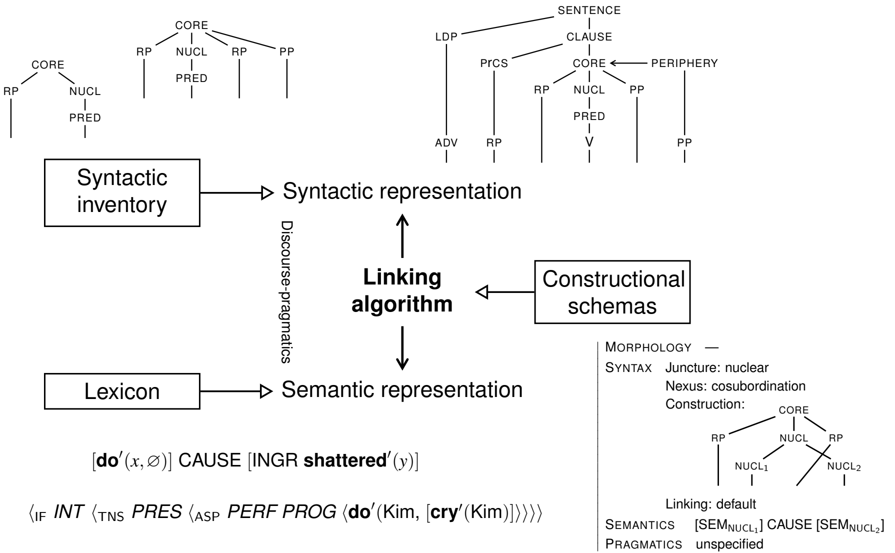
Towards a formalization of Role & Reference Grammar

Laura Kallmeyer & Rainer Osswald
Heinrich-Heine-Universität Düsseldorf

21. 02. 2018

The architecture of Role & Reference Grammar (RRG)



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

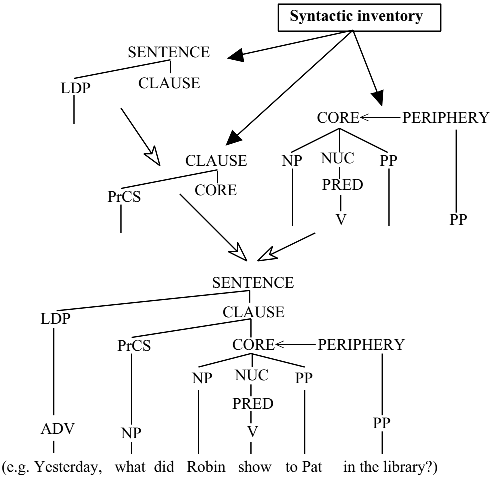
- Take **all explanatory components** of RRG into account.
- Develop a **declarative**, constraint-based formulation.

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

Syntactic representation

The inventory of syntactic templates



[Van Valin 2005, p. 15]

Issues

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

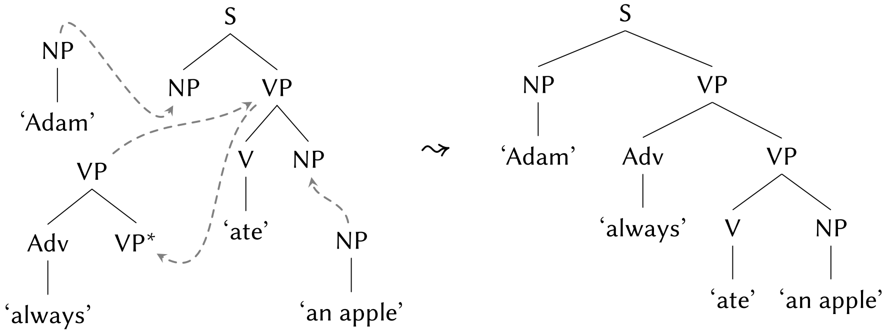
Proposal

- 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.

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.

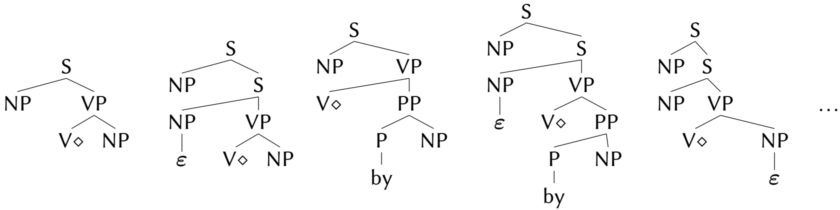
  In particular, there is no need to reproduce the internal structure of an elementary syntactic tree within its associated semantic representation. [Kallmeyer/Joshi 2003]

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



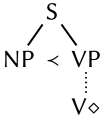
Metagrammar

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

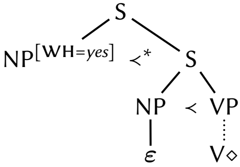
Background: LTAG

Decomposition/factorization in the metagrammar

Class *CanSubj*



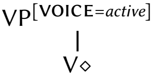
Class *ExtrSubj*



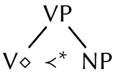
Class *Subj*

CanSubj ∨ *ExtSubj*

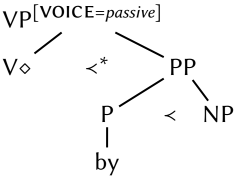
Class *ActV*



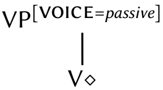
Class *DirObj*



Class *ByObj*

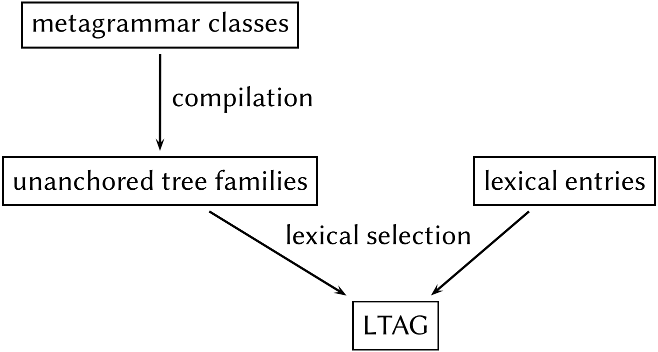


Class *PassV*



Background: LTAG

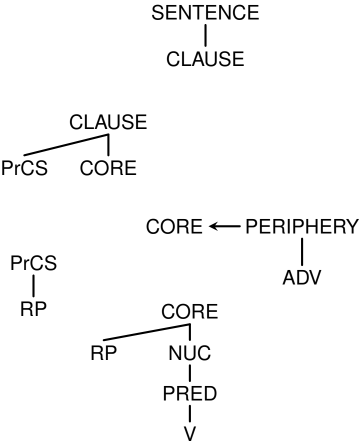
Decomposition/factorization in the metagrammar

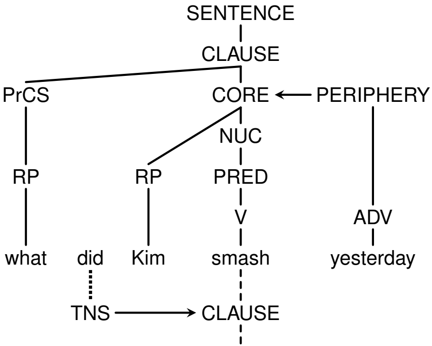


Advantage

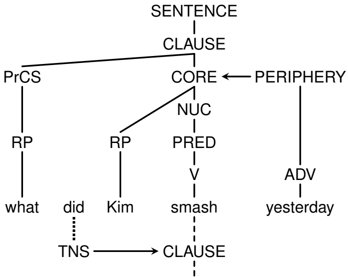
The metagrammar allows one to express and implement lexical and constructional generalizations.

Syntactic templates in RRG

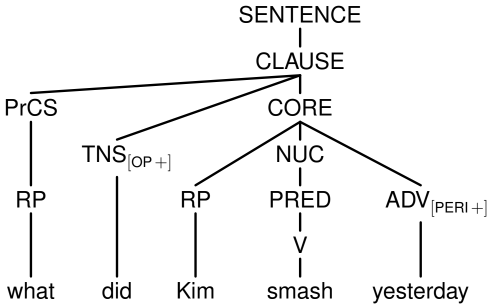




Modified representation



⇝



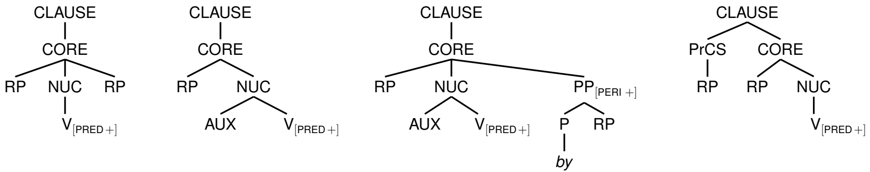
Syntactic representation

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**?

Possible candidates for elementary trees in RRG

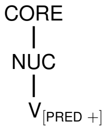
- Basic predication templates and their variants, e.g.



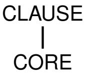
- Constructional schemas (strictly speaking, their syntactic dimension)

Metagrammar sketches

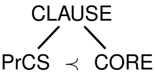
core-spine



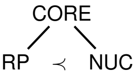
core-clause



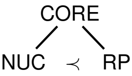
precore-slot



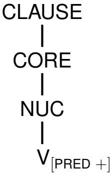
prenuc-rp



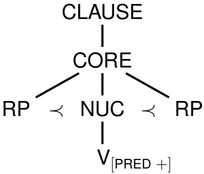
postnuc-rp



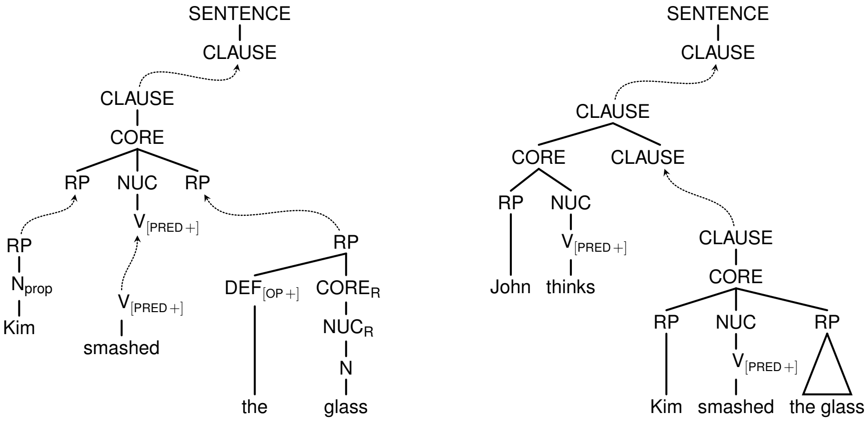
clause-spine $:=$
core-spine $\wedge$ *core-clause*



base-transitive $:=$
clause-spine $\wedge$ *prenuc-rp* $\wedge$ *postnuc-rp*

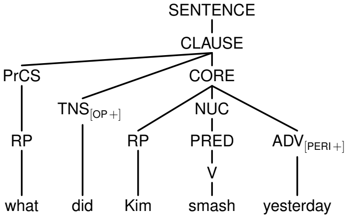


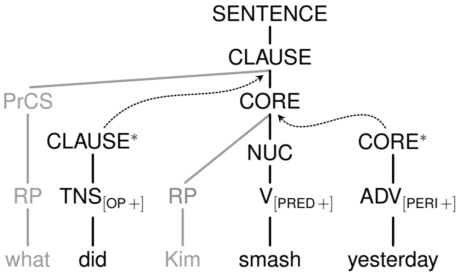
Mode of composition I: **(simple) substitution**



Syntactic composition

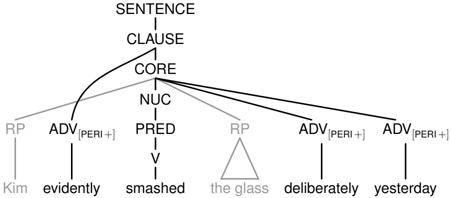
Mode of composition II: (sister) adjunction

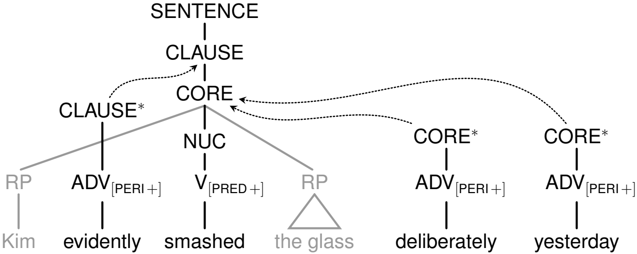




Syntactic composition

Mode of composition II: **(sister) adjunction**



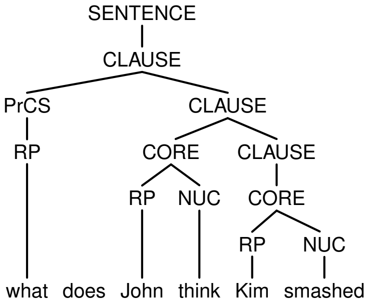


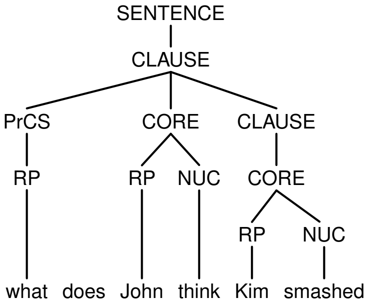
Issue: Crossing branches (more about this later)

Wh-extraction

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

Possible analyses of (1):

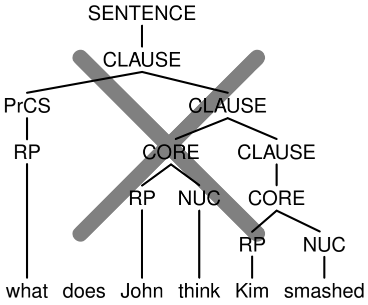




Wh-extraction

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

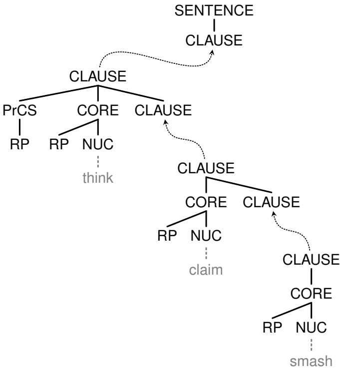
Possible analyses of (1):



Wh-extraction

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

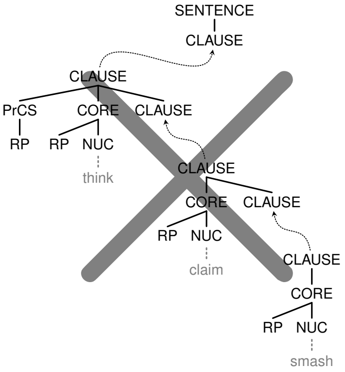
Compositional derivation of (2):

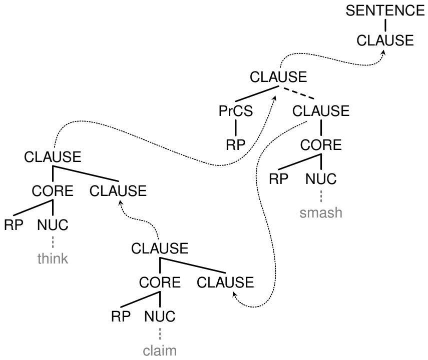


Wh-extraction

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

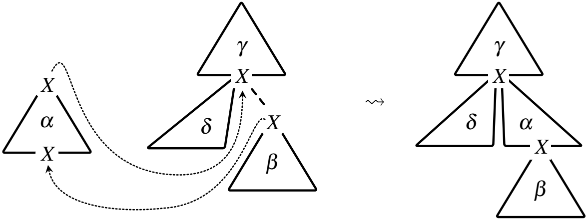
Compositional derivation of (2):

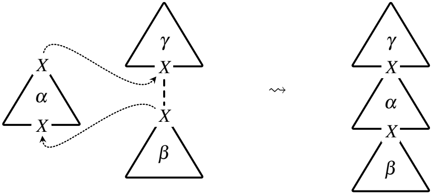




Syntactic composition

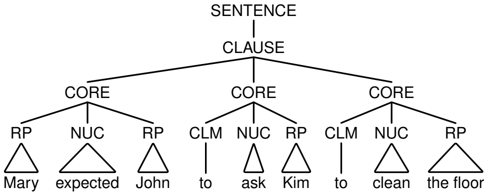
Mode of composition III: wrapping (substitution) (special versions)

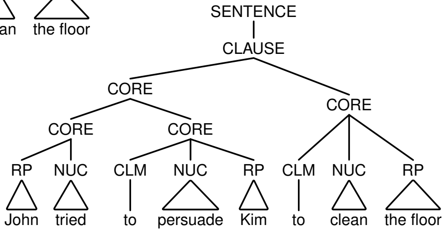


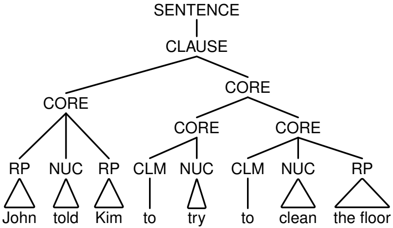


Syntactic composition

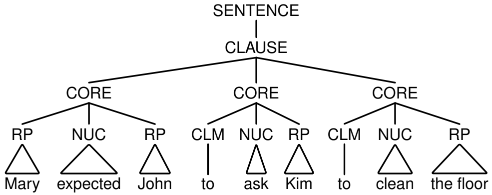
Control and matrix coding (≈ raising)

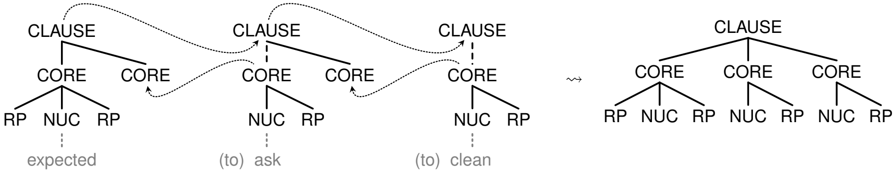




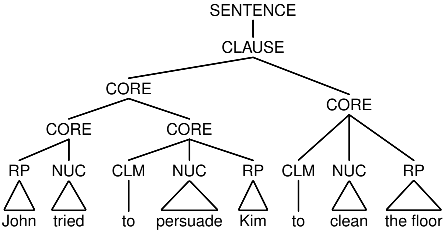


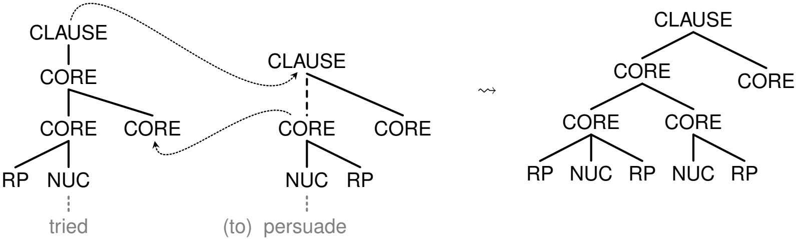
Control and matrix coding (≈ raising)





Control and matrix coding (≈ raising)

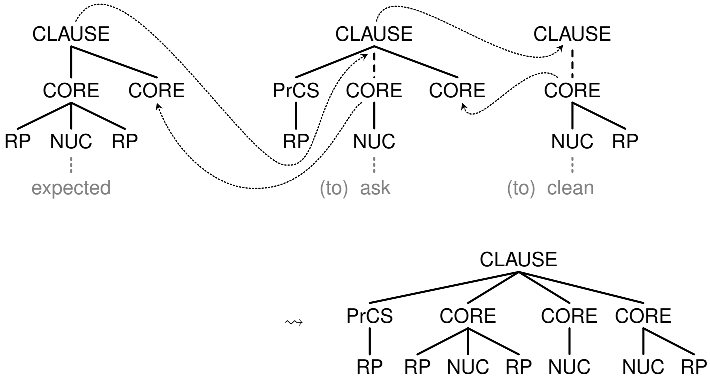




Syntactic composition

Control, matrix coding & wh-extraction

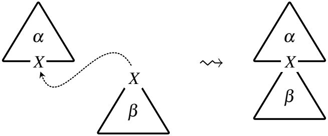
(3) **Whom** did Mary expect John to **ask** to clean the floor?



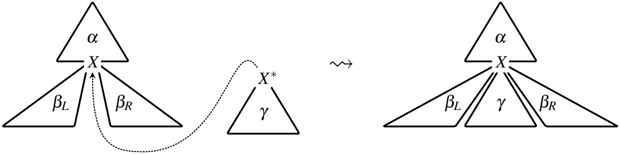
Syntactic composition

Modes of composition (⇝ **Tree Wrapping Grammar**; **TWG**)

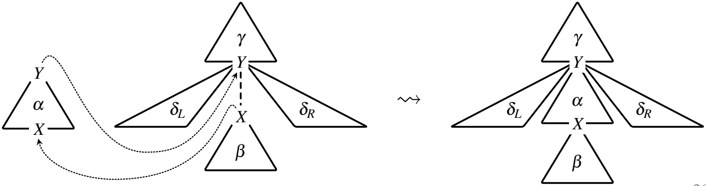
I. Simple substitution



II. Sister adjunction



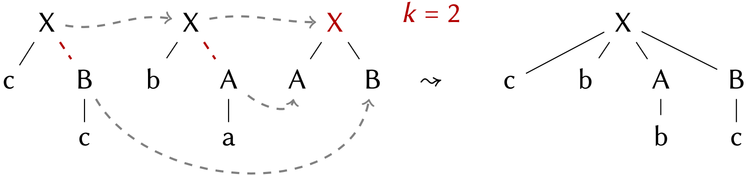
III. Wrapping substitution



Formal properties of TWGs

- For every k-TWG (a constrained form of 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.

Idea of k-TWG: limit the number of times a d-edge can stretch across a specific node to k (except for nested wrappings).



Formal properties of TWGs

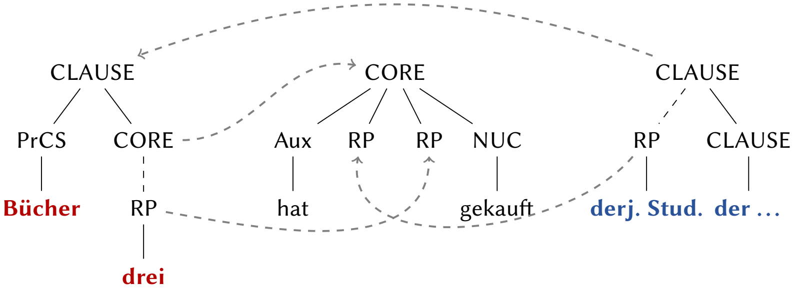
A $k > 1$ allows extraction out of several arguments

(4) **Bücher** hat **derjenige Student** **drei** gekauft **der am meisten Geld hatte**
books has that student three bought who the most money had
'the student with the most money bought three books'

(from Chen-Main & Joshi, 2012)

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

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

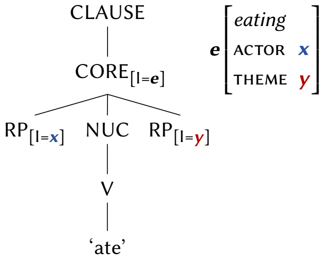


Example

(5) Adam ate an apple.

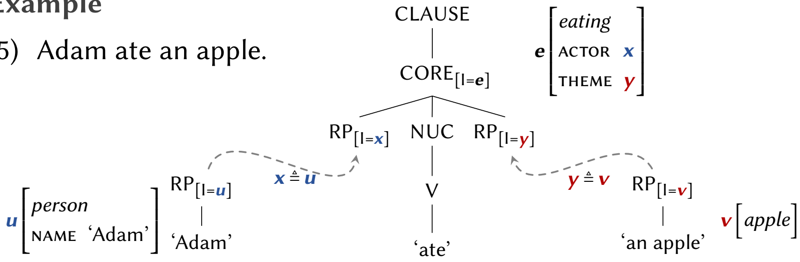
Example

(5) Adam ate an apple.



Example

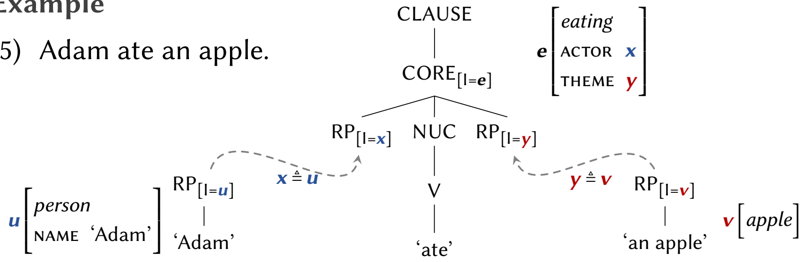
(5) Adam ate an apple.

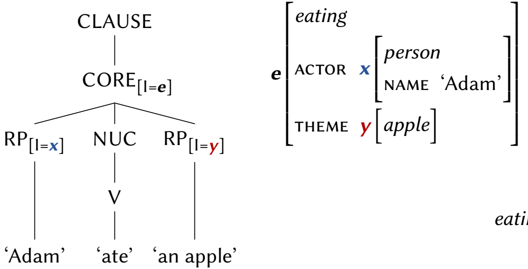


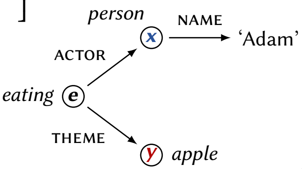
Syntax-semantics interface

Example

(5) Adam ate an apple.







Syntax-semantics interface

Summary of the LTAG + frame semantics perspective on RRG:

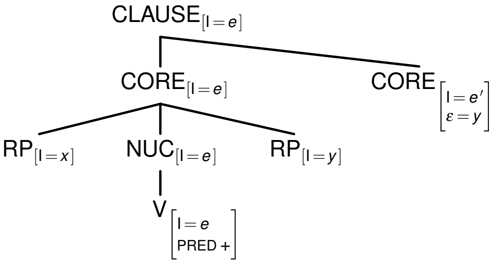
- **Elementary construction**
  = elementary tree (full argument projection) + semantic frame
  + linking of frame node variables to interface features in the tree

- "**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 → 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.

“Raising to object” (e.g. NP *expect / believe / require* NP *to*-INF)



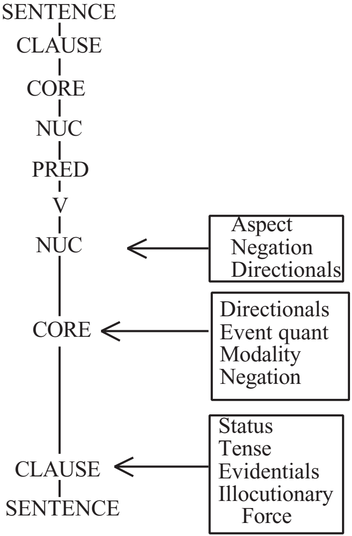
$$e \begin{bmatrix} \textit{de-re-attitude} & \\ \text{COGNIZER} & x \\ \text{TOPIC} & y \\ \text{CONTENT} & e' \begin{bmatrix} \textit{situation} \end{bmatrix} \end{bmatrix}$$

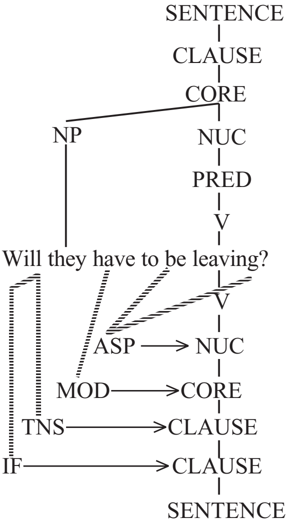
$$\textit{participant-of}\,(y, e')$$

Note Passive is possible in both cores:

(6) Mary **expects** the grant proposal **to be completed** within the next week.
The grant proposal **is expected** **to be completed** within the next week.

Operator projection

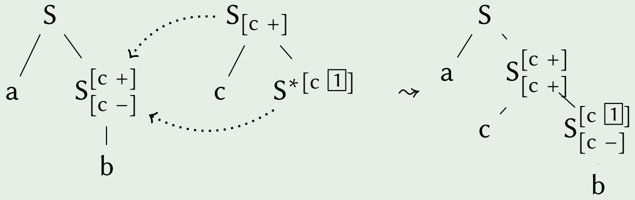




[Van Valin 2005: 12/14]

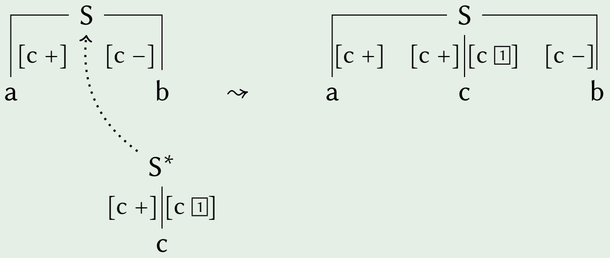
Adding features

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



Adding features

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



Adding features

- 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:

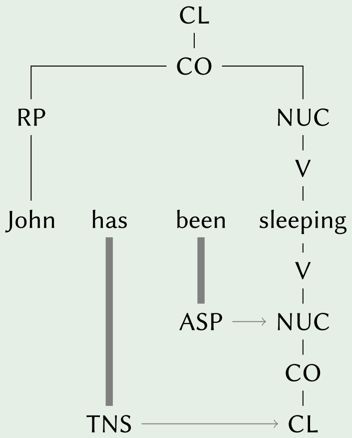
Layer	Nucleus	Core	Clause
Operators	Aspect	Directionals	Status
	Negation	Event quantification	Tense
	Directionals	Modality	Evidentials
		Negation	Illocutionary Force

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.

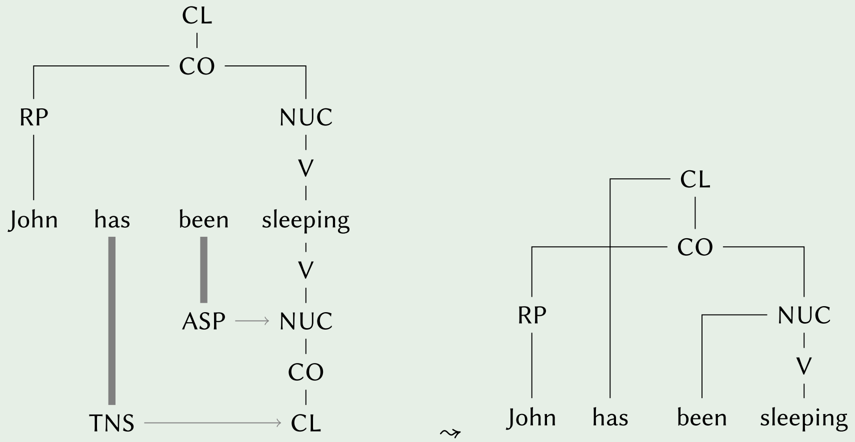
Integrating operators

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.

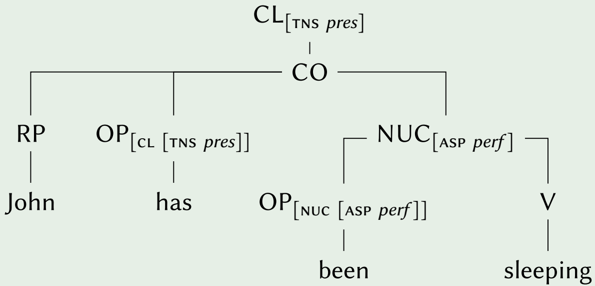


Integrating operators

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.

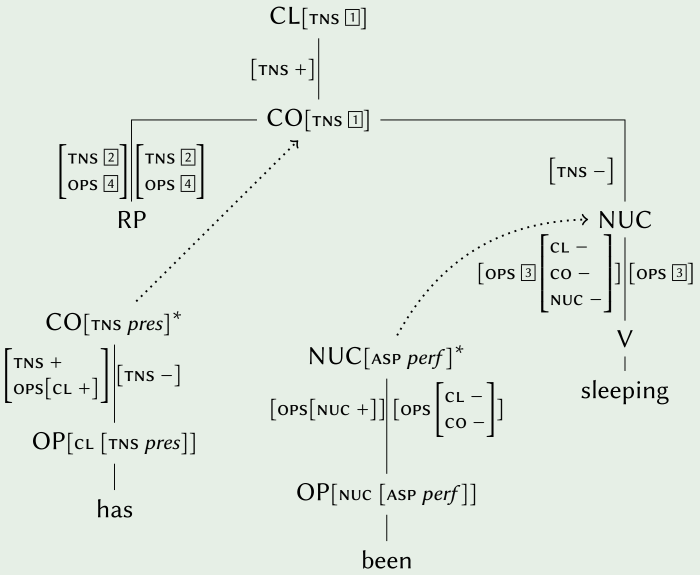


Integrating operators

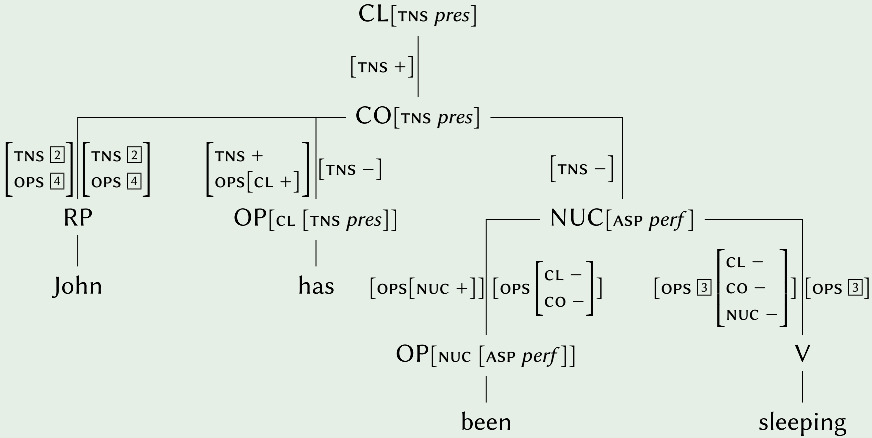
Features for operators (syntactic category OP):

- 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 $\left[\textsc{nuc}\left[\textsc{asp}\ \textit{perf}\right], \textsc{cl}\left[\textsc{tns}\ \textit{past}\right]\right]$ for the operator *had* in "John had slept".

Integrating operators



Integrating operators



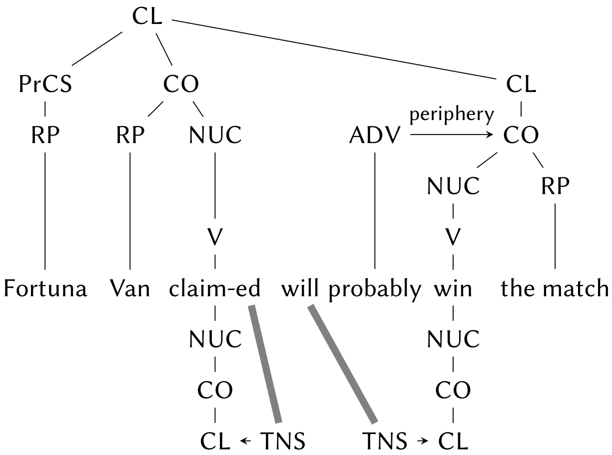
An extended example

(7) Fortuna Van claimed will probably win the match.

An extended example

(7) Fortuna Van claimed will probably win the match.

Syntax:



An extended example

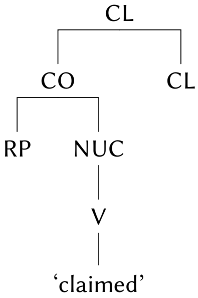
(7) Fortuna Van claimed will probably win the match.

Semantics:

$$
\begin{bmatrix}
\textit{statement} & \\
\text{ACTOR} & \boxed{1}\begin{bmatrix}\textit{person} & \\ \text{NAME} & \text{‘Van´}\end{bmatrix} \\
\text{SPEAKER} & \boxed{1} \\
\text{MESSAGE} & \begin{bmatrix}
\textit{prediction} & \\
\text{ABOUT} & \begin{bmatrix}
\textit{winning} & \\
\text{ACTOR} & \begin{bmatrix}\textit{team} & \\ \text{NAME} & \text{‘Fortuna’}\end{bmatrix} \\
\text{THEME} & \begin{bmatrix}\textit{match}\end{bmatrix}
\end{bmatrix} \\
\text{PROBABILITY} & \begin{bmatrix}\textit{high}\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

An extended example

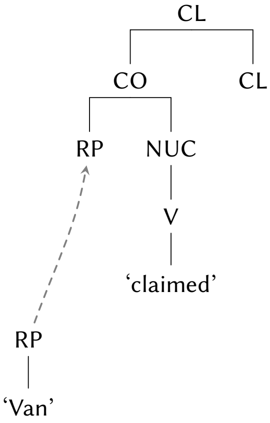
Argument insertion by (wrapping) substitution.



Argument slots (= substitution nodes) have to be filled in order to obtain a well-formed complete syntactic tree.

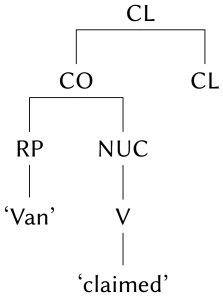
An extended example

Argument insertion by (wrapping) substitution.



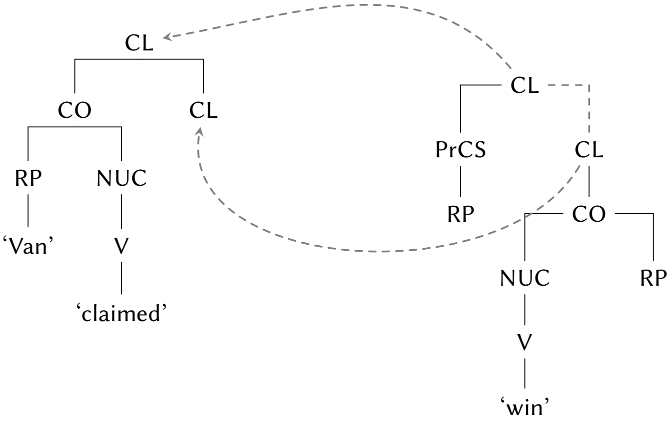
An extended example

Argument insertion by (wrapping) substitution.



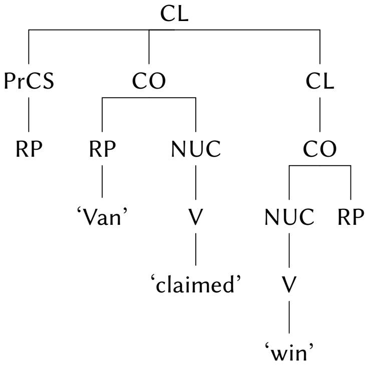
An extended example

Argument insertion by (wrapping) substitution.



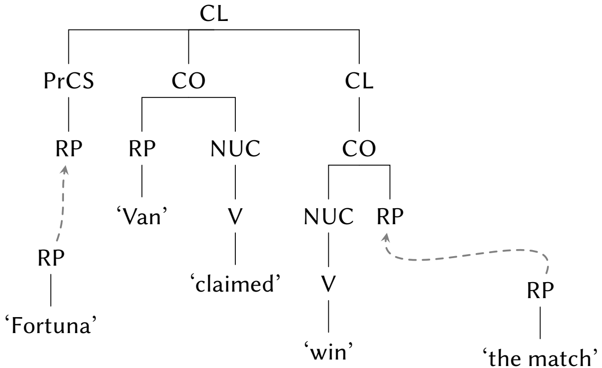
An extended example

Argument insertion by (wrapping) substitution.



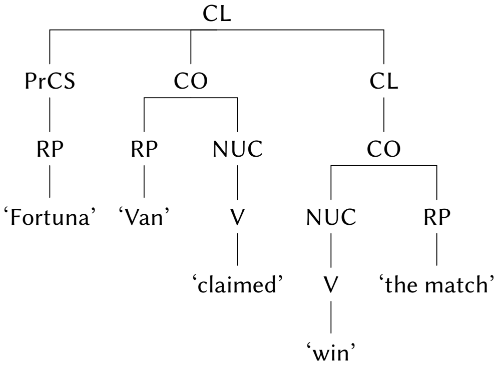
An extended example

Argument insertion by (wrapping) substitution.



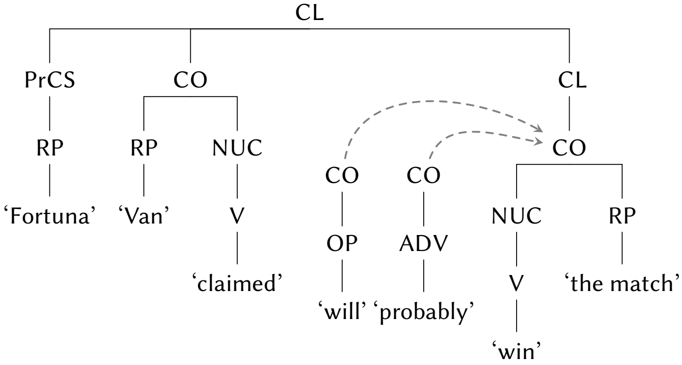
An extended example

Argument insertion by (wrapping) substitution.



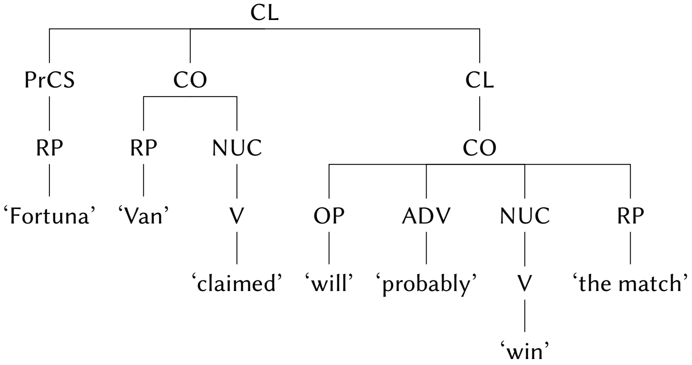
An extended example

Operators and modifiers are added by sister adjunction.



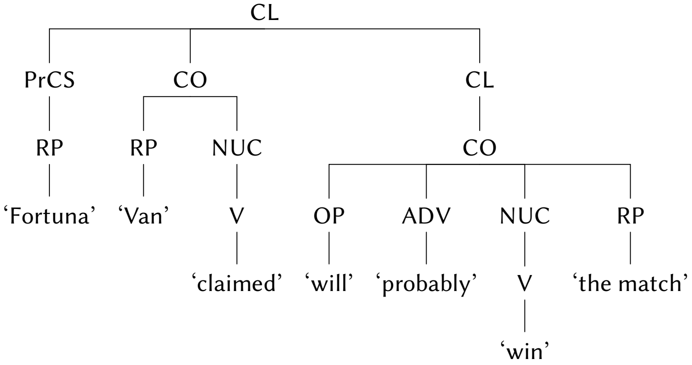
An extended example

Operators and modifiers are added by sister adjunction.



An extended example

Operators and modifiers are added by sister adjunction.



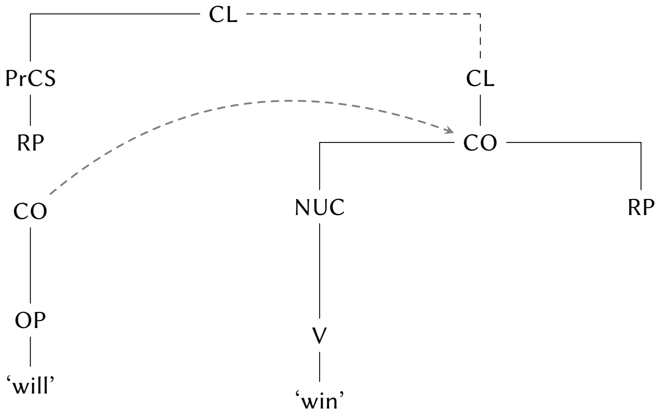
(The operator projection as well as modifier scope is modeled in the features.)

An extended example

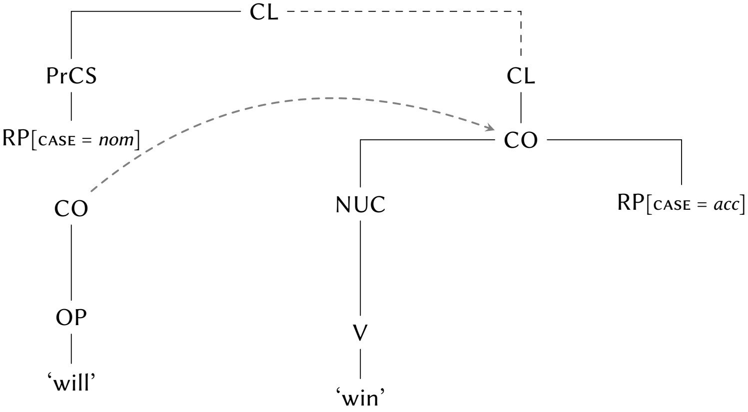
Features

- Features on nodes take care of agreement, case assignment, tense etc.
- Features between edges express constraints on possible adjunctions in between.

An extended example

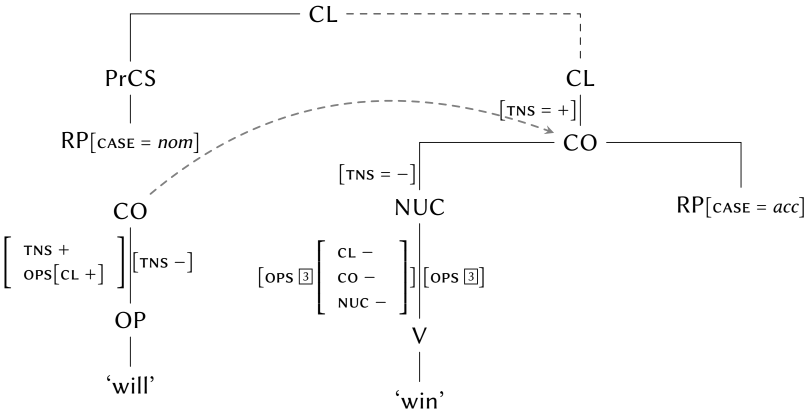


An extended example



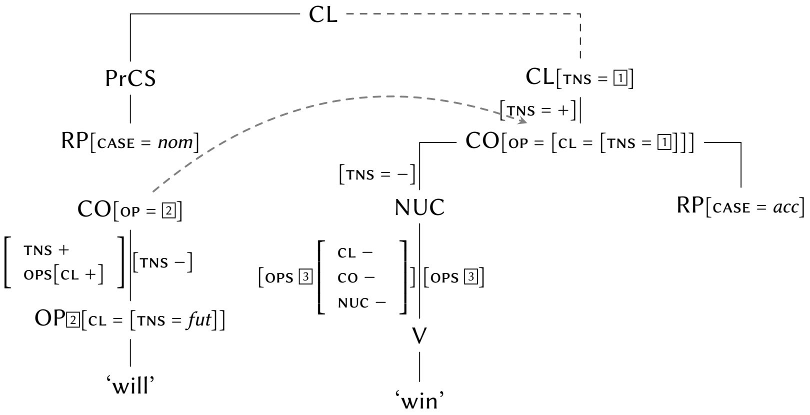
CASE on nodes for case assignment

An extended example



TNS on edges for obligatory adjunction of a single tns operator
OPS on edges to keep track of the the correspondence between surface order and operator hierarchy

An extended example

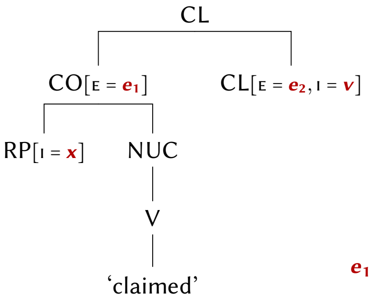


OP on nodes that lists the operators of the entire layered structure
TNS etc. on the corresponding layer nodes
CL, CO, NUC on OP nodes that characterize the operator’s contribution

An extended example

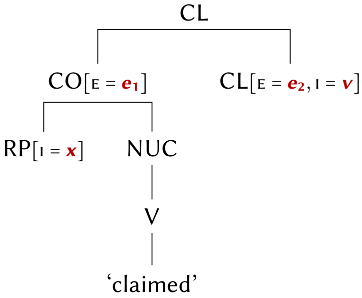
Interfacing syntax and semantics

- Interface features link frame nodes to syntactic nodes.
- Their unification during syntactic composition triggers semantic frame unification.



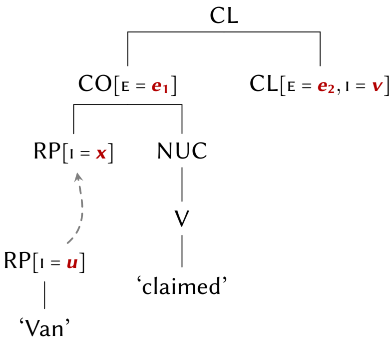
$$e_1 \begin{bmatrix} \textit{statement} & \\ \text{ACTOR} & x \\ \text{SPEAKER} & x \\ \text{MESSAGE} & v \begin{bmatrix} \textit{prediction} & \\ \text{ABOUT} & e_2 \end{bmatrix} \end{bmatrix}$$

An extended example



$$e_1 \begin{bmatrix} \textit{statement} & \\ \text{ACTOR} & x \\ \text{SPEAKER} & x \\ \text{MESSAGE} & v \begin{bmatrix} \textit{prediction} & \\ \text{ABOUT} & e_2 \end{bmatrix} \end{bmatrix}$$

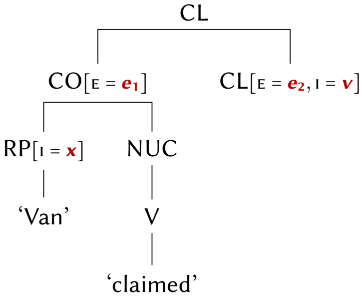
An extended example



$$e_1\begin{bmatrix} \textit{statement} & \\ \textsc{actor} & x \\ \textsc{speaker} & x \\ \textsc{message} & v\begin{bmatrix} \textit{prediction} & \\ \textsc{about} & e_2 \end{bmatrix} \end{bmatrix}$$

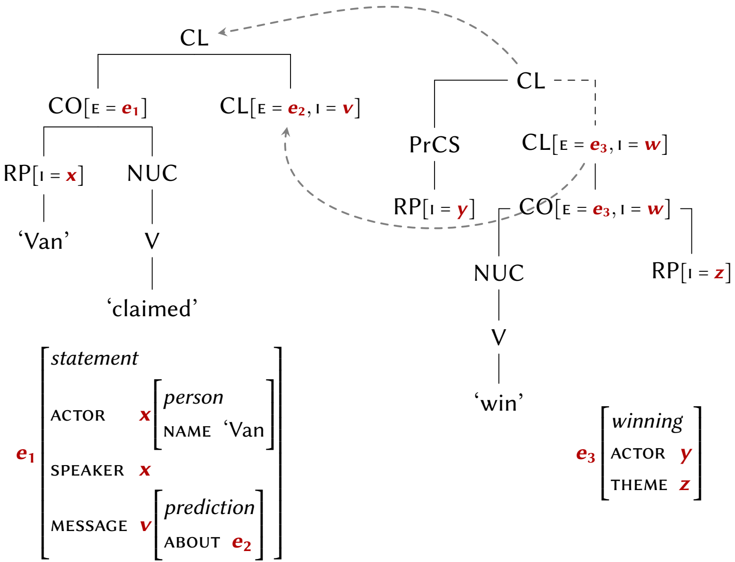
$$u\begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{‘Van’} \end{bmatrix}$$

An extended example

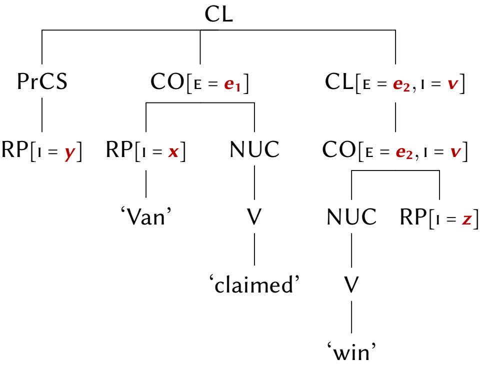


$$
e_1 \begin{bmatrix} \textit{statement} & & \\ \textsc{actor} & x & \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{‘Van} \end{bmatrix} \\ \textsc{speaker} & x & \\ \textsc{message} & v & \begin{bmatrix} \textit{prediction} & \\ \textsc{about} & e_2 \end{bmatrix} \end{bmatrix}
$$

An extended example

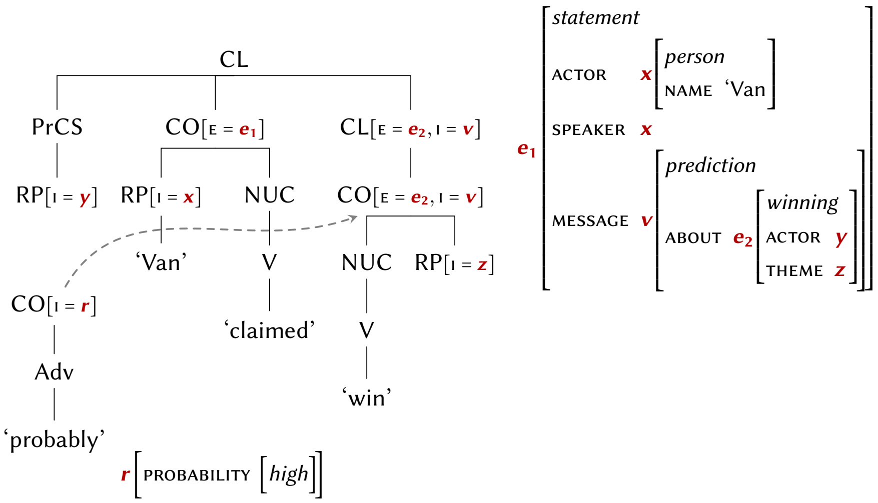


An extended example

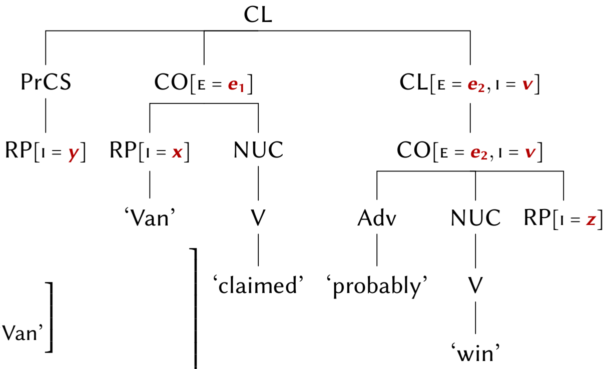


$$
e_1 \begin{bmatrix} \textit{statement} & & \\ \textsc{actor} & x & \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{'Van} \end{bmatrix} \\ \textsc{speaker} & x & \\ \textsc{message} & v & \begin{bmatrix} \textit{prediction} & & \\ \textsc{about} & e_2 & \begin{bmatrix} \textit{winning} & \\ \textsc{actor} & y \\ \textsc{theme} & z \end{bmatrix} \end{bmatrix} \end{bmatrix}
$$

An extended example



An extended example



$$e_1 \begin{bmatrix} \textit{statement} & \\ \text{ACTOR} & x \begin{bmatrix} \textit{person} & \\ \text{NAME} & \text{`Van'} \end{bmatrix} \\ \text{SPEAKER} & x \\ \text{MESSAGE} & v \begin{bmatrix} \textit{prediction} & \\ \text{ABOUT} & e_2 \begin{bmatrix} \textit{winning} & \\ \text{ACTOR} & y \\ \text{THEME} & z \end{bmatrix} \\ \text{PROBABILITY} & \begin{bmatrix} \textit{high} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

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):

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

(9) $[[$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

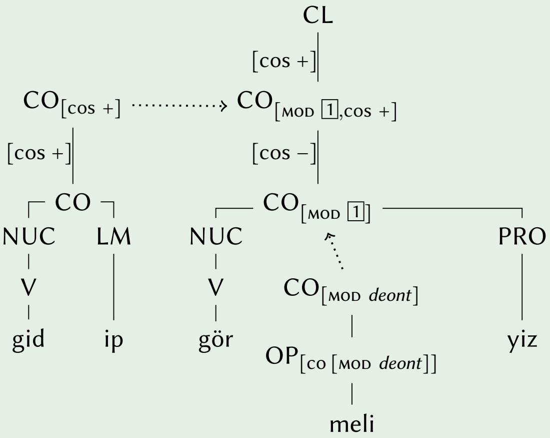
[[Gid-ip]$_{CO}$ [gör-meli-yiz]$_{CO}$]$_{CO}$

Proposal for the elementary trees:

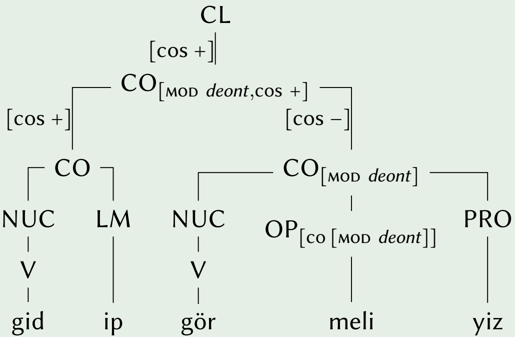
- Special cosubordination tree for *gör PRO* that provides a lower and a higher CO node.
- CO operator features (e.g., MOD) are shared between the two CO nodes and thereby passed upwards from the lower node.
- *gid-ip* is added by adjunction, targeting the higher CO node, thereby adding a second CO daughter.
- 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



Cosubordination structure



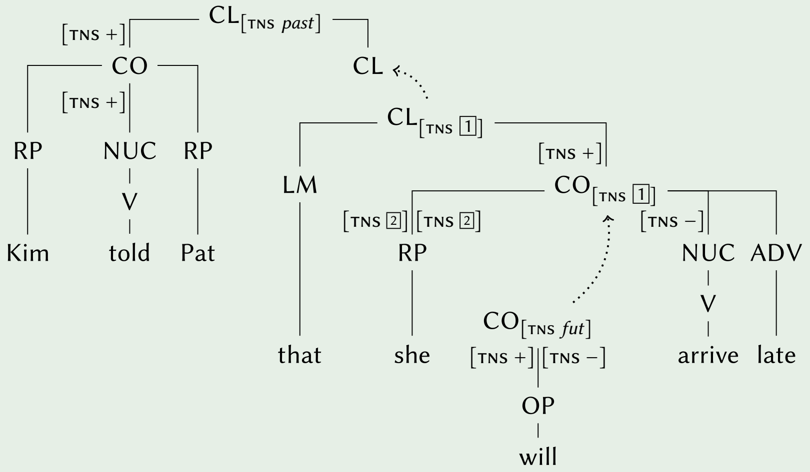
Operators in complex sentences

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

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

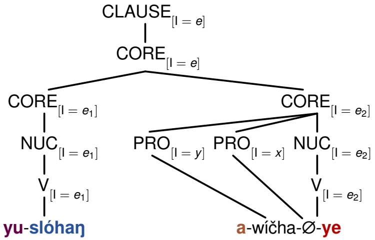
The two CL nodes in this structure have different TNS values, provided by *told* and *will* respectively.

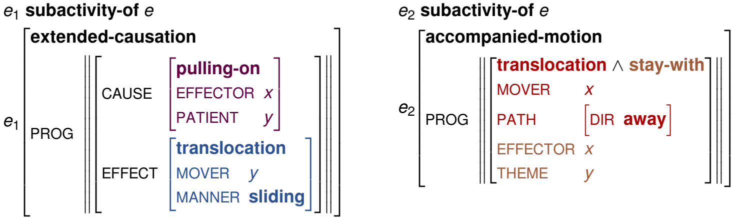
Subordination structure



(11) **Yu-slóhaŋ** **a**-wíčha-∅-**ye**. (Lakhota)
by.pulling-slide AM-3PL.UG.ANIM-3SG.AC-go
'She was dragging them away.'

AC	–	Actor
AM	–	Accompanied motion
ANIM	–	Animate
UG	–	Undergoer





More on cosubordination structures

Question: Are core cosubordination constructions

1. **elementary** constructions (≈ syntactic templates / constructional schemas) or are they
2. **compositional** structures derived by the modes of syntactic (and semantic) composition?

More on cosubordination structures

Question: Are core cosubordination constructions

1. **elementary** constructions (≈ syntactic templates / constructional schemas) or are they
2. **compositional** structures derived by the modes of syntactic (and semantic) composition?

Key argument for Option 2: **iteration**

More on cosubordination structures

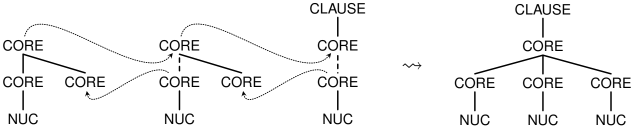
Question: Are core cosubordination constructions

1. **elementary** constructions (≈ syntactic templates / constructional schemas) or are they
2. **compositional** structures derived by the modes of syntactic (and semantic) composition?

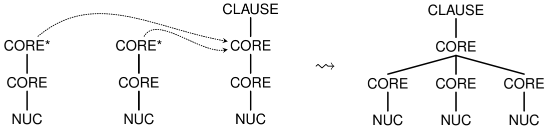
Key argument for Option 2: **iteration**

Possible ways of composition:

wrapping

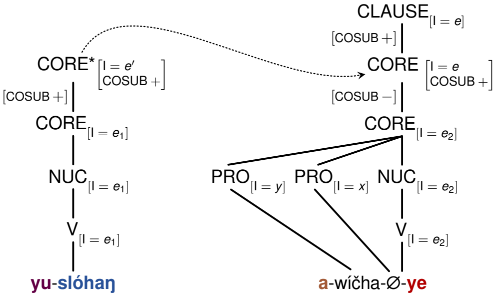


sister adjunction



(12) **Yu-slóhaŋ** **a**-wíčha-∅-**ye**. [= (11)]
by.pulling-slide AM-3PL.UG.ANIM-3SG.AC-go
'She was dragging them away.'

"licenced adjunction"



"macrorole unification"

e_1 **subactivity-of** e'

$$e_1\begin{bmatrix}\textbf{extended-causation} & \\ \text{ACTOR} & x' \\ \text{UNDERGOER} & y' \\ \text{PROG} & \|\ldots\| \end{bmatrix}$$

e_2 **subactivity-of** e

$$e_2\begin{bmatrix}\textbf{accompanied-motion} & \\ \text{ACTOR} & x \\ \text{UNDERGOER} & y \\ \text{PROG} & \|\ldots\| \end{bmatrix}$$

More on cosubordination structures

(13)	Watashi	wa	taru	o	**korogashi-te**	chikashitsu	ni	**ire**-ta.	(Japanese)
	1SG	TOP	barrel	ACC	roll(CAUS)-TE	basement	LOC	take.into-PAST	

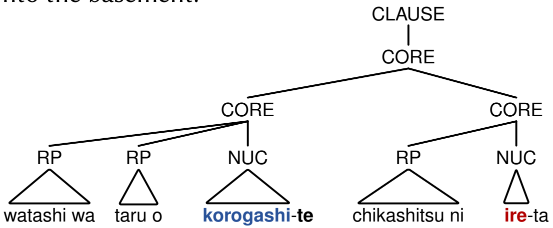
'I rolled the barrel into the basement.'

More on cosubordination structures

(13) Watashi wa taru o **korogashi-te** chikashitsu ni **ire-ta.** (Japanese)

Watashi	wa	taru	o	**korogashi-te**	chikashitsu	ni	**ire**-ta.
1SG	TOP	barrel	ACC	roll(CAUS)-TE	basement	LOC	take.into-PAST

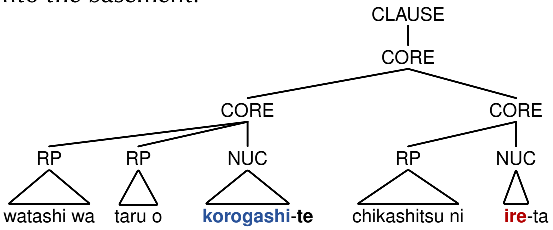
'I rolled the barrel into the basement.'

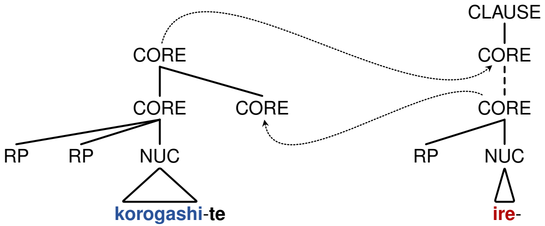


More on cosubordination structures

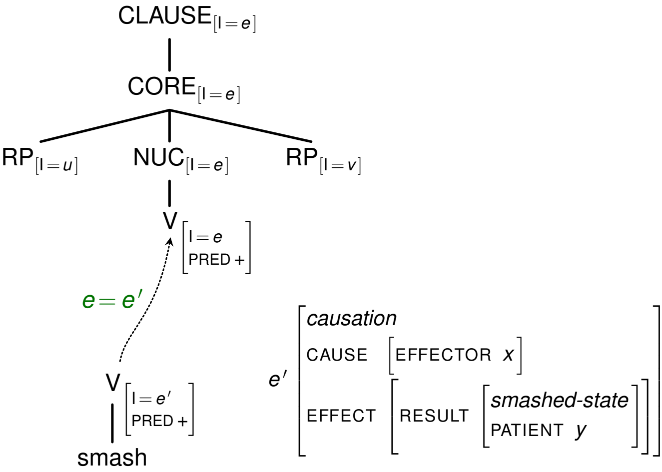
(13)	Watashi wa	taru	o	**korogashi-te**	chikashitsu	ni	**ire**-ta.	(Japanese)
	1SG TOP	barrel	ACC	roll(CAUS)-TE	basement	LOC	take.into-PAST	

‘I rolled the barrel into the basement.’



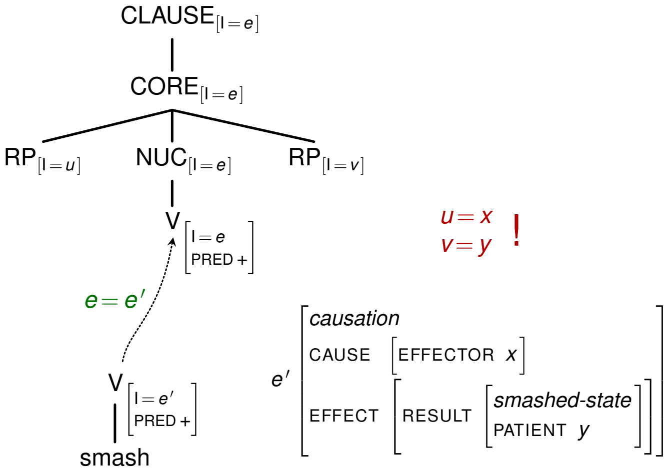


base-transitive



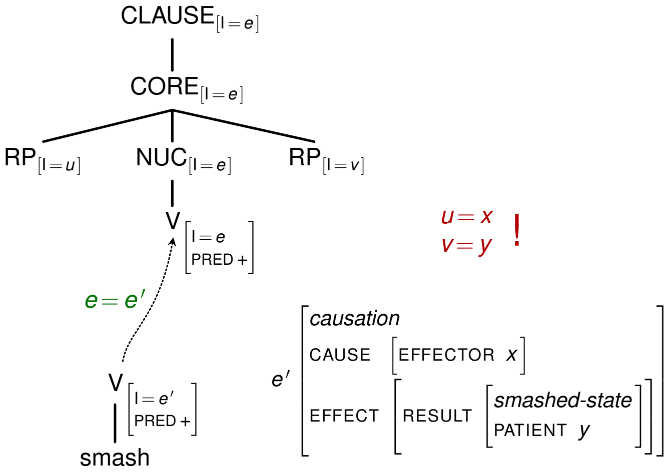
Argument linking

base-transitive

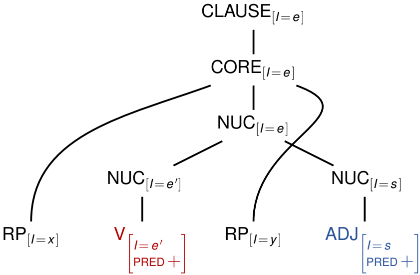


Argument linking

base-transitive [→ Kallmeyer/Lichte/Osswald/Petitjean 2016]



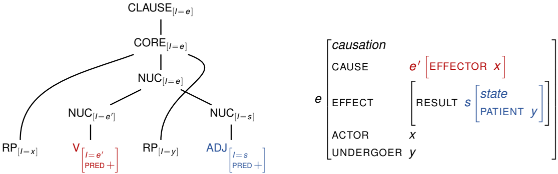
Example Adjectival resultative construction in English
(**kick open**, **push shut**, **wipe clean**, ...)

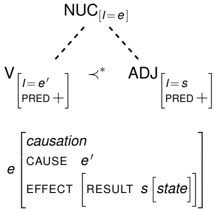


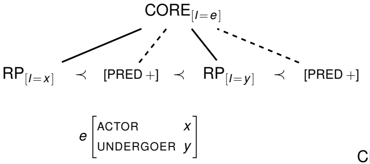
$$
e \begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & e' \begin{bmatrix} \text{EFFECTOR} & x \end{bmatrix} \\ \text{EFFECT} & \begin{bmatrix} \text{RESULT} & s \begin{bmatrix} \textit{state} & \\ \text{PATIENT} & y \end{bmatrix} \end{bmatrix} \\ \text{ACTOR} & x \\ \text{UNDERGOER} & y \end{bmatrix}
$$

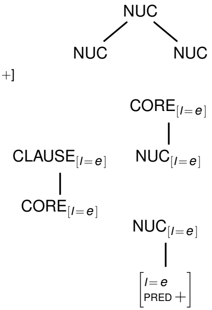
Constructional schemas

Example Adjectival resultative construction in English









$\forall(\textit{state} \rightarrow \text{PATIENT} : \top)$

$\forall(\text{CAUSE EFFECTOR} : \top \rightarrow \text{CAUSE EFFECTOR} \doteq \text{ACTOR})$

$\forall(\text{EFFECT RESULT PATIENT} : \top \rightarrow \text{EFFECT RESULT PATIENT} \doteq \text{UNDERGOER})$

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* .

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.

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

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