RRG & FG

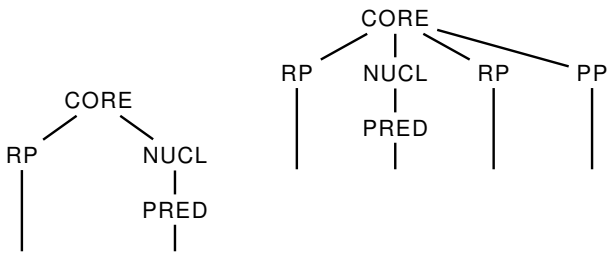
Laura Kallmeyer & Rainer Osswald

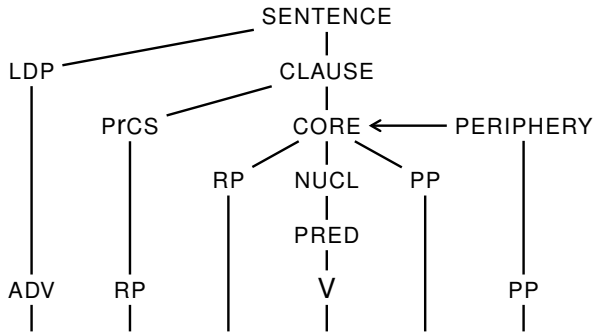
Heinrich-Heine-Universität Düsseldorf

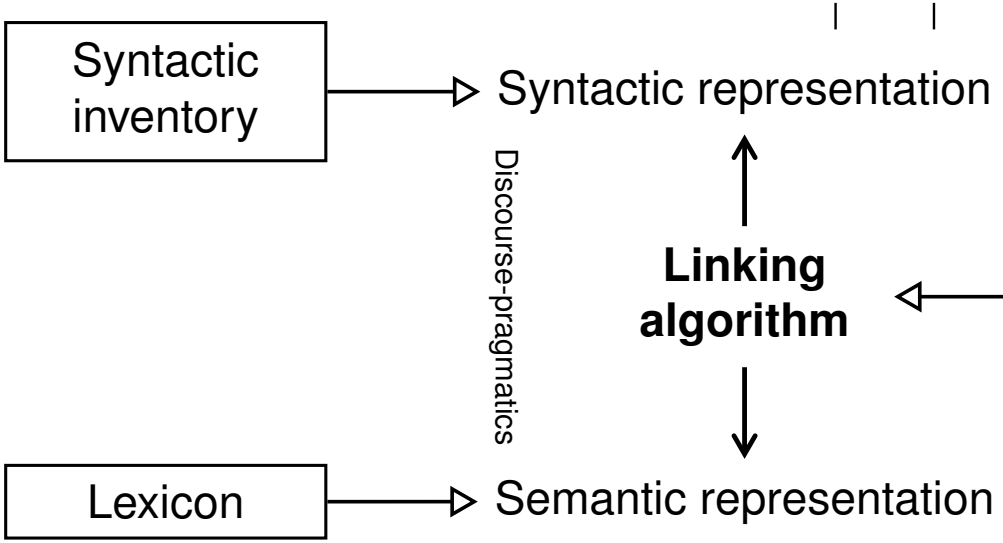
Ask not what RRG can do for you – ask what you can do for RRG.

Ask what you can do for RRG so that RRG can do more for you.

What is RRG (= Role & Reference Grammar) about?







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

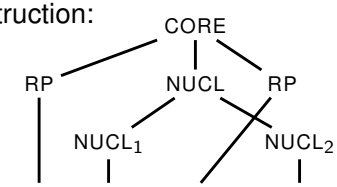
$\langle_{\text{IF}}$ *INT* $\langle_{\text{TNS}}$ *PRES* $\langle_{\text{ASP}}$ *PERF PROG* $\langle$**do**$'$(Kim, [**cry**$'$(Kim)]$\rangle\rangle\rangle\rangle$

MORPHOLOGY —

SYNTAX Juncture: nuclear

Nexus: cosubordination

Construction:



Linking: default

SEMANTICS [SEM$_{\text{NUCL}_1}$] CAUSE [SEM$_{\text{NUCL}_2}$]

PRAGMATICS unspecified

What is FG (= Formal Grammar) about?

- **Formal** (= mathematical/logical) models of grammar

  Precise definition of the set of derivable (tree) structures, …

- **Generative capacity** of grammar formalisms

  Context free languages (but the Swiss!), weakly context sensitive, cross-serial dependencies, copy language, …

- **Complexity** of (parsing, …) **algorithms**

  Polynomial time, …

- **Compositionality** of syntax & semantics

  Montague grammar, λ-calculus, Categorial Grammar, …

Why is an FG perspective on RRG useful (and for whom)?

- Is a formalization relevant for the working typologist?

Why is an FG 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.

Why is an FG 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?

Why is an FG 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.

Why is an FG 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.

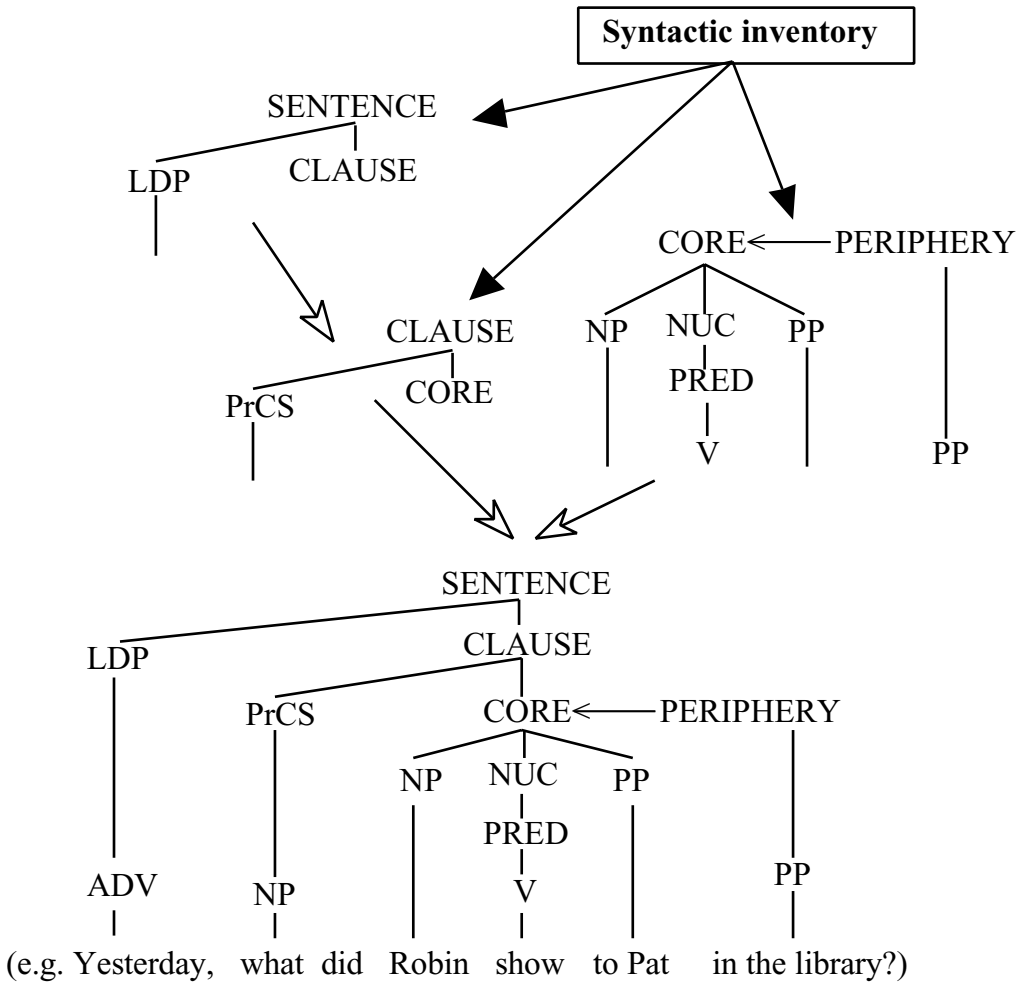
- Further advantages:

  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.

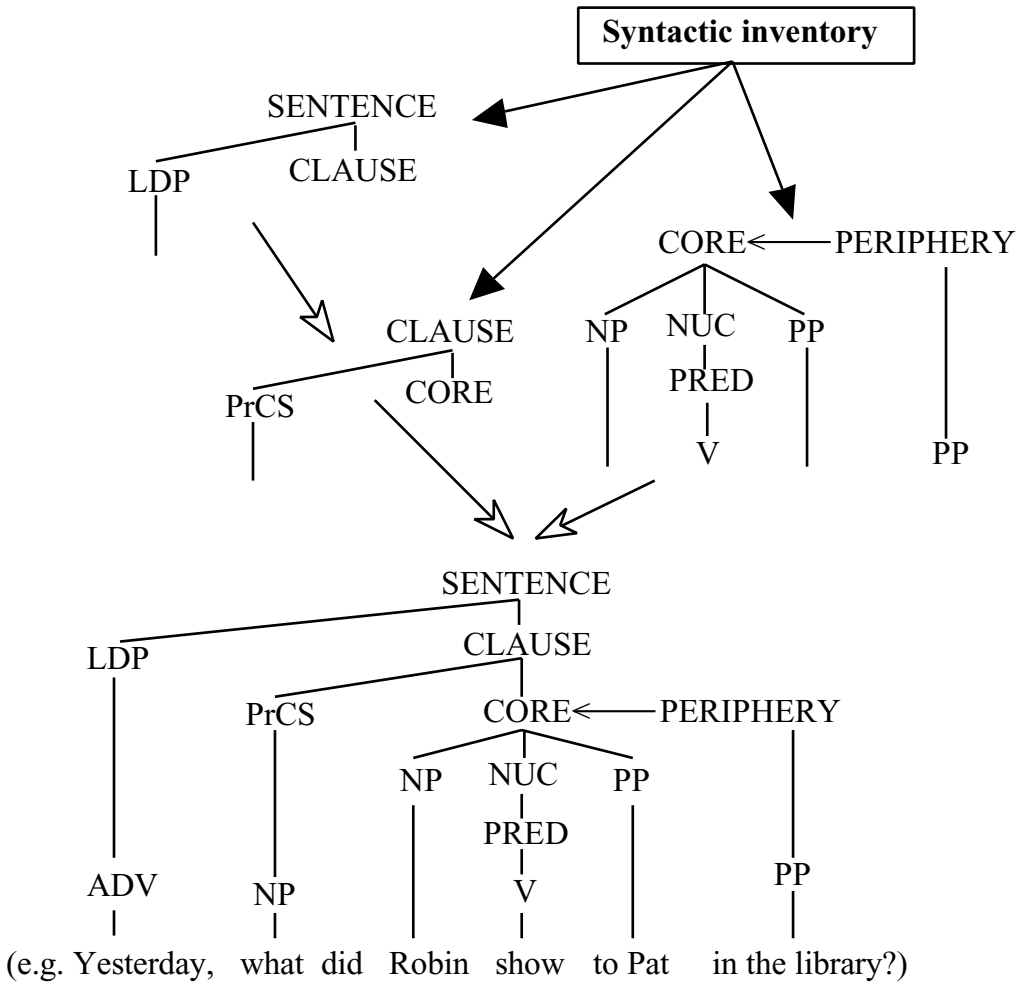
  Moreover, the formalization should make it easier to **extend** and **modify** the theory.

The inventory of syntactic templates



Outline of a formalization of RRG

The inventory of syntactic templates

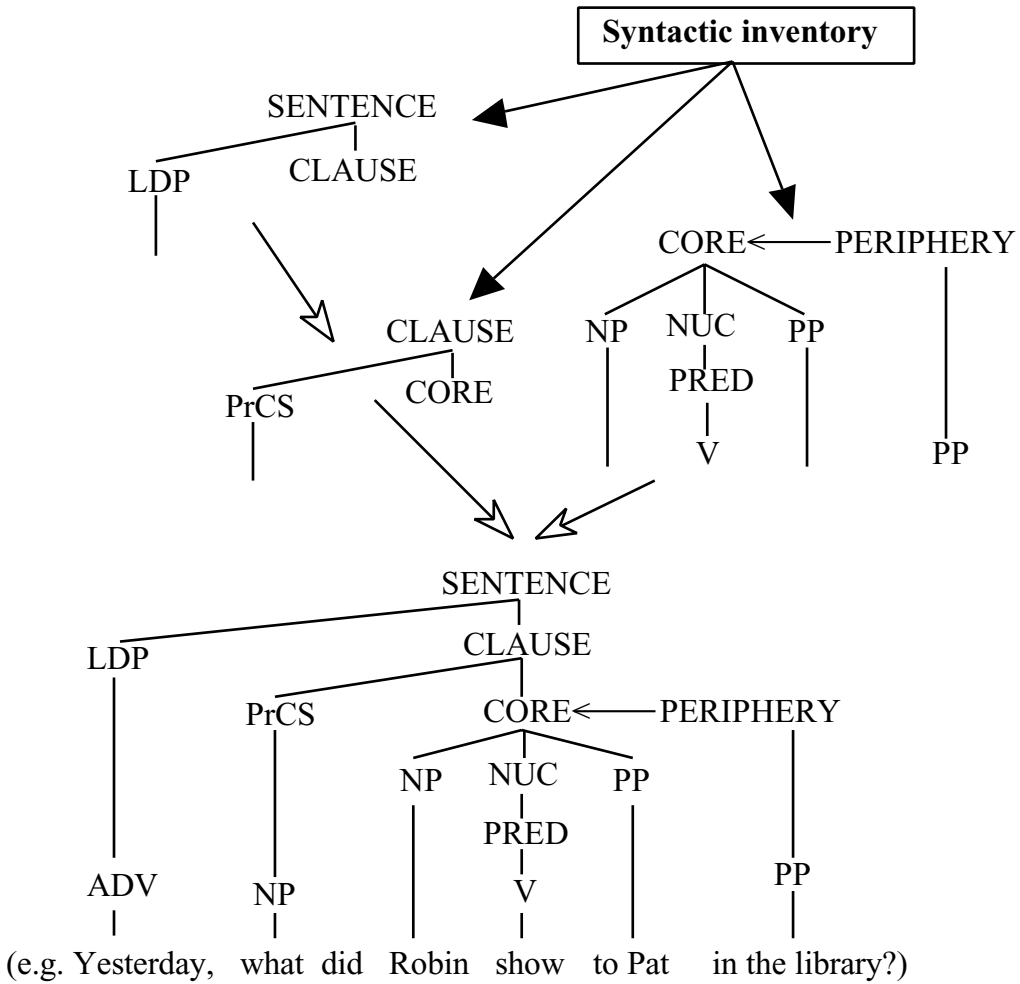


Issues

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

Outline of a formalization of RRG

The inventory of syntactic templates



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

Outline of a formalization of RRG

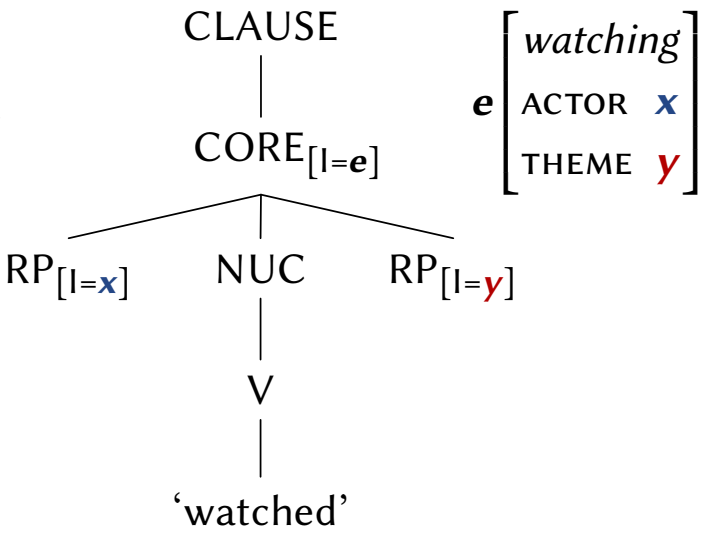
An every-day example

(1) Van watched a match.

Outline of a formalization of RRG

An every-day example

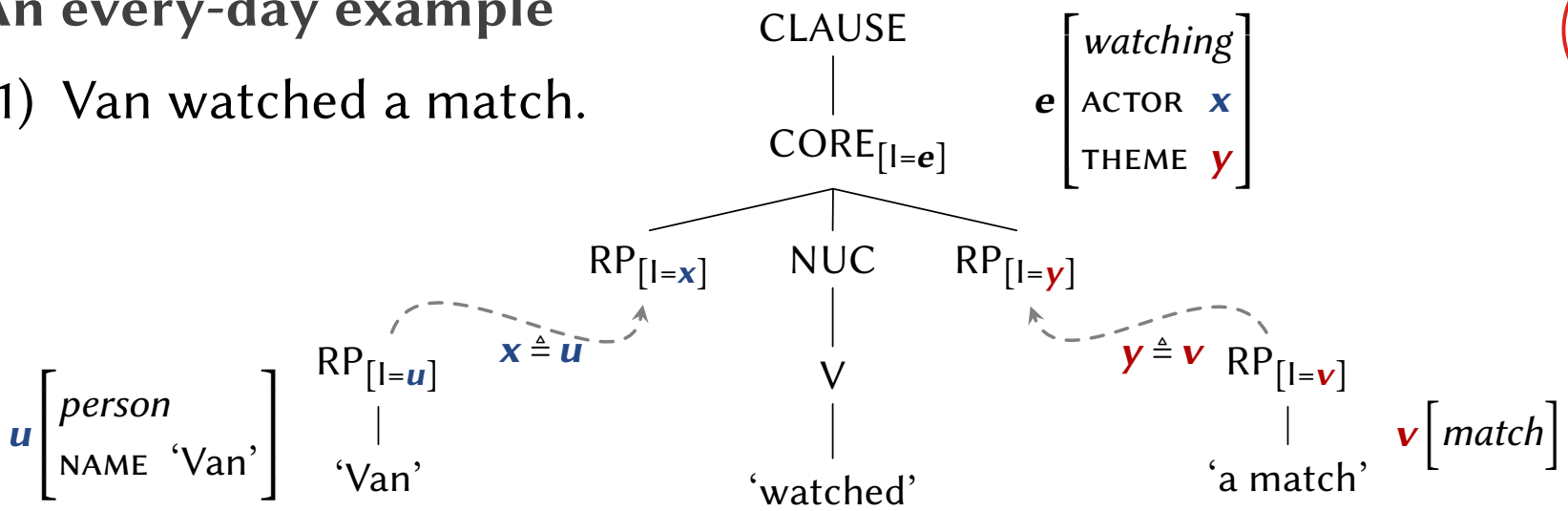
(1) Van watched a match.



Outline of a formalization of RRG

An every-day example

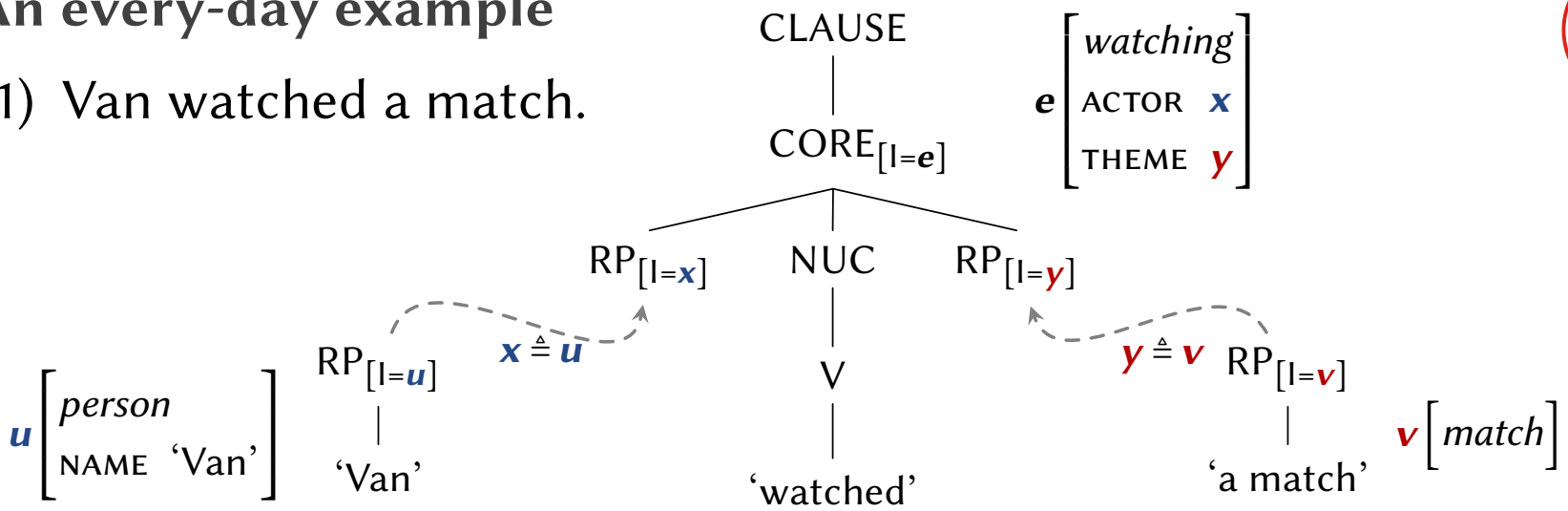
(1) Van watched a match.

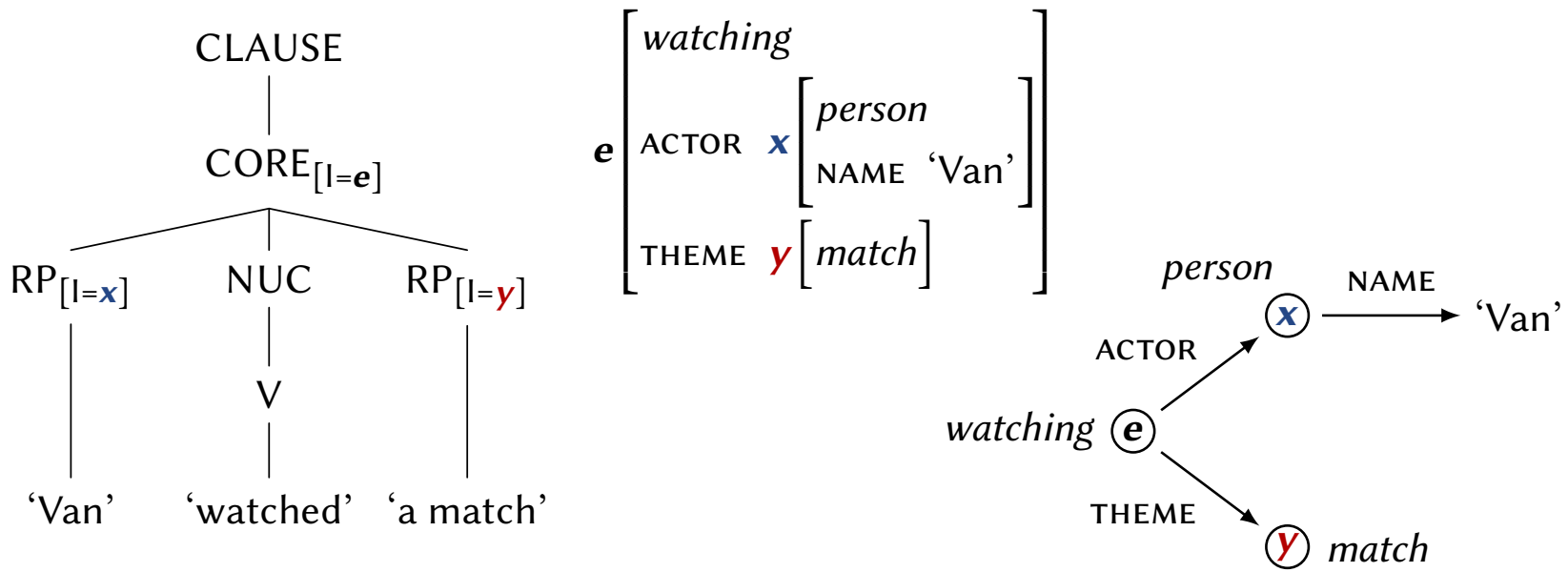


Outline of a formalization of RRG

An every-day example

(1) Van watched a match.





Outline of a formalization of RRG

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

Outline of a formalization of RRG

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**

Outline of a formalization of RRG

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**.

Outline of a formalization of RRG

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.

Outline of a formalization of RRG

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.

"It's the metagrammar where the action is." [≈ Van Valin, p.c.]

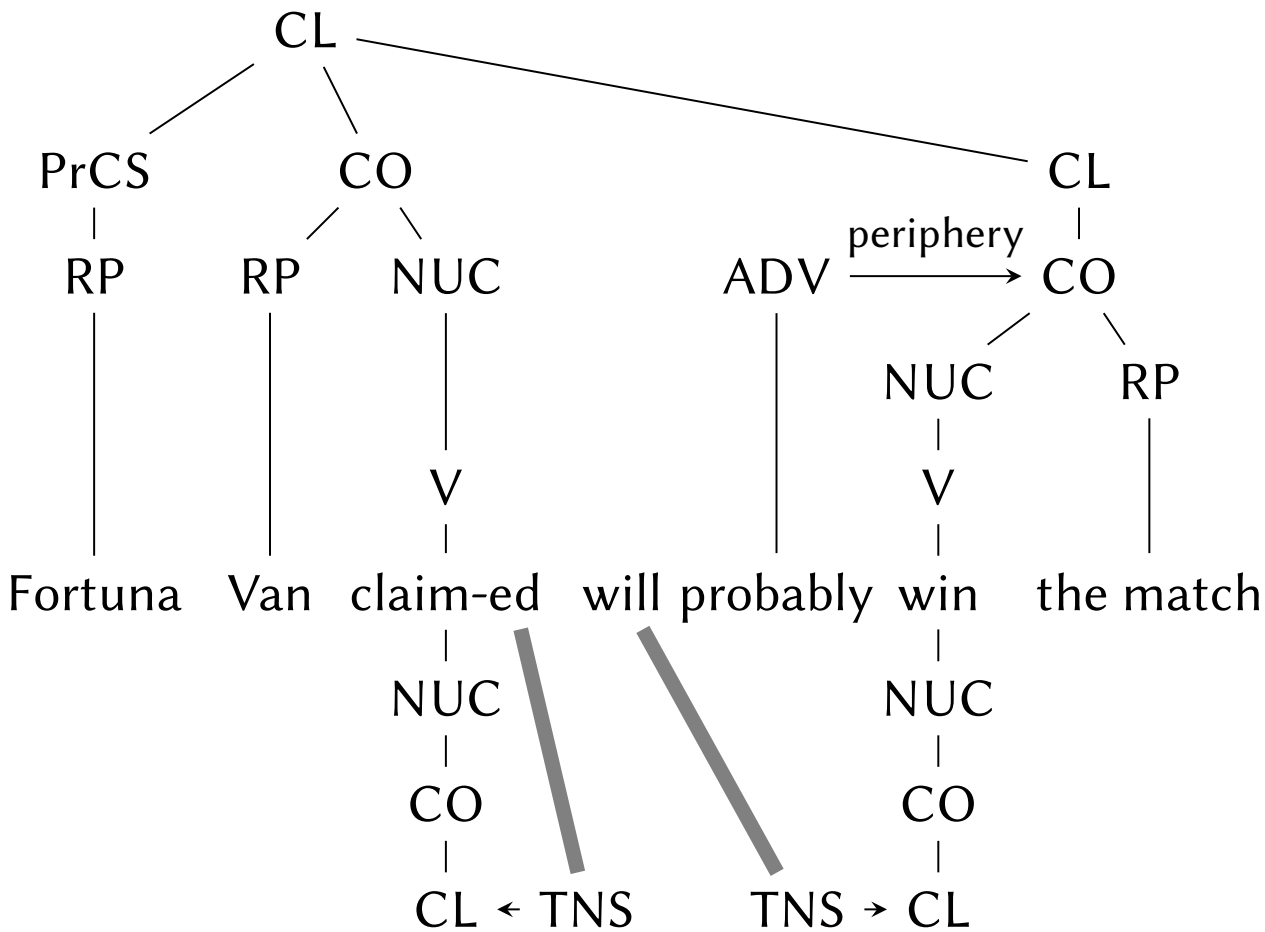
Example (cont'd)

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

Example (cont'd)

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

Syntax:



Example (cont'd)

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

Semantics:

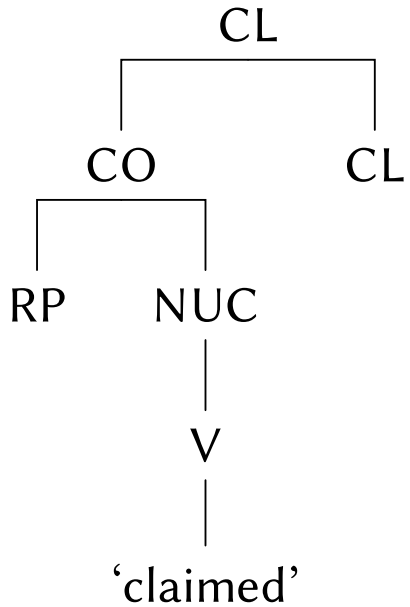
$$\begin{bmatrix} \textit{statement} & \\ \textsc{actor} & \boxed{1}\begin{bmatrix} \textit{person} \\ \textsc{name} \ \text{`Van'} \end{bmatrix} \\ \textsc{speaker} & \boxed{1} \\ \textsc{message} & \begin{bmatrix} \textit{prediction} & \\ \textsc{about} & \begin{bmatrix} \textit{winning} & \\ \textsc{actor} & \begin{bmatrix} \textit{team} \\ \textsc{name} \ \text{`Fortuna'} \end{bmatrix} \\ \textsc{theme} & \begin{bmatrix} \textit{match} \end{bmatrix} \end{bmatrix} \\ \textsc{probability} & \begin{bmatrix} \textit{high} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Questions for the formalization

- What are the elementary building blocks?
- How is the syntactic tree generated?
- What do the funny bold edges in the operator projection mean?
- How are periphery modifiers added to the structure?
- How do we make sure certain parts are obligatory, for instance syntactic arguments but also operators such as TNS?
- How do we link syntax to semantics in such as way as to enable a compositional semantics?

Example: Argument insertion

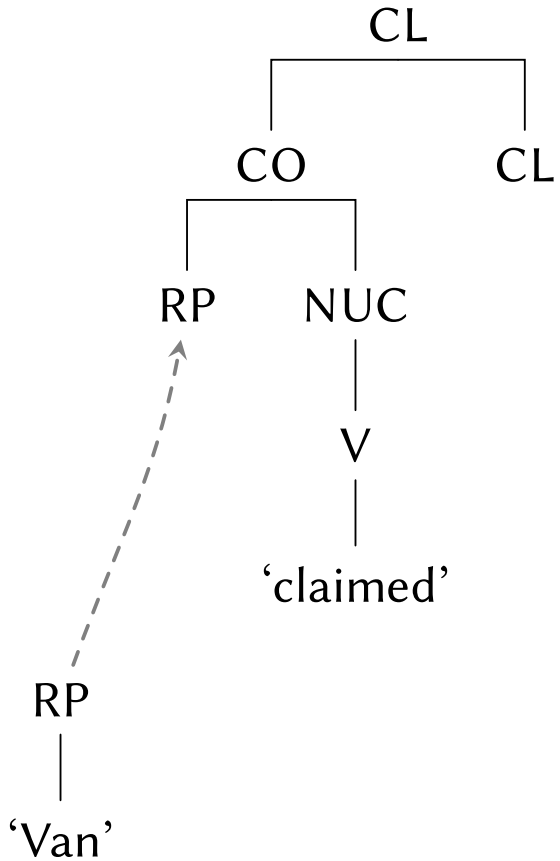
Arguments are added by (wrapping) substitution.



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

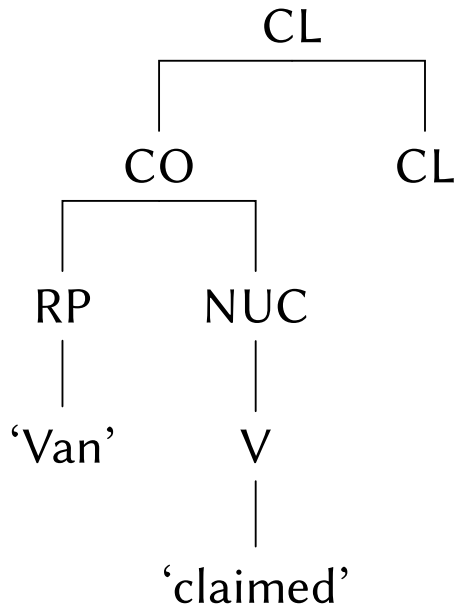
Example: Argument insertion

Arguments are added by (wrapping) substitution.



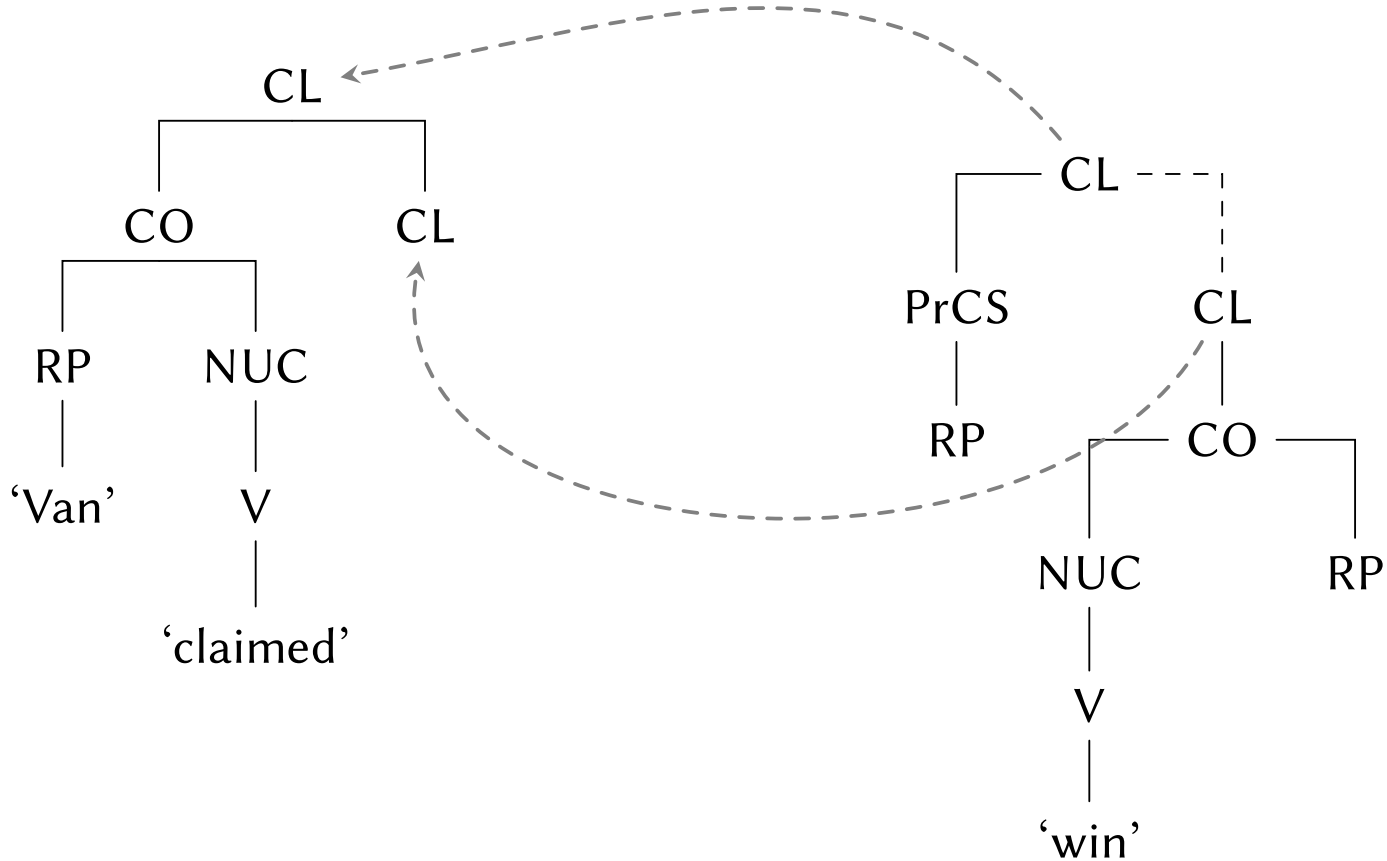
Example: Argument insertion

Arguments are added by (wrapping) substitution.



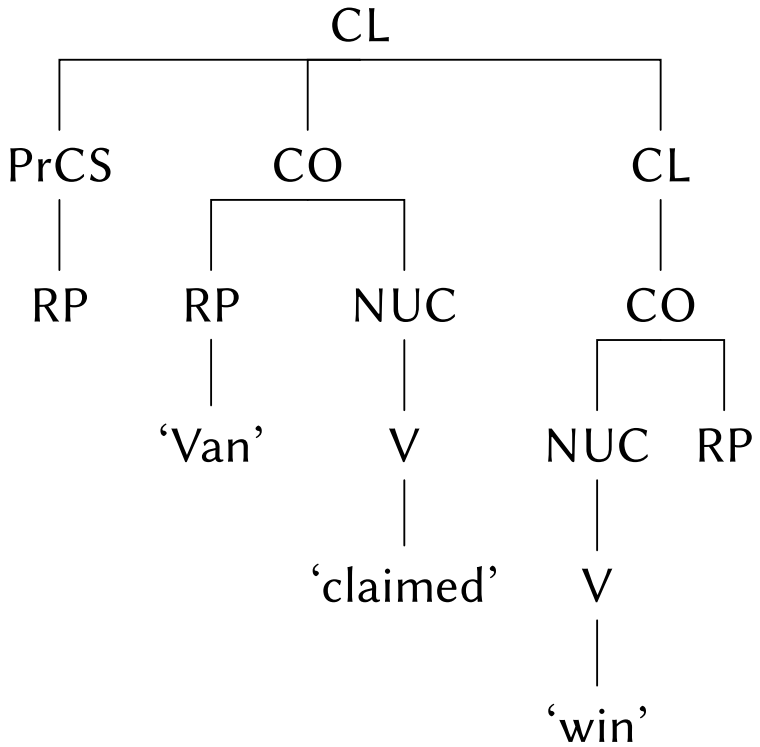
Example: Argument insertion

Arguments are added by (wrapping) substitution.



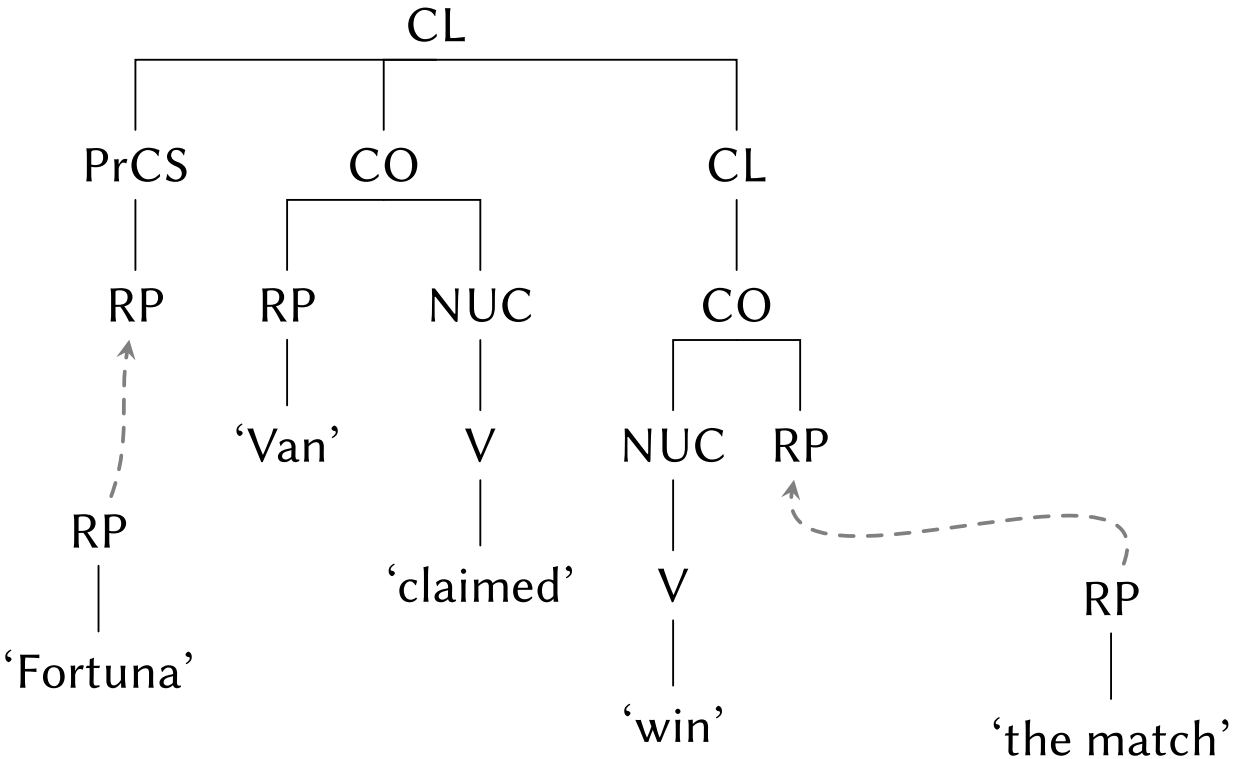
Example: Argument insertion

Arguments are added by (wrapping) substitution.



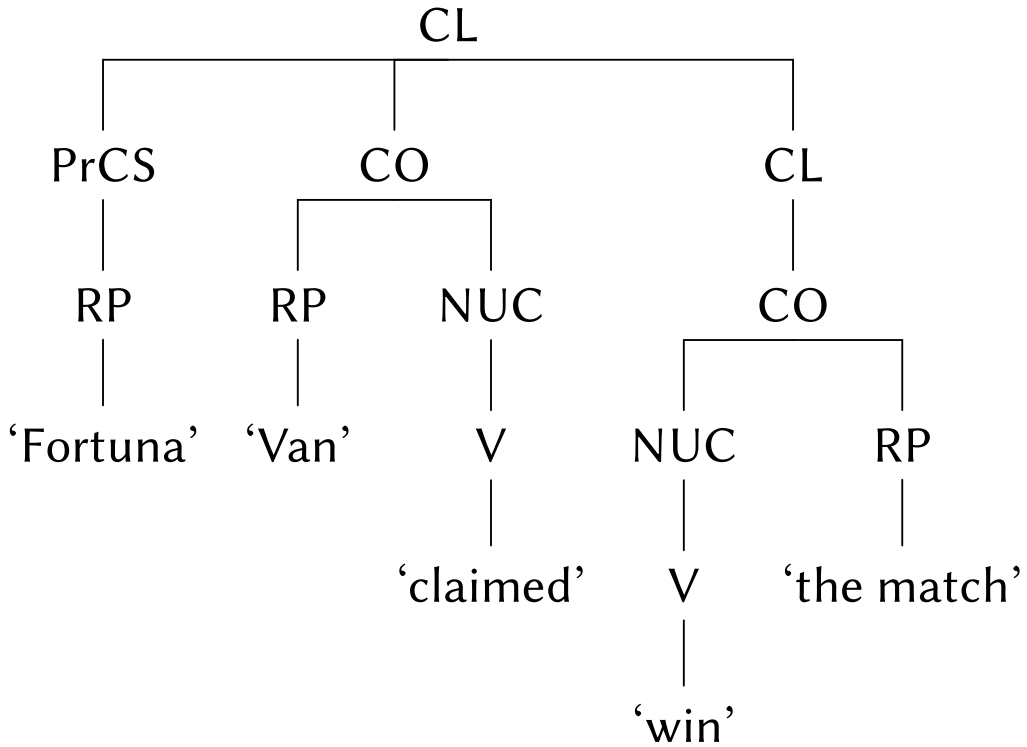
Example: Argument insertion

Arguments are added by (wrapping) substitution.



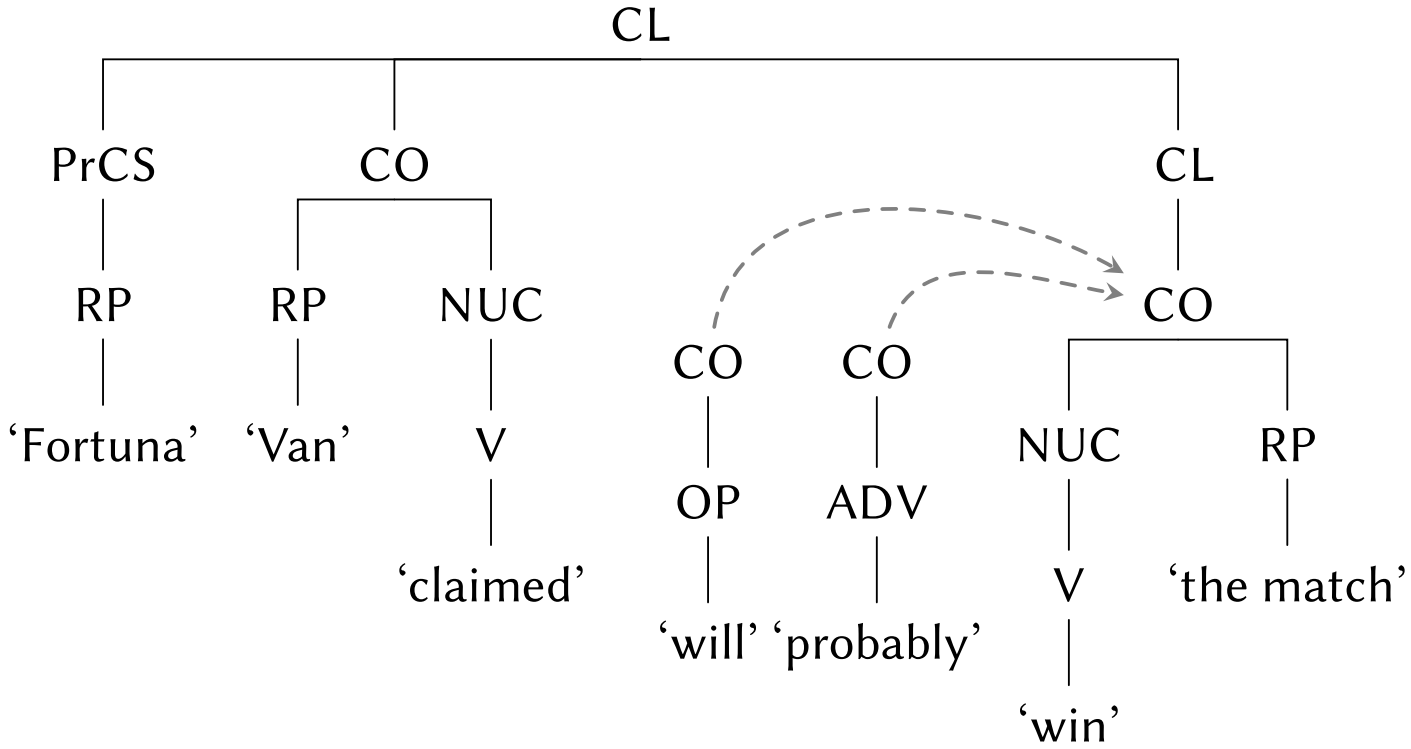
Example: Argument insertion

Arguments are added by (wrapping) substitution.



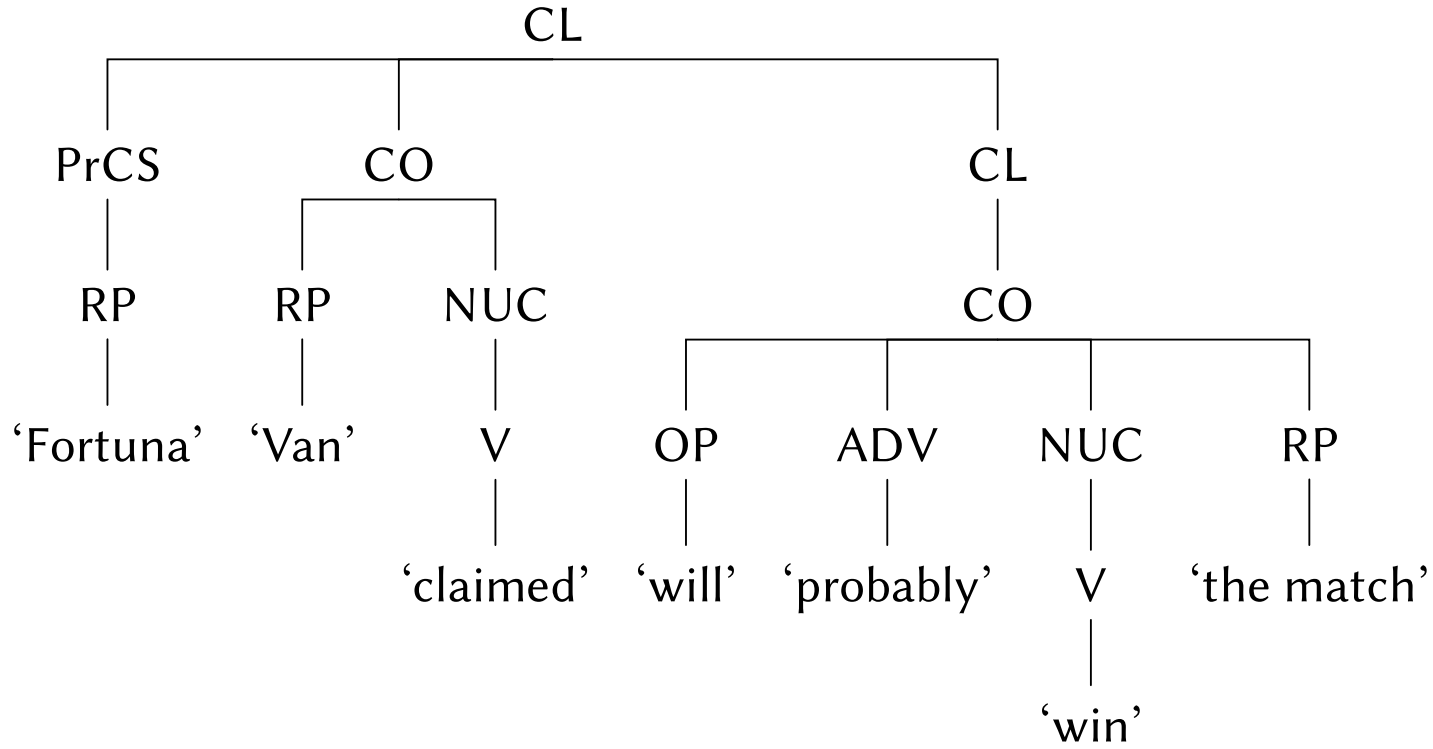
Example: Operators and modifiers

Operators and modifiers are added by sister adjunction.



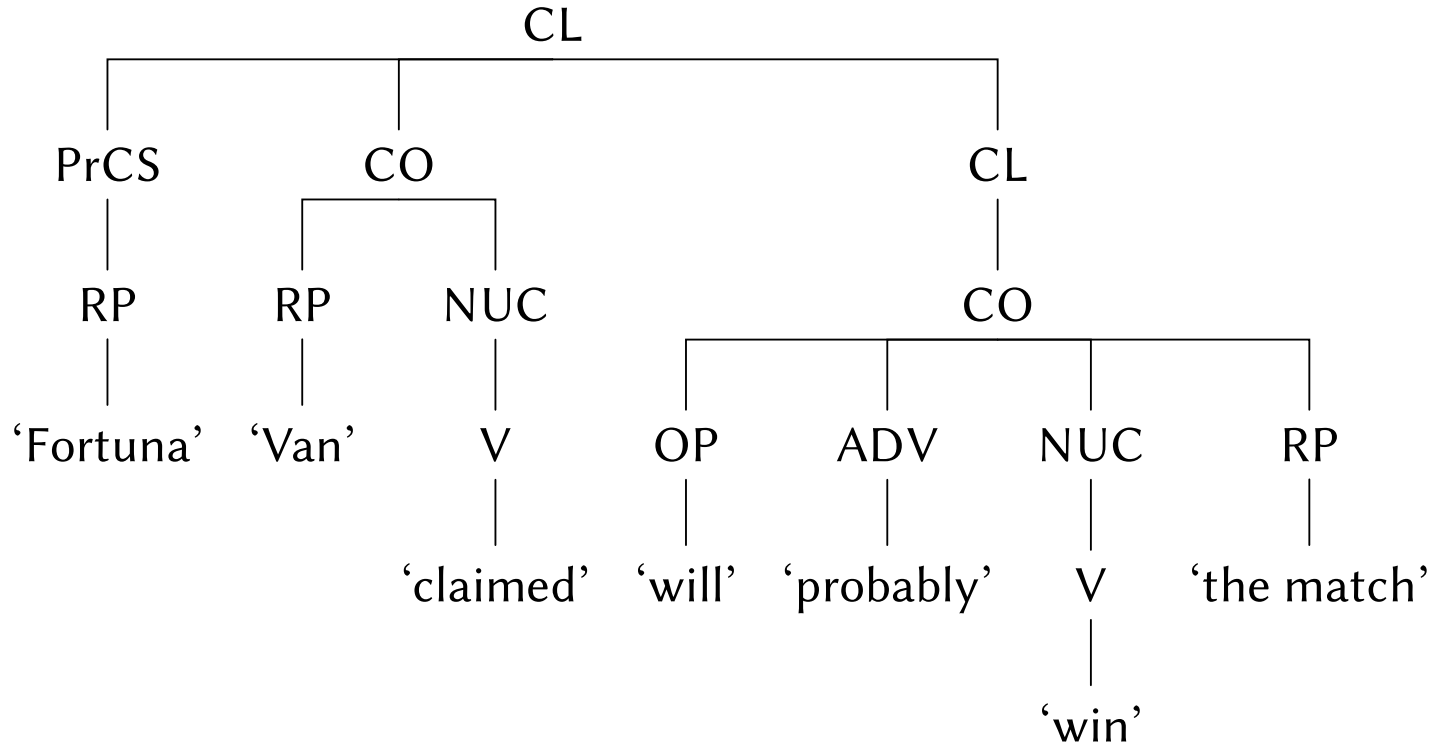
Example: Operators and modifiers

Operators and modifiers are added by sister adjunction.



Example: Operators and modifiers

Operators and modifiers are added by sister adjunction.

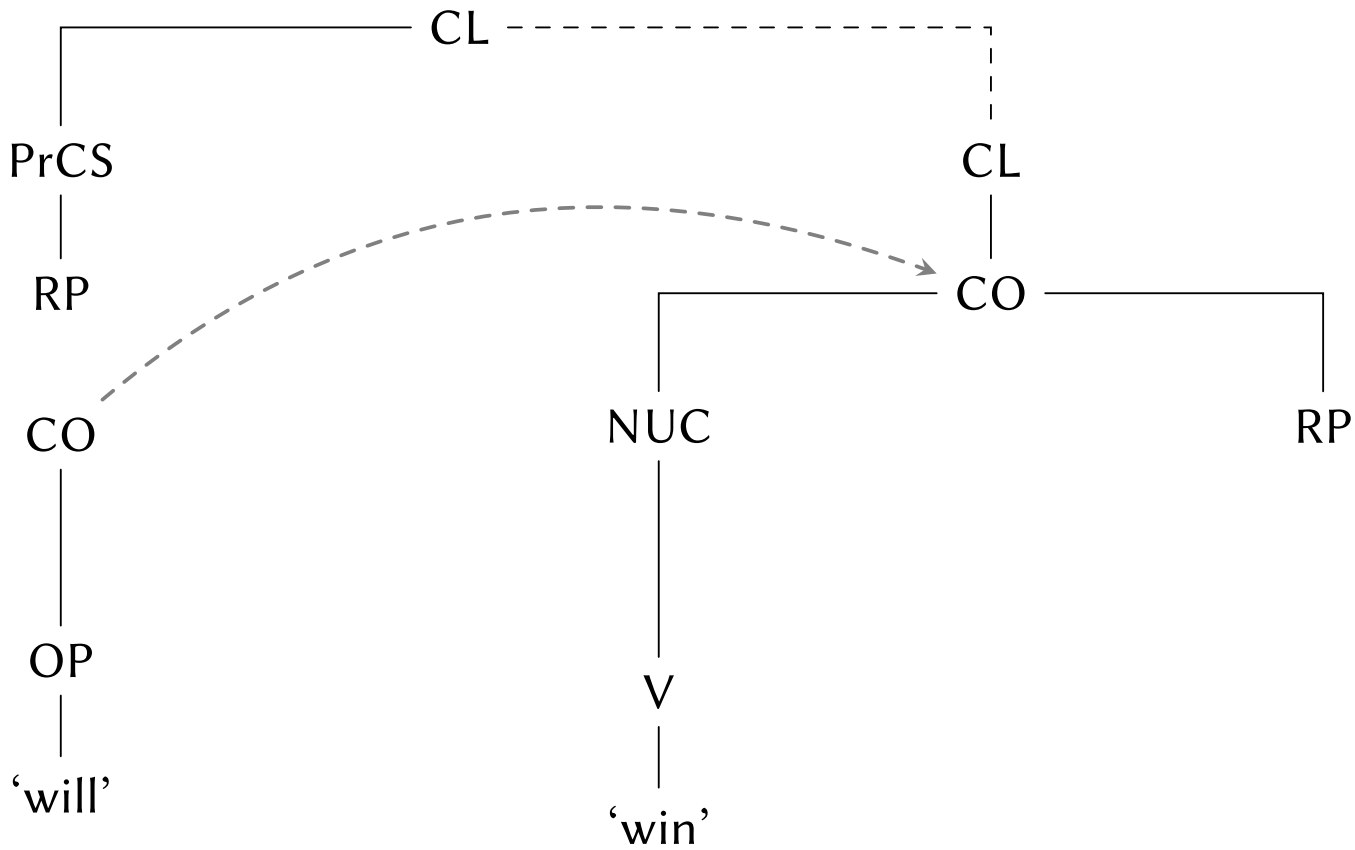


(The operator projection as well as modifier scope is modeled in the features.)

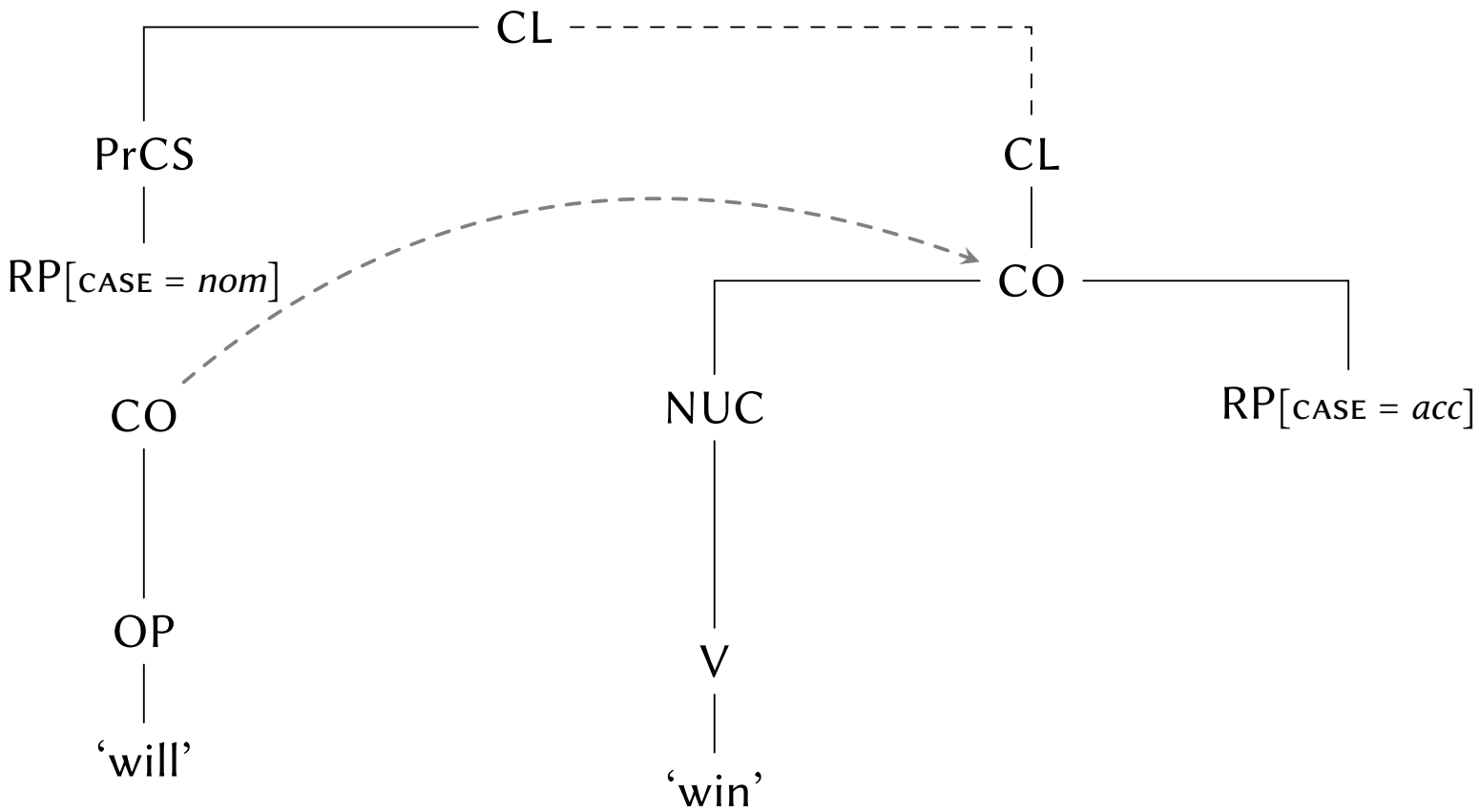
Example: Features

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

Example: Features

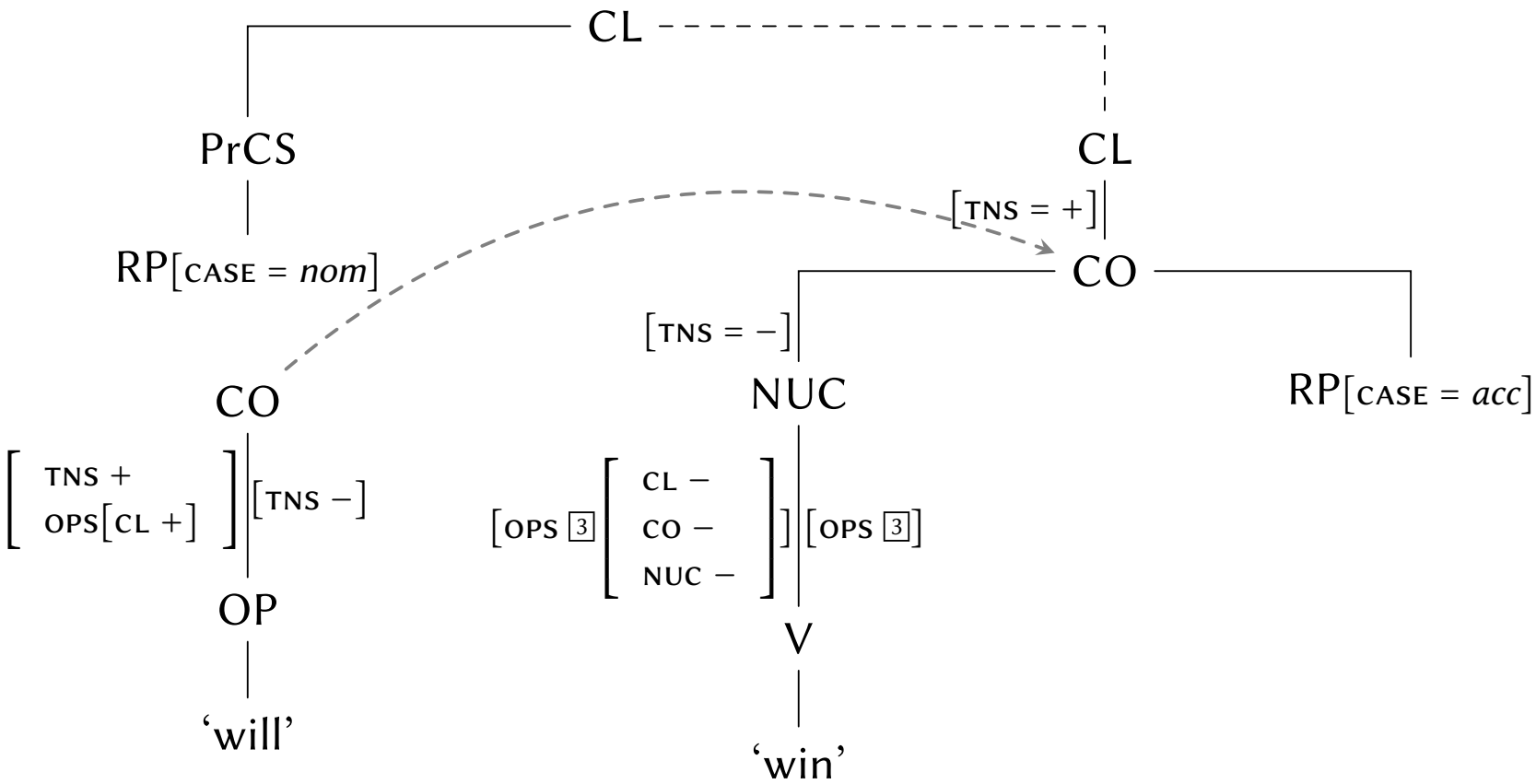


Example: Features



case on nodes for case assignment

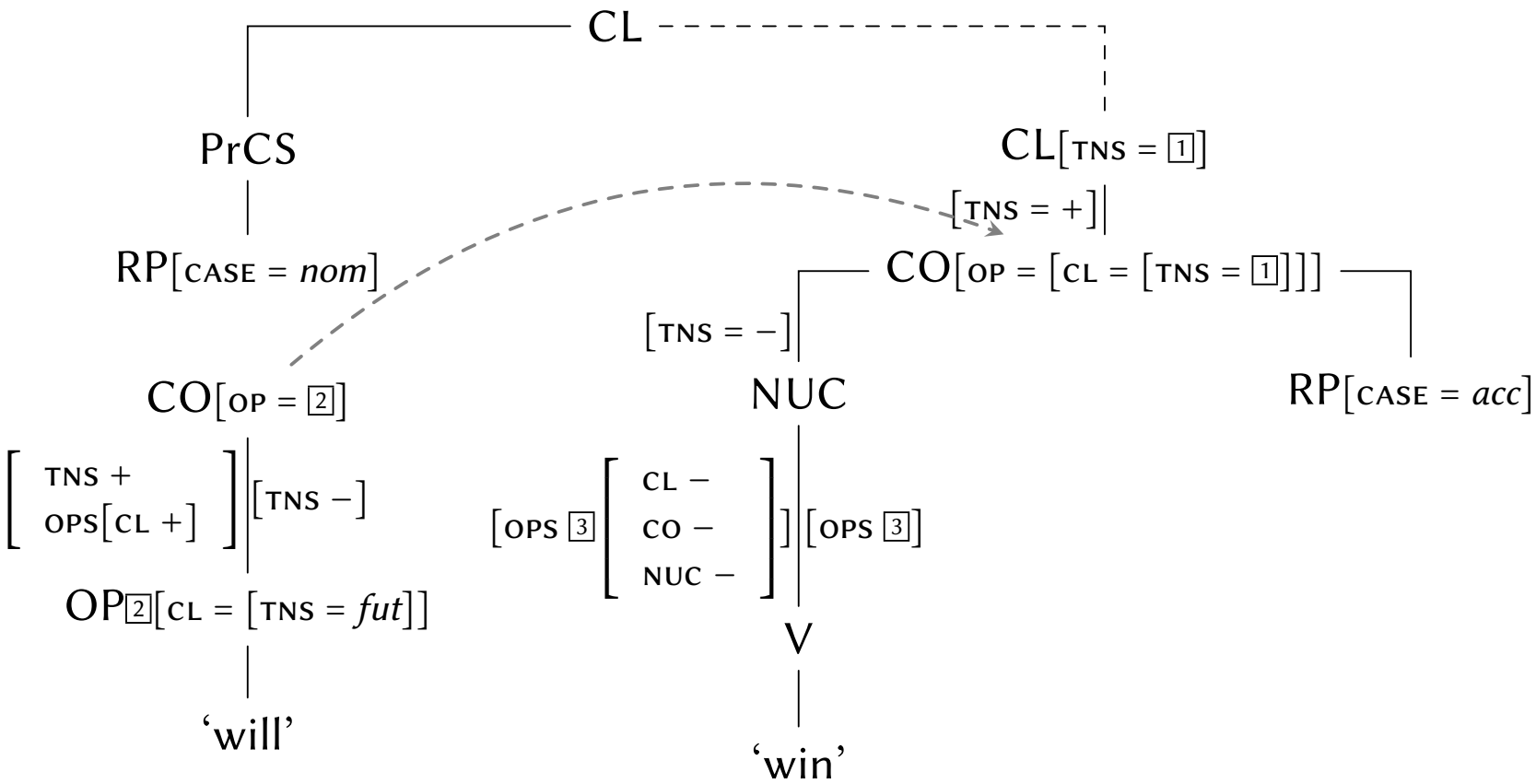
Example: Features



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

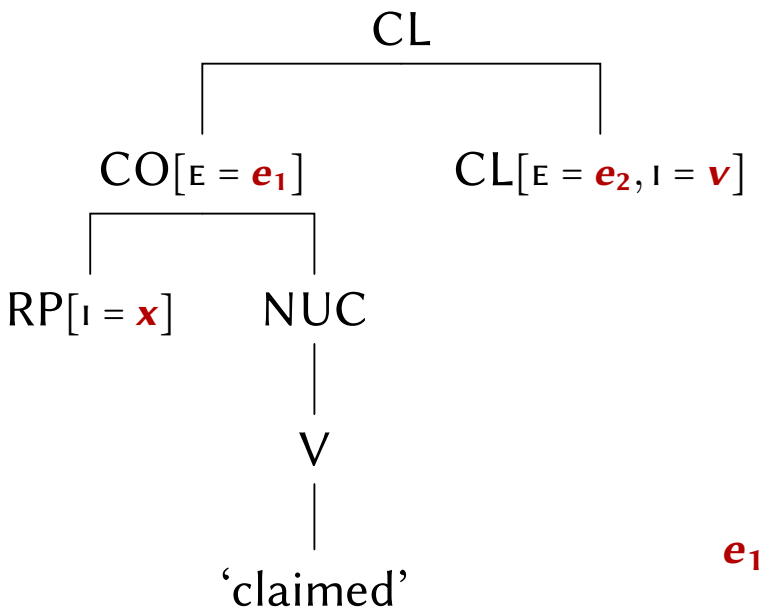
Example: Features



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

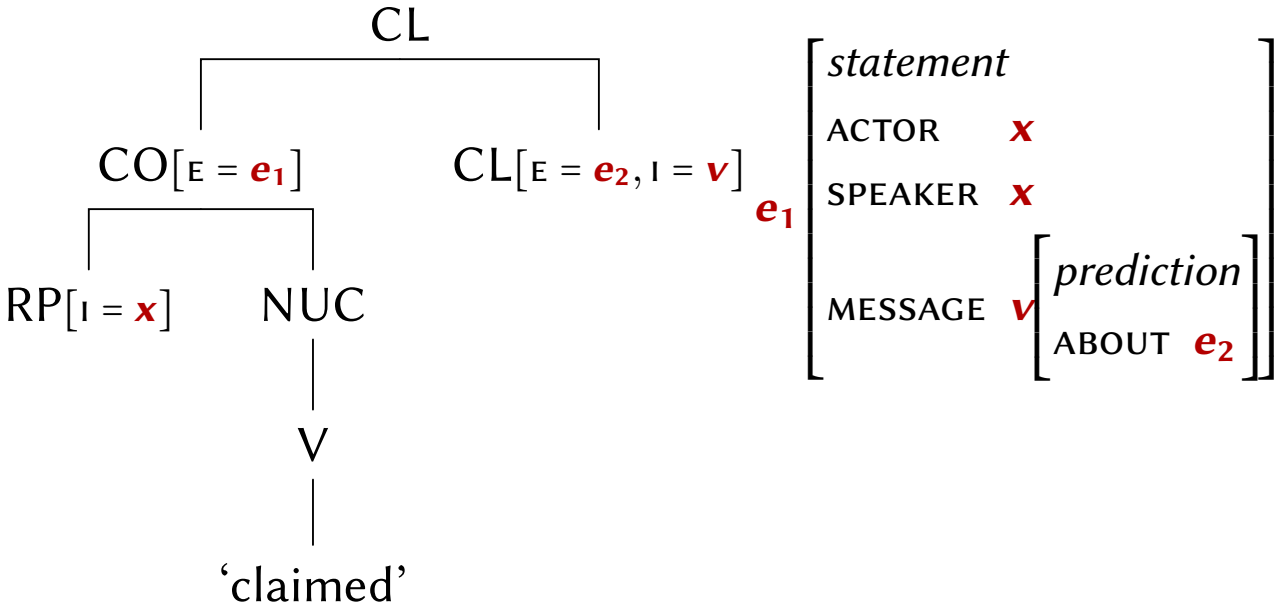
Example: Interfacing syntax and semantics

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

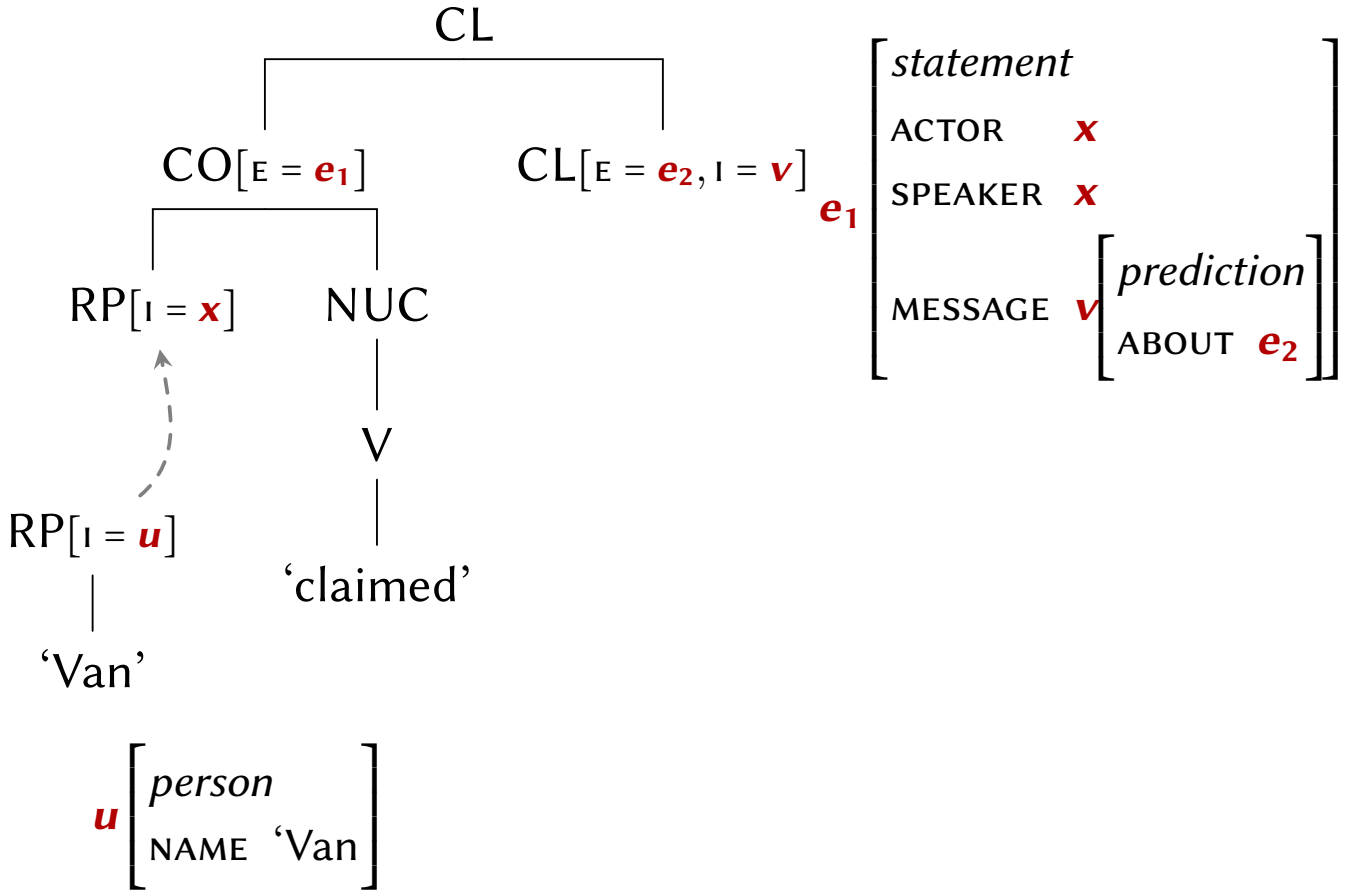


$$e_1 \begin{bmatrix} \textit{statement} \\ \text{ACTOR} \quad x \\ \text{SPEAKER} \quad x \\ \text{MESSAGE} \quad v \begin{bmatrix} \textit{prediction} \\ \text{ABOUT} \quad e_2 \end{bmatrix} \end{bmatrix}$$

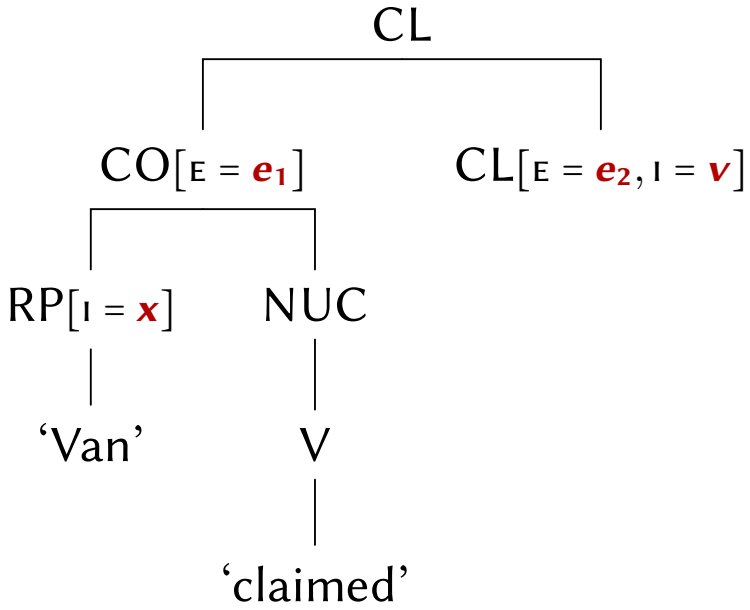
Example: Interfacing syntax and semantics



Example: Interfacing syntax and semantics

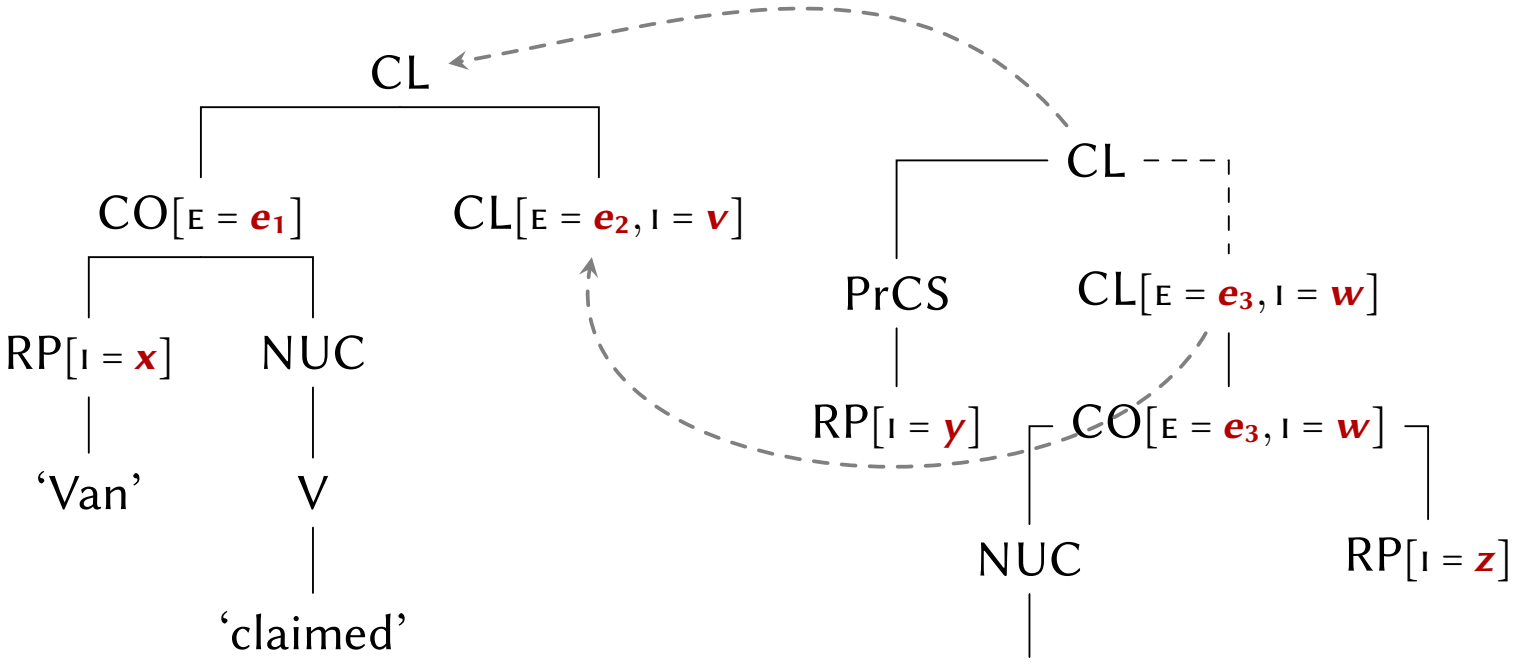


Example: Interfacing syntax and semantics



$$e_1 \begin{bmatrix} \textit{statement} \\ \text{ACTOR} \quad x \begin{bmatrix} \textit{person} \\ \text{NAME} \quad \text{'Van} \end{bmatrix} \\ \text{SPEAKER} \quad x \\ \text{MESSAGE} \quad v \begin{bmatrix} \textit{prediction} \\ \text{ABOUT} \quad e_2 \end{bmatrix} \end{bmatrix}$$

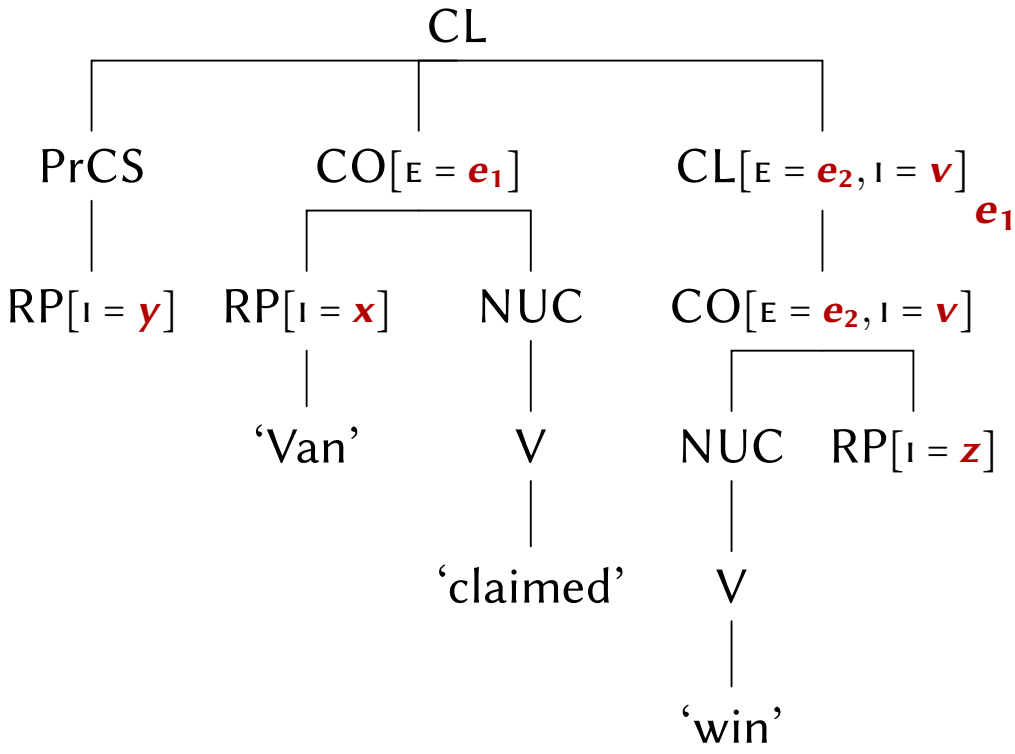
Example: Interfacing syntax and semantics



$$e_1 \begin{bmatrix} \textit{statement} \\ \text{ACTOR} \quad x \begin{bmatrix} \textit{person} \\ \text{NAME} \quad \text{'Van} \end{bmatrix} \\ \text{SPEAKER} \quad x \\ \text{MESSAGE} \quad v \begin{bmatrix} \textit{prediction} \\ \text{ABOUT} \quad e_2 \end{bmatrix} \end{bmatrix}$$

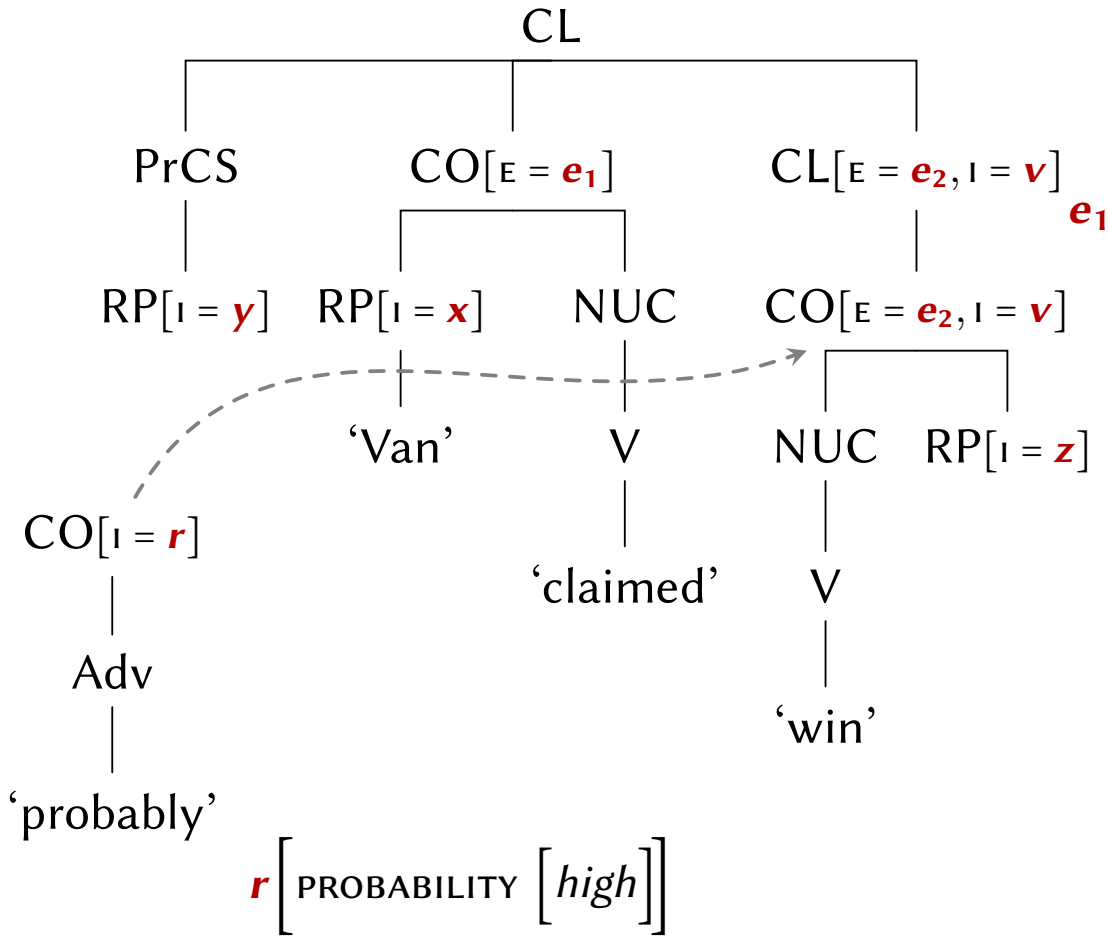
$$e_3 \begin{bmatrix} \textit{winning} \\ \text{ACTOR} \quad y \\ \text{THEME} \quad z \end{bmatrix}$$

Example: Interfacing syntax and semantics



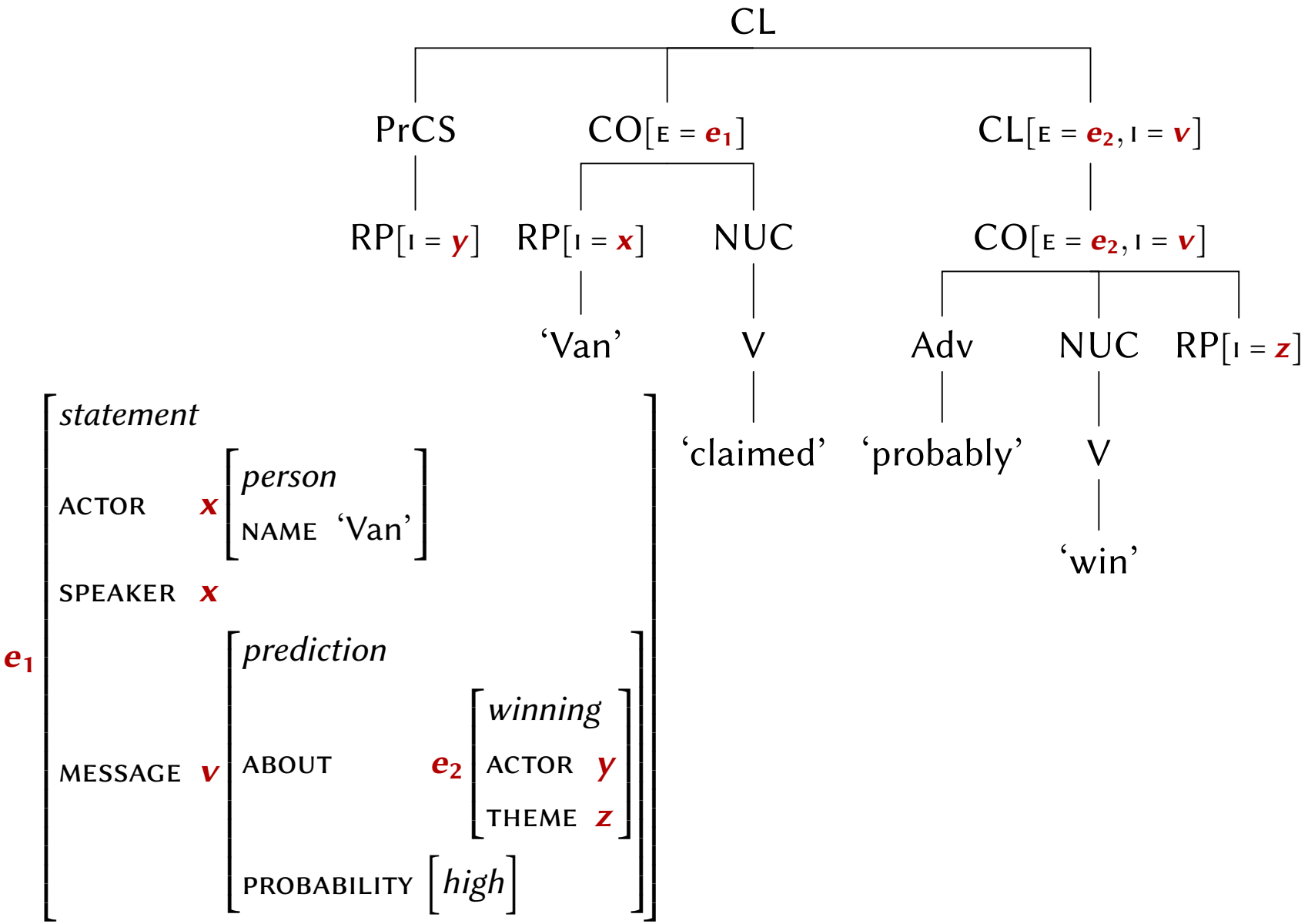
$$
e_1 \begin{bmatrix}
\textit{statement} & \\
\text{ACTOR} & x \begin{bmatrix} \textit{person} \\ \text{NAME} \ \text{`Van} \end{bmatrix} \\
\text{SPEAKER} & x \\
\text{MESSAGE} & v \begin{bmatrix} \textit{prediction} & \\ \text{ABOUT} & e_2 \begin{bmatrix} \textit{winning} \\ \text{ACTOR} \ y \\ \text{THEME}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

Example: Interfacing syntax and semantics



$$e_1 \left[\begin{array}{l} \textit{statement} \\ \text{ACTOR} \quad x \left[\begin{array}{l} \textit{person} \\ \text{NAME} \quad \text{`Van} \end{array} \right] \\ \text{SPEAKER} \quad x \\ \text{MESSAGE} \quad v \left[\begin{array}{l} \textit{prediction} \\ \text{ABOUT} \quad e_2 \left[\begin{array}{l} \textit{winning} \\ \text{ACTOR} \quad y \\ \text{THEME} \quad z \end{array} \right] \end{array} \right] \end{array} \right]$$

Example: Interfacing syntax and semantics



Summary & Conclusion

The good news: RRG can be formalized while retaining its basic outline.

Summary & Conclusion

The good news: RRG can be formalized while retaining its basic outline.

- General composition operations for elementary trees/constructions.
- Contraint-based specification of elementary constructions.
- Linking rules as constraints in the metagrammar.
- …

Summary & Conclusion

The good news: RRG can be formalized while retaining its basic outline.

- General composition operations for elementary trees/constructions.
- Contraint-based specification of elementary constructions.
- Linking rules as constraints in the metagrammar.
- …

The even better news: A lot remains to be done!

- Decision about whether to analyze a given structure (e.g., cosubordination) as a construction or as a composition in the syntax.
- General issue: What is the best methodology for formulating constraints in the metagrammar that capture language-specific and cross-linguistic generalizations in the most appropriate way?
- Formalization of RRG's discourse-pragmatic dimension.
- …

Champions League Finalist !

References

Kallmeyer, Laura. 2016. On the mild context-sensitivity of k-tree wrapping grammar. In Annie Foret et al. (eds.), *Formal Grammar: 20th and 21st International Conferences*, 77–93. Springer.

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)*, 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.