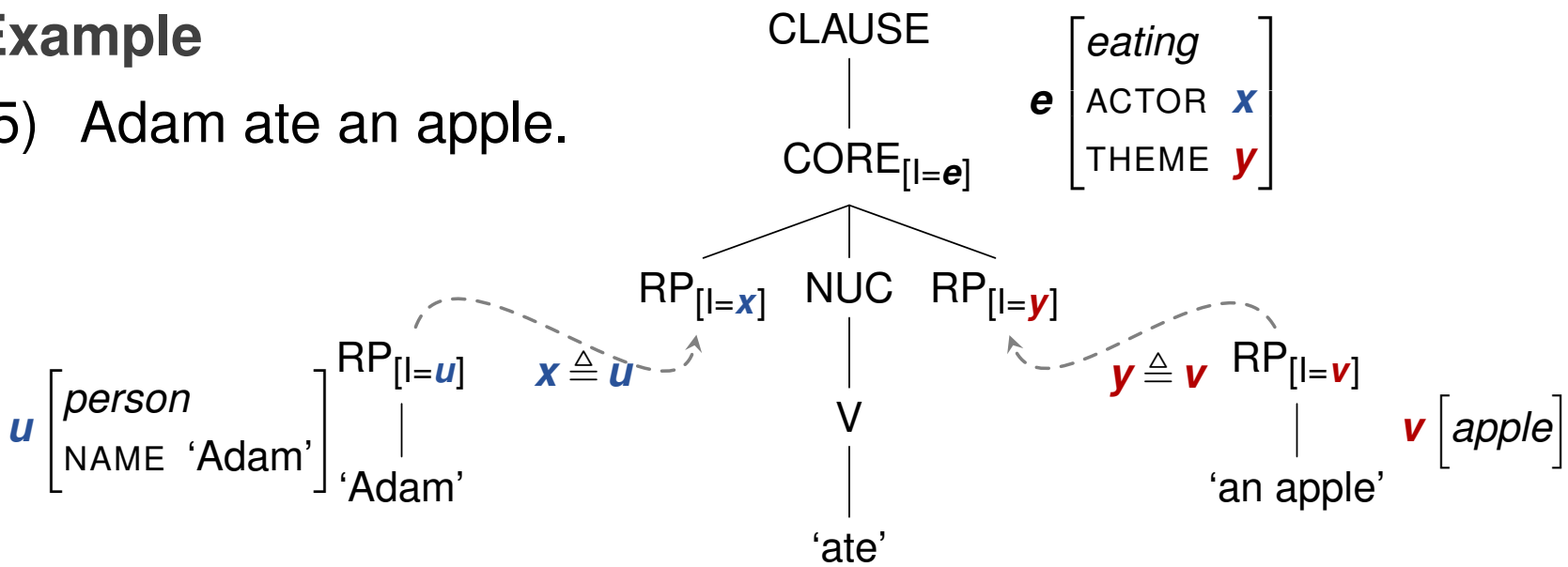
(5) Adam ate an apple.



Syntax-semantics interface

Example

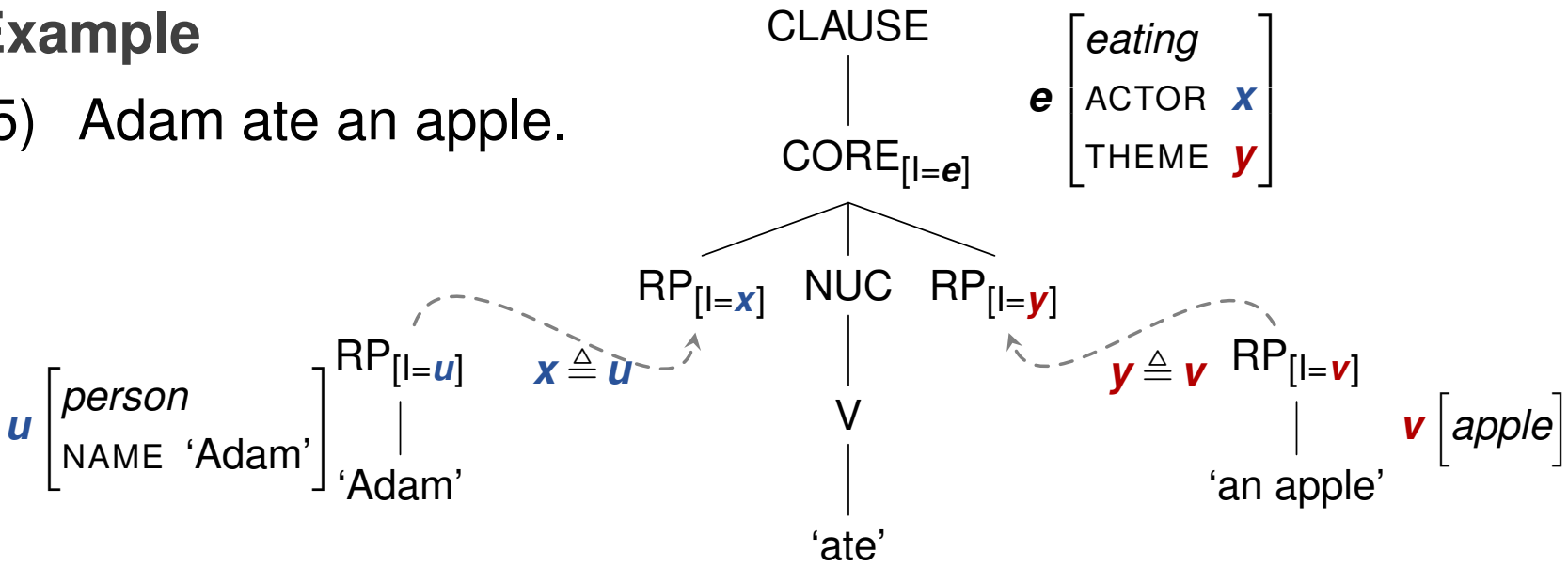
(5) Adam ate an apple.

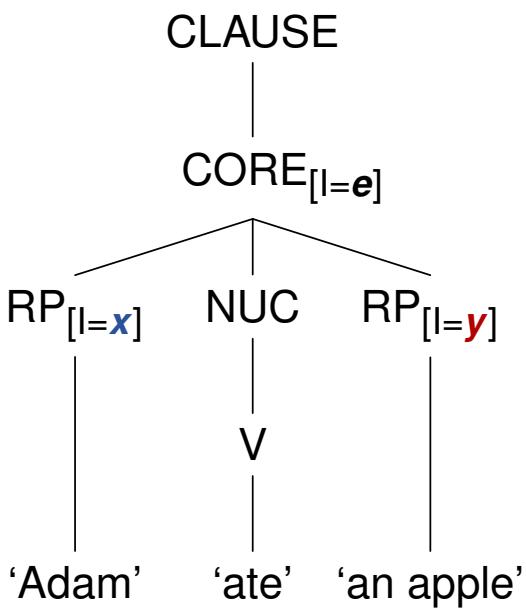


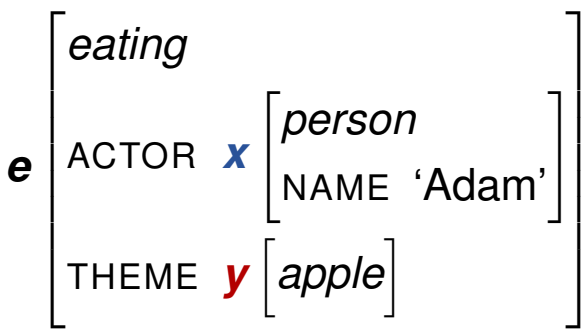
Syntax-semantics interface

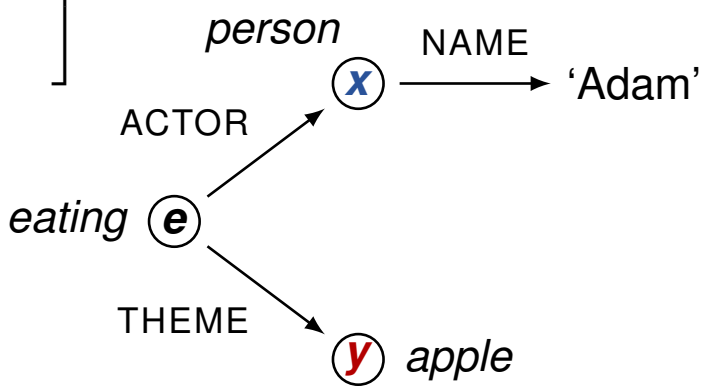
Example

(5) Adam ate an apple.









Syntax-semantics interface

Summary of the LTAG + frame semantics perspective on RRG:

- **Elementary construction**
  = elementary tree (full argument projection) + semantic frame
  + linking of frame node variables to interface features in the tree

Syntax-semantics interface

Summary of the LTAG + frame semantics perspective on RRG:

- **Elementary construction**
  = elementary tree (full argument projection) + semantic frame
  + linking of frame node variables to interface features in the tree

- "**Complicate locally**, **simplify globally**"
  1. A small set of (global) operations for syntactic composition
  2. Many linguistic regularities and generalizations are encoded in elementary constructions $\rightarrow$ decomposition in the **metagrammar**

Syntax-semantics interface

Summary of the LTAG + frame semantics perspective on RRG:

- **Elementary construction**
  = elementary tree (full argument projection) + semantic frame
  + linking of frame node variables to interface features in the tree

- "**Complicate locally**, **simplify globally**"
  1. A small set of (global) operations for syntactic composition
  2. Many linguistic regularities and generalizations are encoded in elementary constructions $\rightarrow$ decomposition in the **metagrammar**

- Special tree operations because of flat syntactic structures:
  (Wrapping) substitution and **sister adjunction**.

Syntax-semantics interface

Summary of the LTAG + frame semantics perspective on RRG:

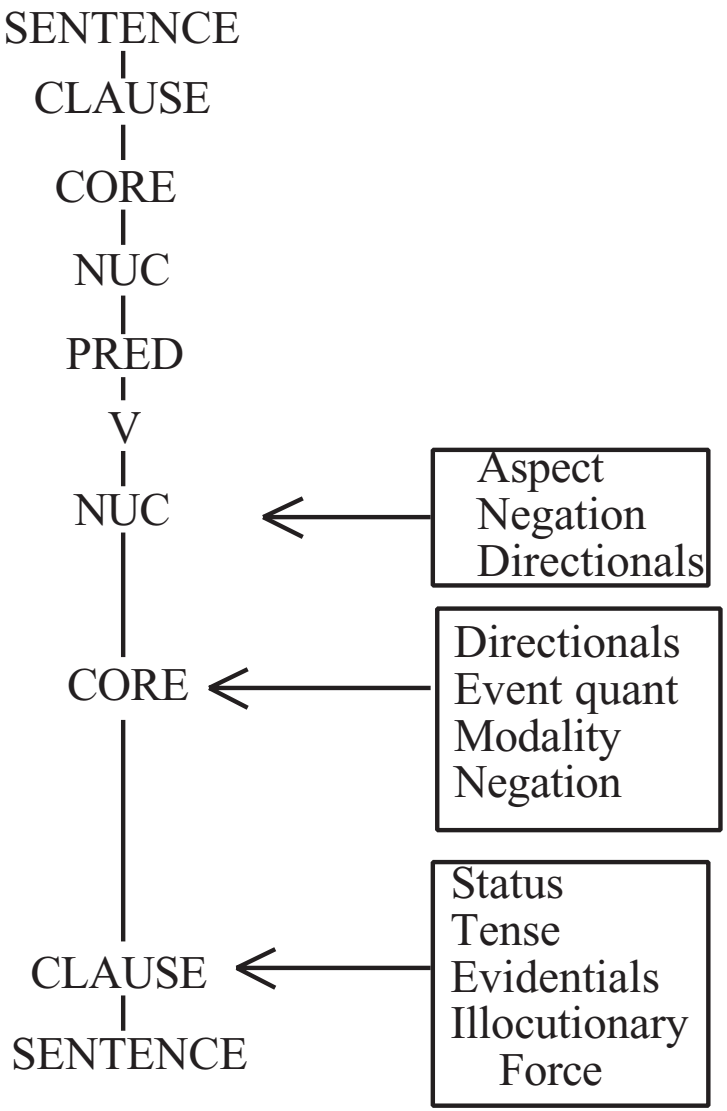
- **Elementary construction**
  = elementary tree (full argument projection) + semantic frame
  + linking of frame node variables to interface features in the tree

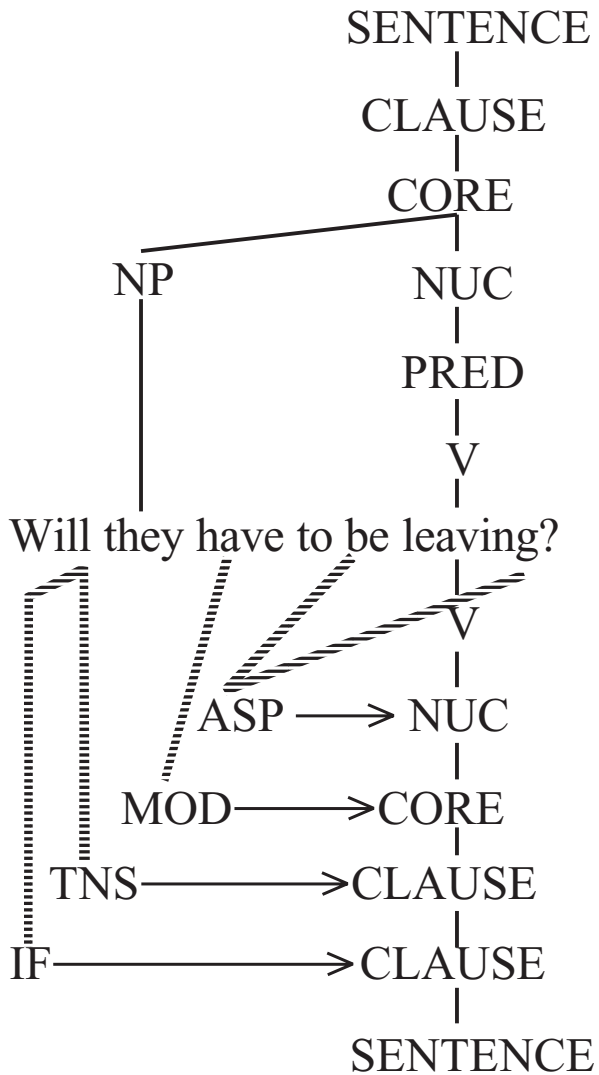
- "**Complicate locally**, **simplify globally**"
  1. A small set of (global) operations for syntactic composition
  2. Many linguistic regularities and generalizations are encoded in elementary constructions $\rightarrow$ decomposition in the **metagrammar**

- Special tree operations because of flat syntactic structures:
  (Wrapping) substitution and **sister adjunction**.

- Argument **linking rules as constraints** in the metagrammar.

Operator projection

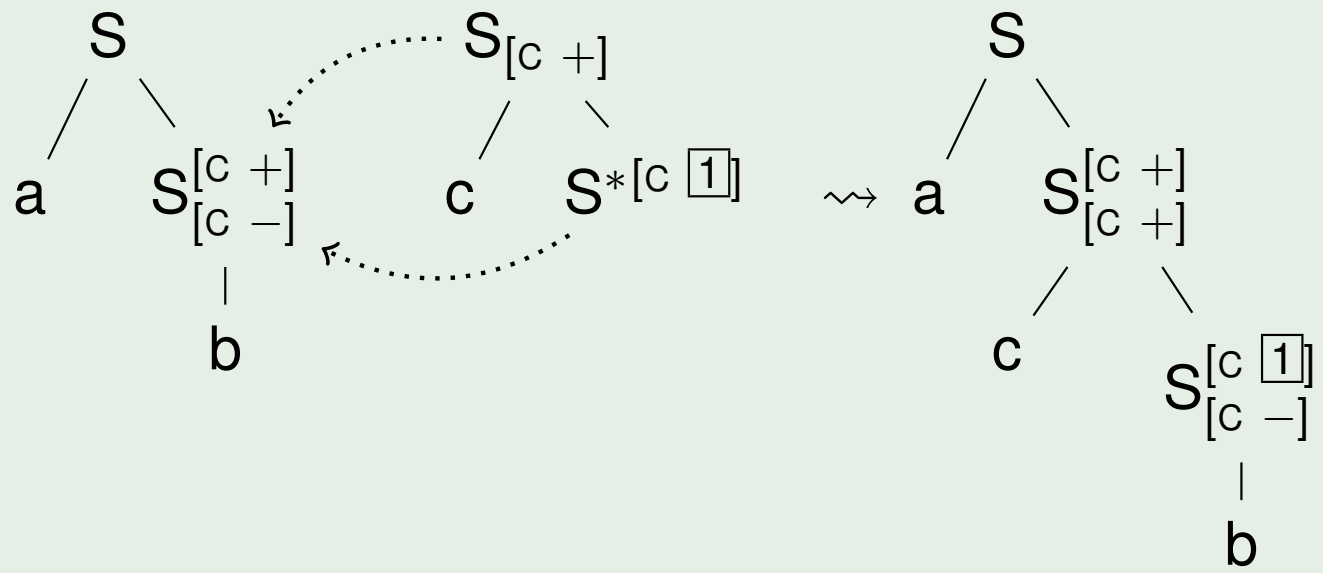




[Van Valin 2005: 12/14]

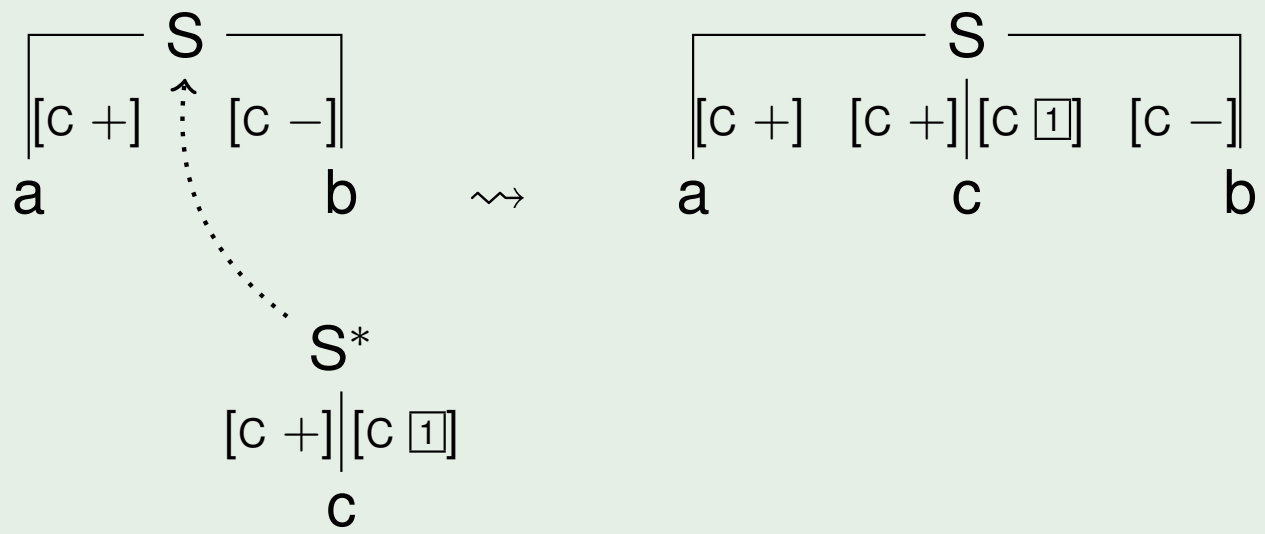
Adding features: FTWG

In TAG (mostly binary tree structures), we have top and bottom feature stuctures that can constrain adjunction.



Adding features: FTWG

In our flat structures with sister adjunction, we use left and right edge features to capture adjunction constraints.



Adding features: FTWG

Feature-based Tree Wrapping Grammar (FTWG)

- Finite set of untyped feature structures with structure sharing within elementary trees (just like TAG, Vijay-Shanker & Joshi, 1988).
- Nodes have a single feature structure while edges have a left one and a right one.
- In a sister adjunction, the feature structure of the root of the adjoined tree unifies with the one of the target node.
- In the final derived tree, the two feature structures between two neighbouring edges have to unify.

  Furthermore, features on the leftmost (resp. rightmost) edge percolate upwards, except if there is a substitution node, which blocks feature percolation.

Integrating operators

Each operator belongs to a certain level of RRG's layered structure:

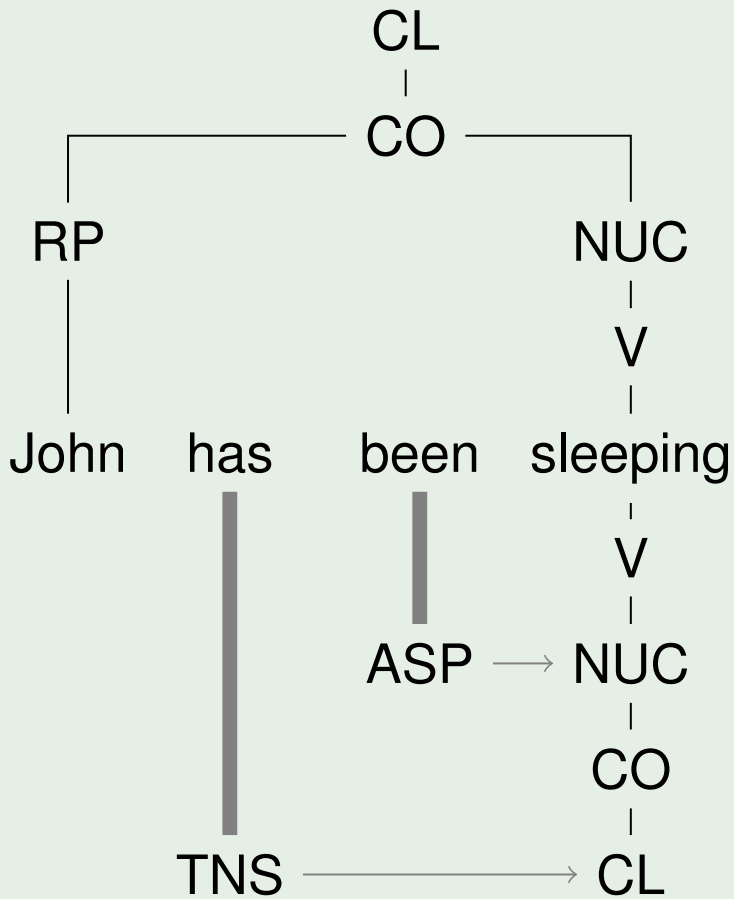
Layer	Nucleus	Core	Clause
Operators	Aspect	Directionals	Status
	Negation	Event quantification	Tense
	Directionals	Modality	Evidentials
		Negation	Illocutionary Force

The operator level explains

- the scope behavior: structurally higher operators take scope over lower ones
- surface order constraints: higher operators are further away from the nucleus of the structure.

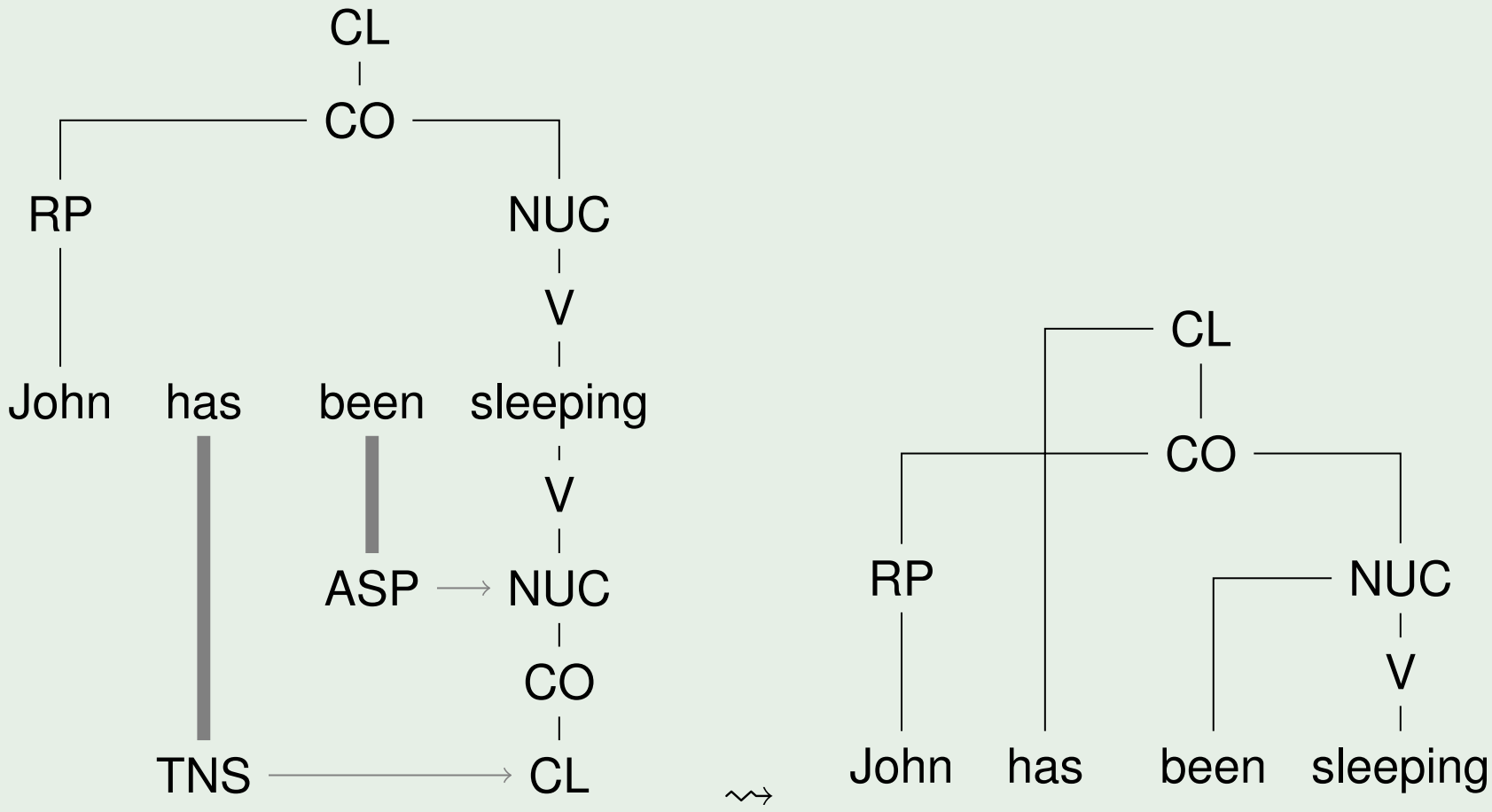
Integrating operators

Problem: constituent and operator structure are not completely parallel. An operator belonging to a specific layer can be surrounded by elements belonging to a lower layer in the constituent structure.



Integrating operators

Problem: constituent and operator structure are not completely parallel. An operator belonging to a specific layer can be surrounded by elements belonging to a lower layer in the constituent structure.

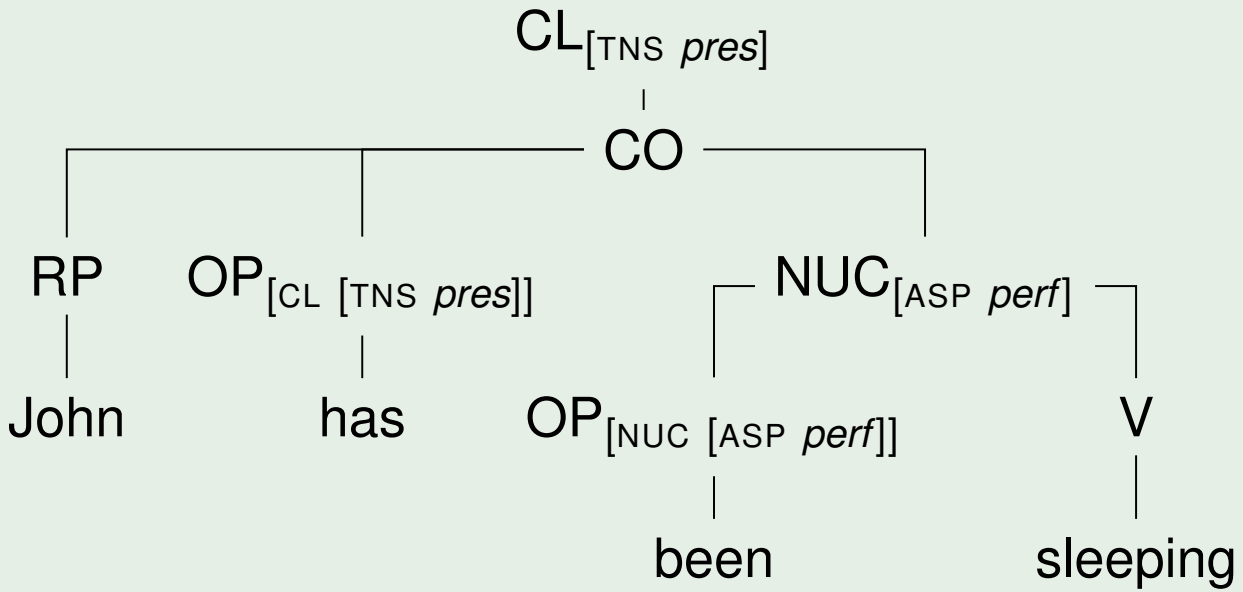


Integrating operators

The following holds:

- The hierarchical order of constituent and operator structure is the same.
- The existence of a layer in the operator projection requires that this layer also exists in the constituent structure.

We model the operator projection within the features while attaching the operators at their surface position.

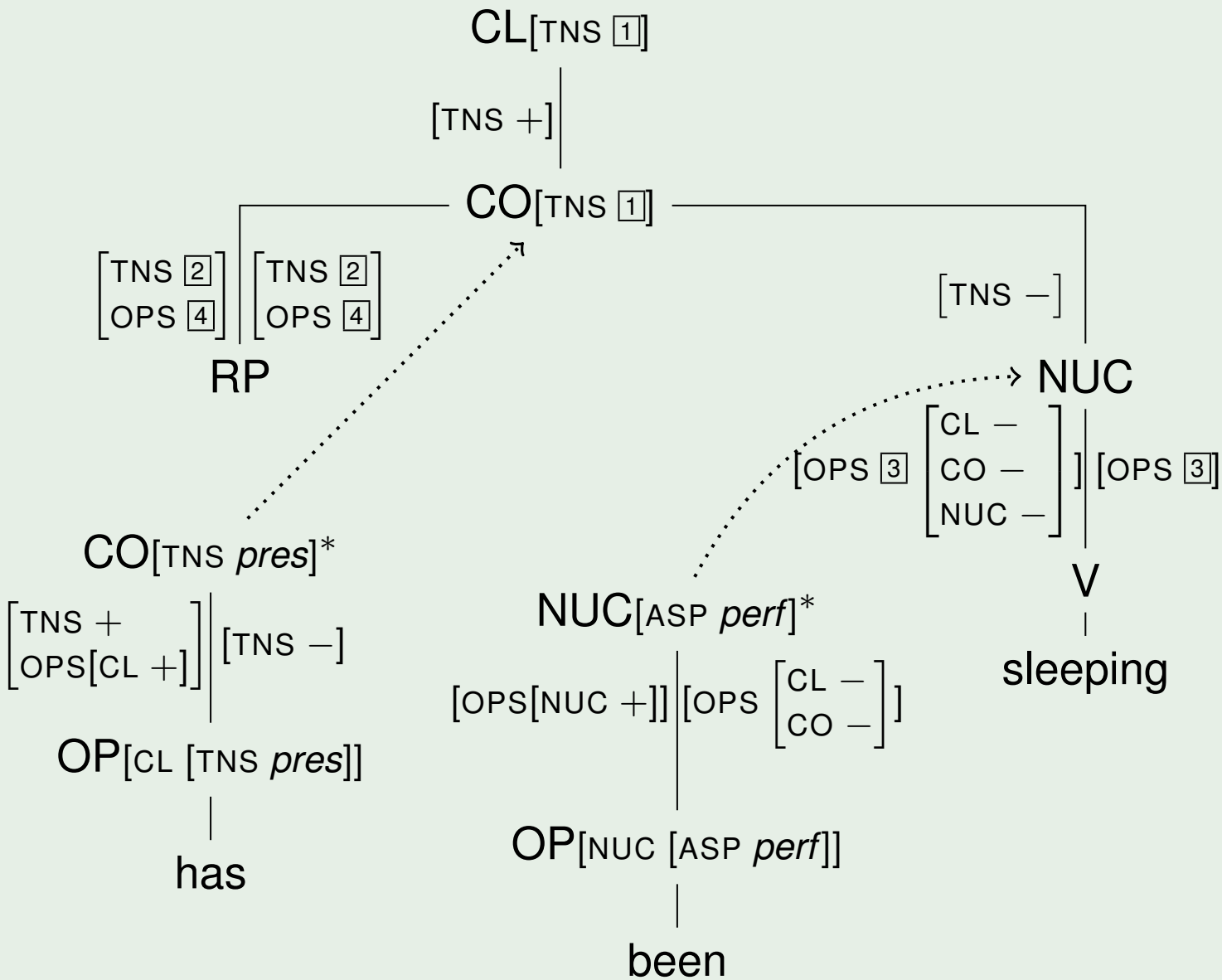


Integrating operators

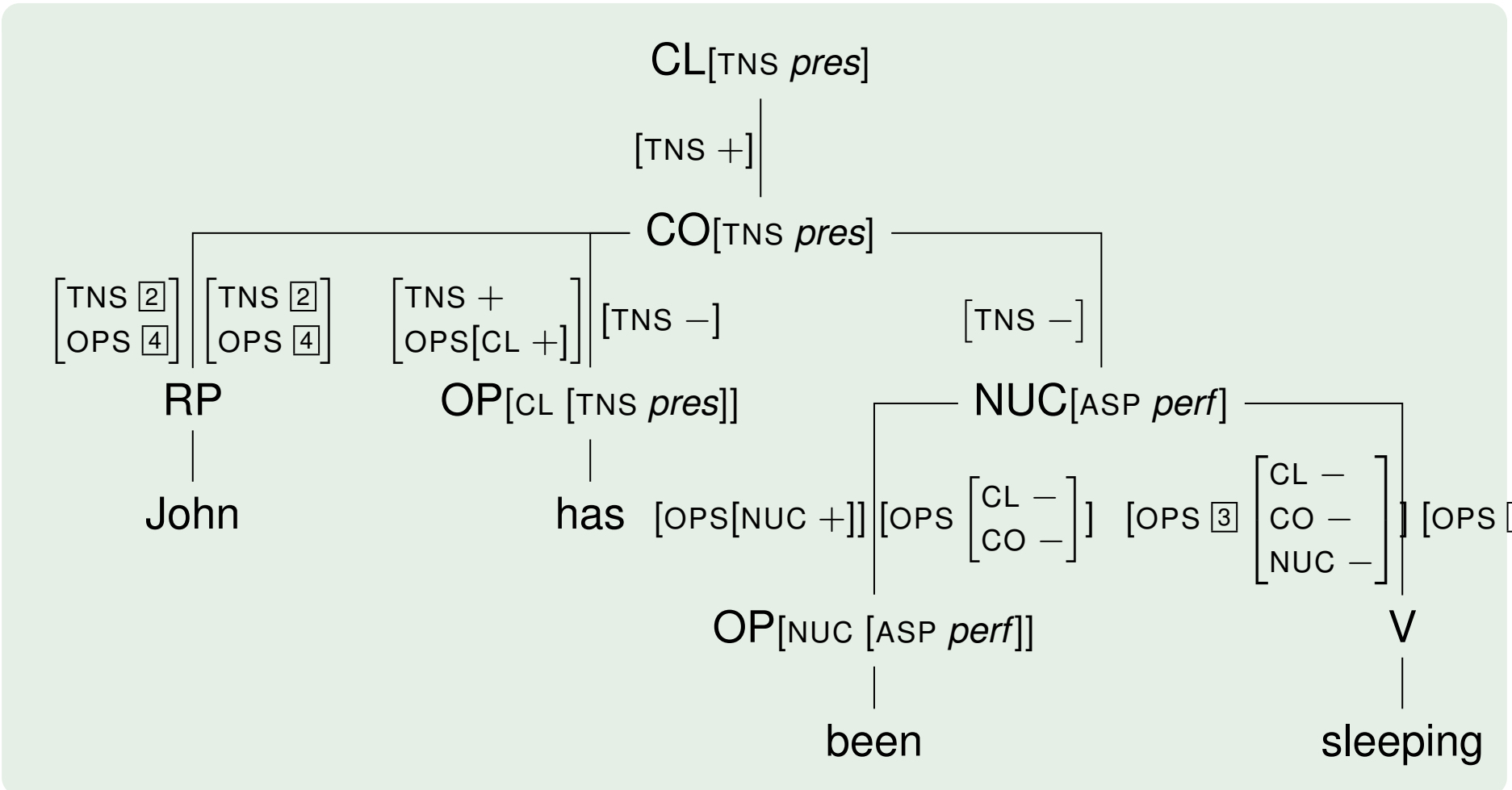
Features for operators (syntactic category OP):

- edge features TNS etc. that express the presence/absence of a specific operator and that can be used to formulate obligatory adjunction constraints.
- edge feature OPS (= operator structure), its value being a feature structure with features CL, CO and NUC with possible values $+$ or $-$.
  OPS guarantees that nuclear, core and clausal operators have to appear in this order when moving outwards from the nuclear predicate.
- node features that specify the contribution of the operator, for instance [NUC [ASP *perf*], CL [TNS *past*]] for the operator *had* in "John had slept".

Integrating operators



Integrating operators



Operators in complex sentences

Cosubordination structures in RRG

- have basically the form $[[\ ]_X\ [\ ]_X]_X$.
- have the characteristic property that X-operators are realized only once but have scope over both constituents.

Examples from Van Valin (2005):

(6) [[Gid-ip]$_{CO}$ [gör-meli-yiz]$_{CO}$]$_{CO}$ (Turkish)
go-LM[1] see-MOD-1PL
'We ought to go and see.'

(7) [[Kim must$_{MOD}$ go]$_{CO}$ [to try]$_{CO}$ [to wash the car]$_{CO}$]$_{CO}$

We assume that it is a general property of cosubordination elementary trees that operator features get passed upwards to the higher X.

[1] LM = linkage marker

Operators in complex sentences

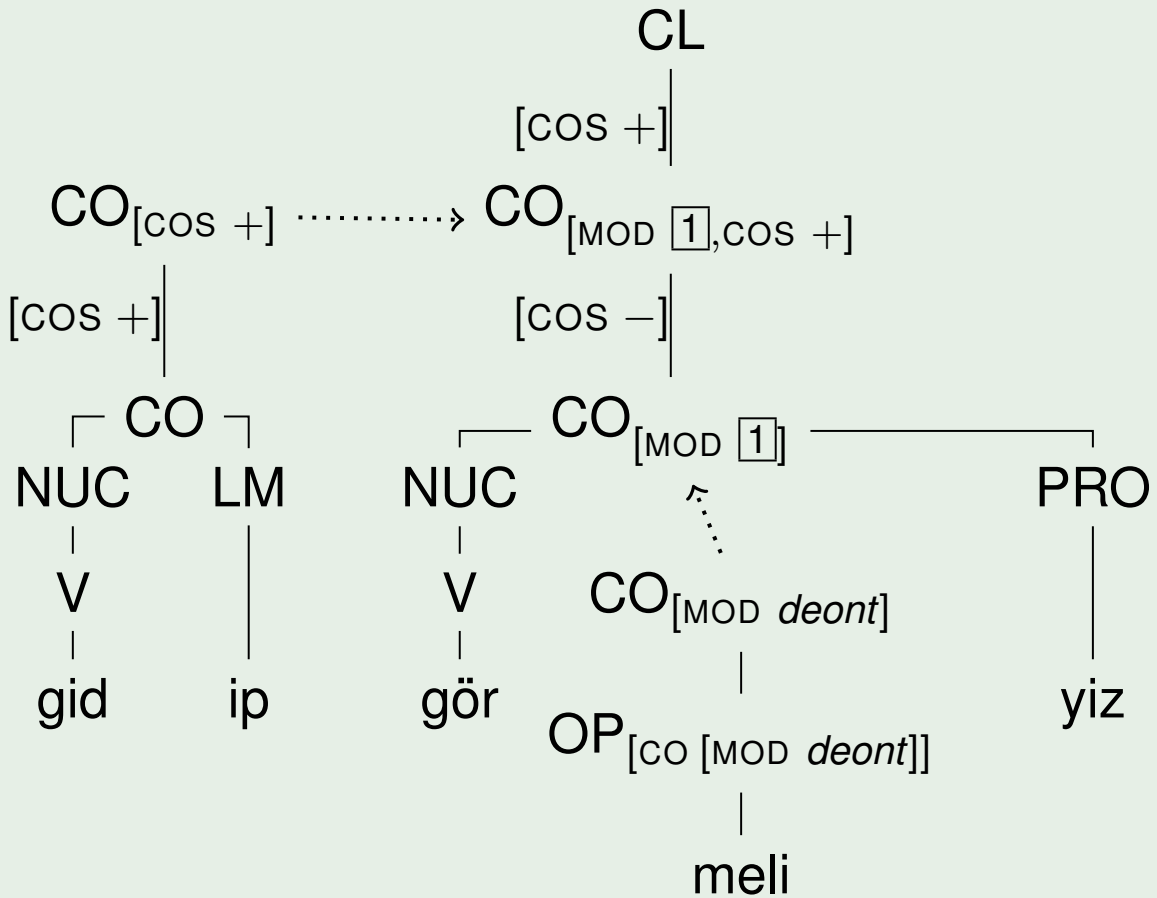
[[Gid-ip]$_{\text{CO}}$ [gör-meli-yiz]$_{\text{CO}}$]$_{\text{CO}}$

Proposal for the elementary trees:

- Special cosubordination tree for *gör PRO* that provides a lower and a higher CO node.
- CO operator features (e.g., MOD) are shared between the two CO nodes and thereby passed upwards from the lower node.
- *gid-ip* is added by adjunction, targeting the higher CO node, thereby adding a second CO daughter.
- Edge feature COS (values +/-) that indicates that adjunction of at least one more core to the left is obligatory.
- Node feature COS (values +/-) that indicate whether a node is the root of a cosubordination structure.

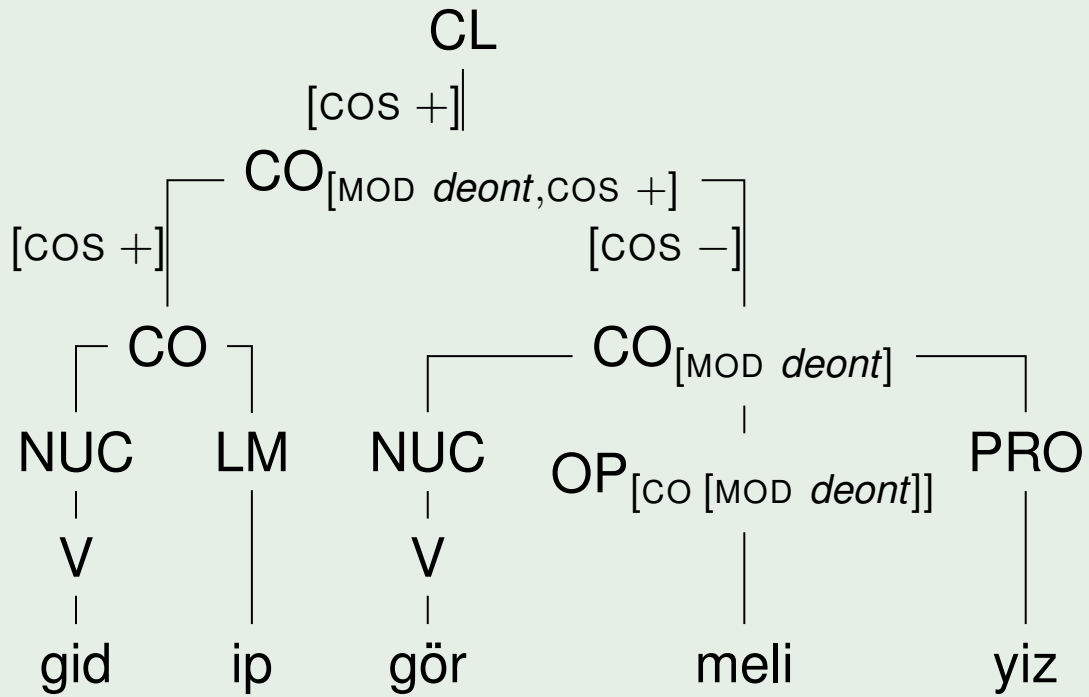
Operators in complex sentences

Cosubordination structure



Operators in complex sentences

Cosubordination structure



Operators in complex sentences

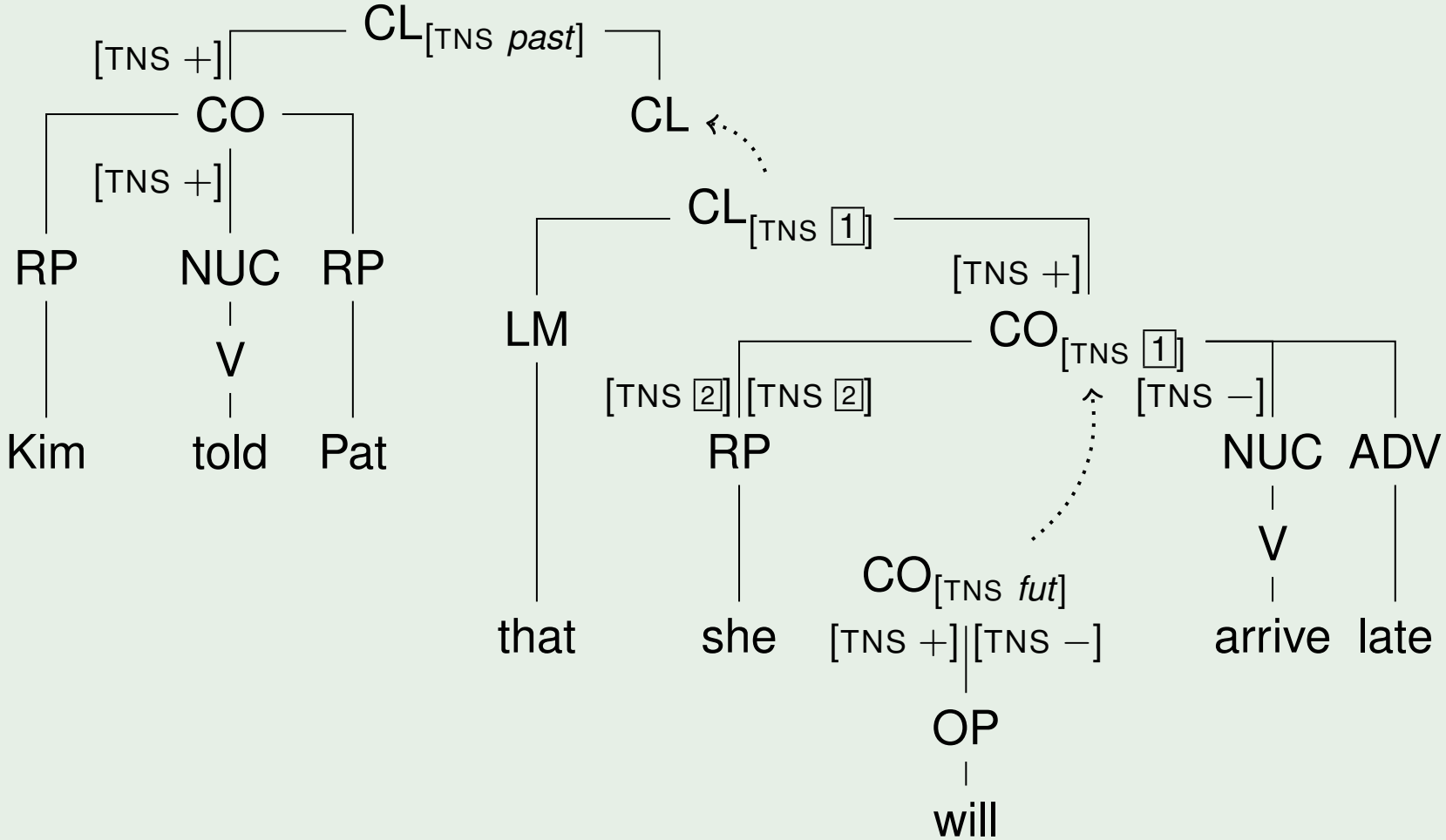
In **subordination** structures, operator projections are built locally. The composition operation is substitution, which means that edge feature percolation is blocked.

(8) [[Kim told Pat]$_{\mathrm{CO}}$ [that [she will arrive late]$_{\mathrm{CO}}$]$_{\mathrm{CL}}$]$_{\mathrm{CL}}$

The two CL nodes in this structure have different TNS values, provided by *told* and *will* respectively.

Operators in complex sentences

Subordination structure



Conclusion: Summary

- We provided a TAG-inspired formalization of RRG as a tree-rewriting grammar.
- Composition operations are (wrapping) substitution for complement insertion and sister adjunction for adding modifiers.
- The resulting formalism, k-TWG is mildly context-sensitive. More conretely, k-TWG are weakly equivalent to well-nested $(k+1)$-LCFRS.
- We introduced features and proposed to use left and right edge features in order to model adjunction constraints.
- Given this architecture, RRG's operator projection can be integrated into the constituent structure, modeling the operator hierarchy and its interaction with the constituent structure within the features.

Conclusion

To do (inter alia)

- Inspect further cases of complex sentences.
- Model the scopal structure of periphery modifiers (e.g., adverbs). The assumption is that this can be done in a similar way as in the case of the operator scope.
- Integrate this formalization of RRG into XMG in order to enable grammar implementation.
- Integrate RRG parsing into TuLiPA in order to enable grammar parsing for testing.
- Long-term goal: full formalization of RRG and integrated framework for RRG-based grammar development.

Merci de votre attention!

References

Bangalore, Srinivas & Aravind K. Joshi. 2010. Introduction. In Srinivas Bangalore & Aravind K. Joshi (eds.), *Supertagging: Using complex lexical descriptions in natural language processing*, 1–31. Cambridge, MA: MIT Press.

Chen-Main, Joan & Aravind Joshi. 2012. A dependency perspective on the adequacy of tree local multi-component tree adjoining grammar. *Journal of Logic and Computation Advance Access* .

Engelfriet, Joost & Erik Meineche Schmidt. 1977. IO and OI. *Journal of Computer and System Sciences* (15). 328—353.

Kallmeyer, Laura. 2016. On the mild context-sensitivity of *k*-Tree Wrapping Grammar. In Annie Foret, Glyn Morrill, Reinhard Muskens, Rainer Osswald & Sylvain Pogodalla (eds.), *Formal grammar. 20th and 21st international conferences, fg 2015, barcelona, spain, august 2015, revised selected papers. fg 2016, bozen, italy, august 2016, proceedings*, vol. 9804 Lecture Notes in Computer Science, 77–93. Springer.

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, Timm Lichte, Rainer Osswald & Simon Petitjean. 2016. Argument linking in LTAG: A constraint-based implementation with XMG. In *Proceedings of the 12th International Workshop on Tree Adjoining Grammars and related formalisms (TAG+12)*, 48–57.

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. 2017. Combining predicate-argument structure and operator projection: Clause structure in Role and Reference Grammar. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and related formalisms (TAG+13)*, 61–70.

Kallmeyer, Laura, Rainer Osswald & Robert D. Van Valin, Jr. 2013. Tree wrapping for Role and Reference Grammar. In Glyn Morrill & Mark-Jan Nederhof (eds.), *Formal grammar (FG 2012/2013)* (Lecture Notes in Computer Science 8036), 175–190. Springer.

Lichte, Timm & Simon Petitjean. 2015. Implementing semantic frames as typed feature structures with XMG. *Journal of Language Modelling* 3(1). 185–228.

Osswald, Rainer & Laura Kallmeyer. to appear. Towards a formalization of Role and Reference Grammar. In Rolf Kailuweit, Eva Staudinger & Lisann Künkel (eds.), *Applying and expanding Role and Reference Grammar*, Freiburg University Press.

Rounds, William C. 1970. Mappings and grammars on trees. *Mathematical Systems Theory* (4). 257—-287.

Van Valin, Robert D., Jr. 2005. *Exploring the syntax-semantics interface*. Cambridge University Press.

Van Valin, Robert D., Jr. & Randy LaPolla. 1997. *Syntax: Structure, meaning and function*. Cambridge University Press.

Vijay-Shanker, K. & Aravind K. Joshi. 1988. Feature structures based tree adjoining grammar. In *Proceedings of coling*, 714–719. Budapest.