Outline of a formal framework for Role and Reference Grammar

Rainer Osswald & Laura Kallmeyer

Heinrich-Heine-Universität Düsseldorf
SFB 991

Role and Reference Grammar 2013
International Conference

Freiburg, 2.8.2013
Why a formal framework for RRG?

► Is this relevant for typological analysis?

MAYBE NOT, BUT ...
a formalization can help to eliminate inconsistencies and gaps of a theory.

► Doesn’t RRG already come with a lot of formal elements?

SURE, BUT ...
these elements are not defined with logical and mathematical rigor.

► Any further advantages?

YES!
A formalization can serve as a basis for a computational treatment of RRG.

► Is that all?

NOT AT ALL!
E.g., a formalization should make it easier to extend and modify the theory.
The architecture of RRG

Syntactic inventory → Syntactic representation

Linking algorithm

Semantic representation

Constructional schemas

Lexicon

[do'(x, ∅)] CAUSE [INGR shattered'(y)]

⟨IF INT ⟨IF PRES ⟨ASP PERF PROG ⟨do'(Kim, [cry'(Kim)])⟩⟩⟩⟩

MORPHOLOGY —
SYNTAX Juncture: nuclear
Nexus: cosubordination
Construction:

Linking: default

SEMANTICS [SEM_{NUCL1}] CAUSE [SEM_{NUCL2}]

PRAGMATICS unspecified
Introduction

General plan of the formalization

- Take all explanatory components of RRG into account.
- Develop a declarative (i.e., non-procedural) constraint-based formulation.

Selection of tasks involved

- 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

Syntactic inventory

CLAUSE
  PrCS
  CORE
    PERIPHERY
  NUC NP PP
  PRED
  PP V

CLAUSE
  LDP
  SENTENCE
    CLAUSE
      PrCS
      CORE
        PERIPHERY
      NUC NP PP
      PRED
      PP V

CLAUSE
  LDP
  SENTENCE
    CLAUSE
      PrCS
      CORE
        PERIPHERY
      NUC NP PP
      PRED
      V
      PP

Issues
  ▶ How are syntactic templates defined?
  ▶ How do they combine?

Proposal
  ▶ Use concepts from Tree Adjoining Grammars (TAG)
  ▶ Adapt TAG formalism to the syntactic dimension of RRG

(e.g. Yesterday, what did Robin show to Pat in the library?)

[Van Valin 2005, p. 15]
Syntactic representation

Background  Lexicalized Tree Adjoining Grammars (LTAG)  
[e.g., Joshi & Schabes 1997]

▶ Tree rewriting system based on a set of **elementary** (initial and **auxiliary**) trees

▶ Two operations: **substitution** of initial trees at leaves  
**adjunction** of auxiliary trees

Example

two substitutions + one adjunction  
derived tree
Syntactic representation

**Background**  Lexicalized Tree Adjoining Grammars

- Elementary trees are *lexicalized*, i.e., have lexical anchors.

- “Complicate locally, simplify globally”  
  All predicate-argument dependencies are encoded in elementary trees.

- De-anchored elementary trees are organized in *tree families*, which capture variations in subcategorization frames.

**Example**  transitive verb family

```
S  NP  VP
  NP  V  NP
```

- Modular characterization of elementary trees in the *metagrammar*, a system of *tree descriptions*.  
  [Crabbé & Duchier 2005]
Syntactic representation

**Background**  Metagrammar for LTAGs

- Specification of elementary trees as **minimal models** of tree descriptions (tree classes)

**Example**  Metagrammar fragment for transitive verb class

Class *CanSubj*

```
S
   NP  VP
      V
```

Class *DirObj*

```
VP
   V  NP
```

Class *Subj*  

```
CanSubj ∨ ExtractedSubj
```

Class *ExtractedSubj*

```
S
   NP[WH=yes]
      S
         NP  VP
            ε  V
```

Class *ByObj*

```
VP[VOICE=passive]
   V  NP
      P  NP
         by
```

Class *ActV*

```
VP[VOICE=active]
   V
```

Class *PassV*

```
VP[VOICE=passive]
   V
```

Class *Transitive*

```
((Subj ∧ ActV) ∨ ByObj ∨ PassV) ∧ ((DirObj ∧ ActV) ∨ (Subj ∧ PassV))
```
Syntactic representation

Application to the syntactic inventory of RRG

1. What are the elementary trees of RRG?
2. How can they be combined?
3. 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)
e.g., the nuclear cosubordination templates of resultative constructions
Syntactic representation

Metagrammar sketches

```
core-spine  core-clause  precore-slot  prenuc-np  postnuc-np

CORE          CLAUSE          CLAUSE          CORE          CORE
|               |                | PrCS ≺ CORE | NP ≺ NUC   | NUC ≺ NP
NUC           CORE           |              |            |           |
PRED          |                |              |            |           |
V °            |                |              |            |           |

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

base-transitive :=
clause-spine ∧ prenuc-np ∧ postnuc-np
```
Tree operations for RRG

1. **Standard substitution**

   **Derived tree:**
   - CLAUSE
     - CORE
       - NUC
         - NP
           - PRED
             - V
             John claims
         - NP
           that
       - NUC
         - NP
           - PRED
             - V
             won
           the game
   
   **Derivation:**
   - CLAUSE
     - CORE
       - NUC
         - NP
           - PRED
             - V
   - CLAUSE
     - CORE
       - NUC
         - NP
           - PRED
             - V

2. **Sister adjunction**

   **Derived tree:**
   - CLAUSE
     - CORE
       - NUC
         - NP
           - ADV
             - V
             Mary deliberately
           - NP
             left
         - NP
           the party earlier
   
   **Derivation:**
   - CLAUSE
     - CORE
       - NUC
         - NP
           - PRED
             - V
             deliberately
           - NP
             left
         - NP
           earlier
Syntactic representation

Tree operations for RRG

3. Wrapping substitution

Derived tree:

```
PrCS        CLAUSE
  NP        CLAUSE
    NP      CORE
      NP     NUC
        PRED
          V
      NP
        PRED
          V
    NP
      NUC
    PRED
  NP
    NUC
    PRED
      V
```

Derivation:

```
PrCS
  CLAUSE
    NP
      CORE
        NP
          NUC
            PRED
              V
          NP
            NUC
              PRED
                V
      NP
        NUC
      PRED
    NP
      NUC
      PRED
        V
```

Tree Wrapping Grammar as a formal grammar framework

- More expressive than context-free grammars (can express cross-serial dependencies)
- CYK parsing algorithm with complexity $O(n^6)$

[Kallmeyer, Osswald & Van Valin 2013]
Logical structures in the lexicon and beyond

a. \( \text{do}'(x, \text{hit}'(x, y)) \)

b. \( \text{INGR shattered}'(y) \)

c. \([\text{do}'(x, \emptyset)] \text{CAUSE } [\text{INGR shattered}'(y)]\)

d. \([\text{do}'(x, \text{hit}'(x, y))] \text{CAUSE } [\text{INGR shattered}'(y)]\)

Logical analysis of logical structures

Basic (uncontroversial) assumptions

- RRG’s logical structures describe activities, states, changes of state, causations, etc.

- The decompositional structure of logical structures reflects the internal structure of the described events.

  E.g., causative events have a cause and an effect component.
Semantic representation

Logical analysis of logical structures (cont’d)

Example \[ \text{do}'(x, \emptyset) \] \ CAUSE \[ \text{INGR shattered}'(y) \]

\[ \exists e [\text{causation}(e) \land \exists e' \exists e'' [\text{CAUSE}(e, e') \land \text{EFFECT}(e, e'') \land \text{activity}(e') \land \exists x [\text{EFFECTOR}(e', x) \land \text{ingr-of-state}(e'') \land \exists s [\text{RESULT}(e'', s) \land \text{shattered-state}(s) \land \exists y [\text{PATIENT}(s, y)]]]]] \]

logical formula (first-order logic)

causal chain:
  \[ \text{cause} \to \text{effect} \to \text{effect of state} \to \text{shattered state} \to \text{patient} \]

generic model / “decompositional frame”
Semantic representation

Decompositional frames \(\approx\) multi-base feature structures with sorts and relations [Kallmeyer & Osswald, submitted]

frame / feature structure

\[
\begin{array}{c}
\text{causation} \\
\text{CAUSE} \\
\text{activity} \\
\text{EFFECTOR} \\
\text{EFFECT} \\
\text{ingr-of-state} \\
\text{RESULT} \\
\text{shattered-state} \\
\text{PATIENT} \\
\text{RESULT PATIENT} = 2
\end{array}
\]

description in attribute-value logic

\[
\begin{align*}
0 & : \text{ causation } \land 0 \cdot \text{ CAUSE } : \text{ activity } \land 0 \cdot \text{ CAUSE EFFECTOR } = 1 \\
0 & : \text{ EFFECT } : \text{ ingr-of-state } \land 0 \cdot \text{ EFFECT RESULT } : \text{ shattered-state } \\
0 & : \text{ EFFECT RESULT PATIENT } = 2
\end{align*}
\]

description in predicate logic

\[
\exists e' \exists e'' \exists s (\text{ causation}(0) \land \text{ CAUSE}(0, e') \land \text{ EFFECT}(0, e'') \land \text{ activity}(e') \land \text{ EFFECTOR}(e', 1) \land \text{ ingr-of-state}(e'') \land \text{ RESULT}(e'', s) \land \text{ shattered-state}(s) \land \text{ PATIENT}(s, 2))
\]

attribute-value matrix notation

\[
\begin{array}{c}
causation \\
\text{CAUSE} \\
\text{activity} \\
\text{EFFECTOR} \\
\text{ingr-of-state} \\
\text{EFFECT} \\
\text{RESULT} \\
\text{shattered-state} \\
\text{PATIENT} \\
\text{RESULT} \\
\text{PATIENT} = 2
\end{array}
\]
Advantages of decompositional frames

Frame representations allow us to combine two key aspects of RRG’s template-based structures and genuine logical representations:

▶ Like decompositional templates they are **concept-centered** and have inherent structural properties.

  I.e., **structural positions** relevant to the **linking** between syntax and semantics are accessible by attribute paths.

▶ Like logical representations, frame descriptions have a **well-defined** model-theoretic **interpretation**, and they are easily **extensible** by additional subcomponents and constraints.

Moreover:

▶ Subcomponents of frames can be **unified** with other frames (as, e.g., triggered by syntactic substitution) through base label identification.
Adjectival resultative construction in English (*wipe clean*, *paint white*, ...)

**resultative-nuc-cosubord**

**serial-pred-core**

CAUSE EFFECTOR : \( \top \rightarrow \text{CAUSE EFFECTOR} \equiv \text{ACTOR} \\
\text{EFFECT RESULT PATIENT} : \( \top \rightarrow \text{EFFECT RESULT PATIENT} \equiv \text{UNDERGOER} \)
Outline of a formalization of RRG

- Identify the **elementary syntactic trees** and characterize them as combinations of **tree constraints** in the **metagrammar**.

- Describe the combination of elementary trees by a small set of general **tree operations**.

- Re-analyze the logical structures of RRG as (descriptions of) **decompositional frames**.

- Draw a distinction between **frame constraints** and associated **generic models** similar to what is proposed for the syntax.

- Combine tree operations in the syntactic dimension with **frame unification** in the semantic dimension.

- Characterize the syntax-semantics interface in the metagrammar; (try to) capture **linking constraints** in metagrammar classes.
References


