Argument linking in RRG: A constraint-based implementation

Laura Kallmeyer, Timm Lichte & Rainer Osswald

Heinrich-Heine-Universität Düsseldorf

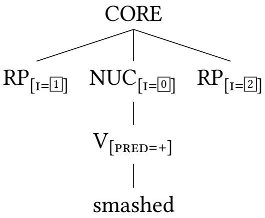
International Conference on Role and Reference Grammar (RRG), Düsseldorf, 02.08.2015

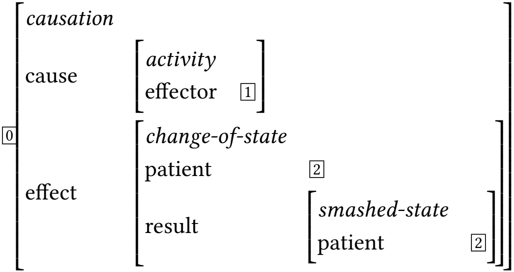
SFB 991

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Introduction

- Overall aim: A thorough formalization and a computational implementation of Role and Reference Grammar (RRG)
- Focus of this talk: Argument linking in simple cores.



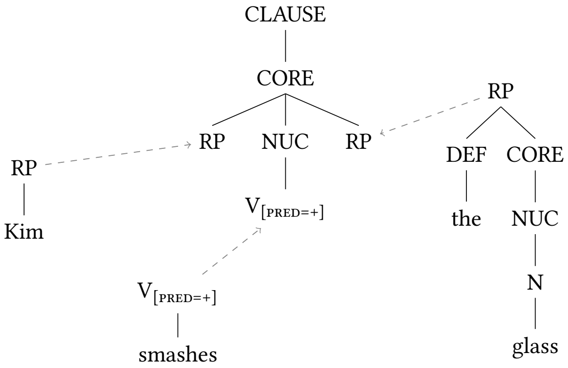


Question: How can we formalize and implement a principled linking theory?

Background

We make use of the following formal and computational elements (along the lines of Kallmeyer et al., 2013; Osswald & Kallmeyer, to appear):

(1) A formal characterization of syntactic composition consisting of substitution, sister adjunction, and wrapping substitution. For this talk, only substitution is relevant.



Background

(2) A formal specification of the elementary syntactic templates to which these composition operations apply.
Assumption: argument structure templates are clause templates with slots for each of the arguments occurring in the clause.
These syntactic templates are specified via a metagrammar.

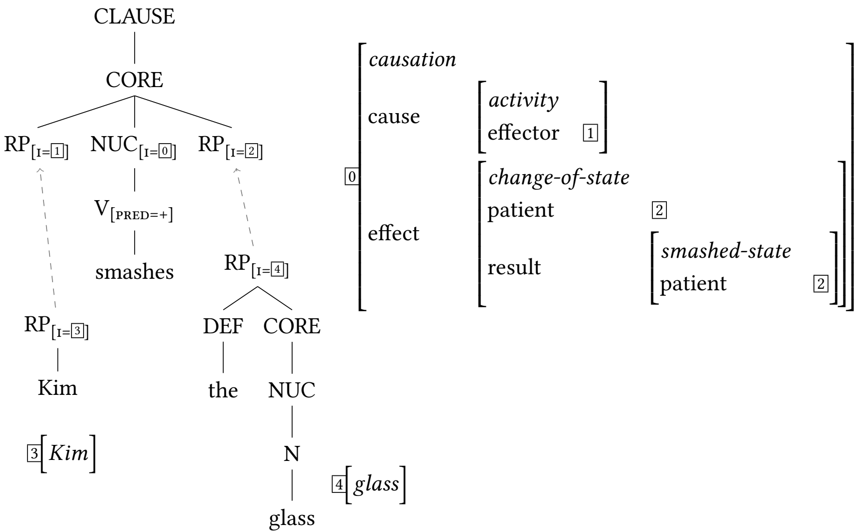
(3) A formalization of the logical structure in terms of decompositional frames (= typed feature structures).

$$[\mathbf{do}(\mathit{Kim}, \varnothing)]\ \text{CAUSE}\ [\text{BECOME}\ \mathbf{smashed}(\mathit{glass})]$$

$$\boxed{0}\begin{bmatrix} \mathit{causation} & \\ \text{cause} & \begin{bmatrix} \mathit{activity} & \\ \text{effector} & \boxed{1}\begin{bmatrix}\mathit{Kim}\end{bmatrix} \end{bmatrix} \\ \text{effect} & \begin{bmatrix} \mathit{change\text{-}of\text{-}state} & \\ \text{patient} & \boxed{2}\begin{bmatrix}\mathit{glass}\end{bmatrix} \\ \text{result} & \begin{bmatrix} \mathit{smashed\text{-}state} & \\ \text{patient} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Background

Syntactic and semantic composition:



Background

(4) A specification of linking constraints that capture the way syntactic arguments and semantic roles are systematically related. These constraints are part of the metagrammar, together with syntactic tree fragments and fragments of semantic frames.

- Constraint-based formulation of the principles underlying the linking algorithm from Van Valin (2005) instead of a procedural specification in the form of an algorithm.
- Advantage: we can separate between the linguistic generalizations to be captured and algorithmic considerations.
- We use the XMG grammar development system (Lichte & Petitjean, 2015) for the implementation of (2)–(4).

Constraints on event frames

- First element of our linking system:
  Universal constraints on semantic roles and macroroles.
- Constraints can be of the form (Kallmeyer & Osswald, 2013)
  - *path* : *type* (the value of *path* is of type *type*)
  - $path_1 \doteq path_2$ ($path_1$ and $path_2$ yield the same value)
  - $constraint_1 \preceq constraint_2$ ($constraint_1$ implies $constraint_2$)

Some examples:

- MOVER : ⊤ ⪯ MOVER ≐ EFFECTOR
  (a mover is always an effector)
- CAUSE EFFECTOR : ⊤ ⪯ EFFECTOR ≐ CAUSE EFFECTOR
  (if the causing event of a causation has an effector, then this effector is also the effector of the embedding causation event)
- EFFECTOR : ⊤ ⪯ EFFECTOR ≐ ACTOR
  (if an effector is given, then the effector is also the actor)
- PATIENT : ⊤ ⪯ PATIENT ≐ UNDERGOER
  (a patient is always an undergoer)

MG classes for semantic arguments

Second element of our linking system:
Classes for semantic arguments of rank 1, 2, 3 or 4 respectively

Arg. of DO	1st arg. of **do′** (x, ...	1st. arg. of **pred′** (x,y)	2nd arg. of **pred′** (x,y)	Arg. of state **pred′** (x)
AGENT	EFFECTOR	LOCATION	THEME	PATIENT
	MOVER	PERCEIVER	STIMULUS	ENTITY
	ST-MOVER	COGNIZER	CONTENT	
	L-EMITTER	WANTER	DESIRE	
	S-EMITTER	JUDGER	JUDGMENT	
	PERFORMER	POSSESSOR	POSSESSED	
	CONSUMER	EXPERIENCER	SENSATION	
	CREATOR	EMOTER	TARGET	
	OBSERVER	ATTRIBUTANT	ATTRIBUTE	
	USER	IDENTIFIED	IDENTITY	
		VARIABLE	VALUE	
			PERFORMANCE	
			CONSUMED	
			CREATION	
			IMPLEMENT	

(Fig. 2.3 from Van Valin, 2005, p. 58)

MG classes for semantic arguments

class *ArgRank1*

?e0 [effector ?x1]

class *ArgRank2*

?e0 [location|wanter|judger|... ?x2]

class *ArgRank3*

?e0 [theme|stimulus|content|... ?x3]

class *ArgRank4*

?e0 [patient|... ?x4]

(?e0, ?x1, ... are metagrammar variables)

Macrorole assignment

- Goal: For combinations of these argument classes, we want to obtain the correct macrorole assignment.
- Due to the universal constraints, we already have that 1) if a rank 1 argument is present, it is the actor and 2) if a rank 4 argument is present, it is the undergoer.

Two possible solutions:

1. List all possible argument combinations and specify the corresponding marcorole assignment (see abstract).
2. Further factorization of these combinations using a variable for the highest rank and binary variables giving information about the rank of the undergoer.

Macrorole assignment

First solution yields for instance the classes

class *Pred_Rank2*
ArgRank2, ?x2 = ?promi,

$?e0\begin{bmatrix}\text{actor} & ?x2\end{bmatrix}$

class *Pred_Rank1_2*
ArgRank1, *ArgRank2*,
?x1 = ?promi, ?x2 = ?b-promi

$?e0\begin{bmatrix}\text{undergoer} & ?x2\end{bmatrix}$

?promi and ?b-promi are interface variables:

- ?promi is the argument with the highest rank,
- ?b-promi the one with the lowest, provided there is a higher one.

Macrorole assignment

Second solution: We use

- a variable ?highest for the highest argument rank,
- boolean variables ?und_lower_2 and ?und_lower_3, indicating whether the undergoer has a rank lower than 2 (resp. 3), and
- variables ?r1, . . . , ?r4 giving the argument of rank 1, . . . , 4 respectively.

We add ?ri = ?xi to each of the previous argument classes.

For instance: class *ArgRank1*
?r1 = ?x1, ?e0[effector ?x1]

For each rank, we introduce new classes for arguments of this rank a) being an actor, b) being an undergoer, c) without macrorole and d) not realized (if applicable).

Macrorole assignment

class *Rank2_actor*
ArgRank2, ?highest = 2, ?promi = ?x2, ?e0$\begin{bmatrix}\text{actor} & \text{?x2}\end{bmatrix}$

class *Rank2_undergoer*
ArgRank2, ?highest = 1, ?b-promi = ?x2, ?und_lower_2 = false,
?e0$\begin{bmatrix}\text{undergoer} & \text{?x2}\end{bmatrix}$

class *Rank2_no_macrorole*
ArgRank2, ?highest = 1, ?und_lower_2 = true

class *Rank2_no_argument*
?r2 = nil

Macrorole assignment

Class comprising all these alternatives:

class *Rank2*
Rank2_actor
OR *Rank2_undergoer*
OR *Rank2_no_macrorole*
OR *Rank2_no_argument*

Final class describing all possible semantic frames:

class *Event_Frame*
Rank1, *Rank2*, *Rank3*, *Rank4*

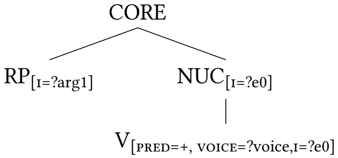
XMG compiles this into all possible combinations, creating thereby all possible event frames.

Linking syntax and semantics

- Within the metagrammar, elementary syntactic templates are decomposed.
- The variables ?promi, ?b-promi, ?r1, ?r2, ?r3, ?r4 can be used as interface variables.

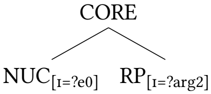
Language-specific classes for English:

class *Privileged_syn_arg*



(?voice = active, ?promi = ?arg1) OR
(?voice = passive, ?b-promi=?arg1)

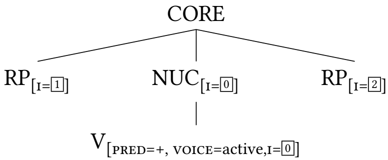
class *Obj_RP*



?r2=?arg2
OR ?r3=?arg2
OR ?r4=?arg2

Linking syntax and semantics

Putting together these two classes and the class *Event_Frame* yields among others:



$$\boxed{0}\begin{bmatrix} \textit{event} & \\ \text{actor} & \boxed{1} \\ \text{undergoer} & \boxed{2} \\ \text{effector} & \boxed{1} \\ \text{patient} & \boxed{2} \end{bmatrix}$$

Interface variables:
?promi=$\boxed{1}$, ?b-promi=$\boxed{2}$,
?r1=$\boxed{1}$, ?r2 = nil, ?r3 = nil, ?r4=$\boxed{2}$

This can then be combined with the lexical entry of the verb.

Linking syntax and semantics

Lexical entry of *smashed*:

$$\begin{array}{c} \text{V}_{[\textsc{pred}=+,\ \textsc{voice}=\text{active},\textsc{i}=\boxed{3}]} \\ | \\ \text{smashed} \end{array} \qquad \boxed{3}\begin{bmatrix} \textit{causation} & \\ \text{cause} & \begin{bmatrix} \textit{activity} & \\ \text{effector} & \boxed{4} \end{bmatrix} \\ \text{effect} & \begin{bmatrix} \textit{change-of-state} & \\ \text{patient} & \boxed{5} \\ \text{result} & \begin{bmatrix} \textit{smashed-state} & \\ \text{patient} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

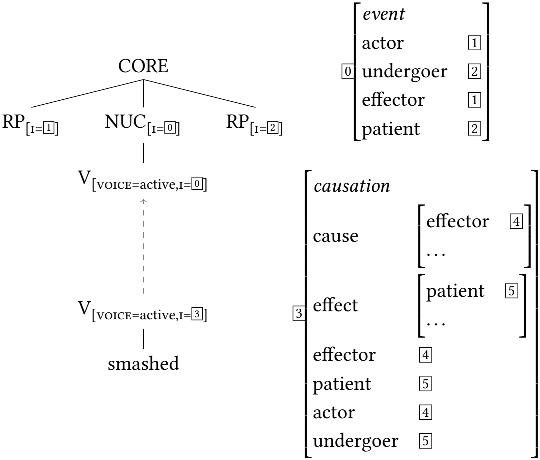
Interface variables: ?r1=$\boxed{4}$, ?r2 = nil, ?r3 = nil, ?r4=$\boxed{5}$

The universal constraints on event frames yield:

$$\boxed{3}\begin{bmatrix} \textit{causation} & \\ \text{cause} & [\ldots] \\ \text{effect} & [\ldots] \\ \text{effector} & \boxed{4} \\ \text{patient} & \boxed{5} \\ \text{actor} & \boxed{4} \\ \text{undergoer} & \boxed{5} \end{bmatrix}$$

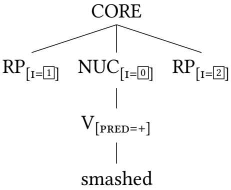
Linking syntax and semantics

Combining the template with the lexical predicate:



Linking syntax and semantics

Result:



$$\boxed{0}\begin{bmatrix} \textit{causation} & \\ \text{actor} & \boxed{1} \\ \text{undergoer} & \boxed{2} \\ \text{effector} & \boxed{1} \\ \text{patient} & \boxed{2} \\ \text{cause} & \begin{bmatrix} \textit{activity} & \\ \text{effector} & \boxed{1} \\ \text{actor} & \boxed{1} \end{bmatrix} \\ \text{effect} & \begin{bmatrix} \textit{change-of-state} & \\ \text{patient} & \boxed{2} \\ \text{undergoer} & \boxed{2} \\ \text{result} & \begin{bmatrix} \textit{smashed-state} & \\ \text{patient} & \boxed{2} \\ \text{undergoer} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Summary

- We propose a constraint-based implementation of argument linking in RRG using a metagrammar approach.
- Universal constraints on event frames and semantic roles.
- Constraints (MG classes) capturing the macrorole possibilities for each argument rank. The possible combinations of these classes are controlled by MG variables.
- The syntactic tree is decomposed into single fragments for each argument that carry constraints about the ranking of this argument, expressed via interface variables.
- The lexical predicates substitute into the argument structure templates, which triggers a unification of the lexical frame and the template frame.

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

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 2012/2013*, vol. 8036 Lecture Notes in Computer Science, 175–190. Berlin, Heidelberg: 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 *Proceedings of the 2013 Conference on Role and Reference Grammar*, .

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