Syntax-Driven Semantic Frame Composition in Lexicalized Tree Adjoining Grammars

[Tutorial]

Laura Kallmeyer & Rainer Osswald
Heinrich-Heine-Universität Düsseldorf

13th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+13)
September 4–6, 2017
Umeå University, Sweden
Goal: an LTAG architecture of the syntax-semantics interface that

- is compositional: the meaning of a complex expression can be computed from the meaning of its subparts and its composition operation.
- pairs entire elementary trees with meaning components.

Three principal approaches:

1. LTAG semantics with synchronous TAG (STAG)  
   [Shieber 1994; Nesson/Shieber 2006, 2008]

2. Unification based LTAG semantics with predicate logic  
   [Kallmeyer/Joshi 2003; Gardent/Kallmeyer 2003; Kallmeyer/Romero 2008]

3. Unification based LTAG semantics with frames  
   [Kallmeyer/Osswald 2013; Kallmeyer/Osswald/Pogodalla 2016]

This tutorial introduces the third approach.
A simple example

(1) Adam ate an apple.
A simple example

(1) Adam ate an apple.

\[
S \rightarrow NP[I=x] VP[I=e] \\
NP[I=u] \rightarrow 'Adam' \\
V \rightarrow 'ate' \\
VP[I=e] \rightarrow eating \[e\rightarrow\{\text{ACTOR } x, \text{THEME } y\}\] \\
NP[I=u] \rightarrow 'Adam' \\
NP[I=y] \rightarrow 'an apple' \\

u \[\text{person} \rightarrow \text{NAME} \rightarrow 'Adam'\]

"person" ‘Adam’
A simple example

(1) Adam ate an apple.
A simple example

(1) Adam ate an apple.

\[
\begin{align*}
S & \quad \text{[eating]} \\
\text{NP} & \quad \text{[actor \ x]} \\
\text{NP} & \quad \text{[theme \ y]} \\
\text{VP} & \quad \text{[reme]} \\
\text{V} & \quad \text{[ate]} \\
\text{NP} & \quad \text{[u \ Adam]} \\
\text{u} & \quad \text{[person \ Adam]} \\
\end{align*}
\]
A simple example

(1) Adam ate an apple.

NP[\(I=x\)]

\[ NP[\(I=u\)] \quad \text{NAME} \quad \text{‘Adam’} \]

\[ \text{‘Adam’} \]

\[ \text{‘ate’} \quad \text{‘an apple’} \]

VP[\(I=e\)]

\[ \text{eating} \]

\[ \text{ACTOR} \quad \text{x} \]

\[ \text{THEME} \quad \text{y} \]

NP[\(I=y\)]

\[ \text{‘an apple’} \]

S

\[ NP[\(I=v\)] \]

\[ \text{‘an apple’} \]

\[ \text{v} \quad \text{[apple]} \]
Introduction

A simple example

(1) Adam ate an apple.

\[
\begin{align*}
S & \rightarrow NP[I=x] \ \ VP[I=e] \\
NP[I=x] & \rightarrow 'Adam' \\
VP[I=e] & \rightarrow 'ate' \ NP[I=y] \\
NP[I=y] & \rightarrow 'an apple' \\
\end{align*}
\]
General properties of the syntax-semantics interface

- Semantic composition ($\approx$ unification) is triggered by syntactic composition ($\approx$ substitution and adjunction).
- Semantic representations are linked to entire elementary trees ($\approx$ elementary constructions).
- Interface features relate nodes in the syntactic tree to components in the semantic representation.
- Elementary constructions can be decomposed/defined by means of “metagrammatical” constraints.
Introduction

General properties of the syntax-semantics interface

- Semantic composition ($\approx$ unification) is triggered by syntactic composition ($\approx$ substitution and adjunction).
- Semantic representations are linked to entire elementary trees ($\sim$ elementary constructions).
- Interface features relate nodes in the syntactic tree to components in the semantic representation.
- Elementary constructions can be decomposed/defined by means of "metagrammatical" constraints.

Main components of the approach

1. Lexicalized Tree Adjoining Grammars (LTAG)  
   [Joshi/Schabes 1997; Abeille/Rambow 2000; …]
2. Decompositional Frame Semantics  
   [Kallmeyer/Osswald 2013; Osswald/Van Valin 2014]
3. Metagrammatical specification and decomposition  
   [Crabbé/Duchier 2005; Crabbé et al. 2013, Lichte/Petitjean 2015]
Introduction

Overview of the tutorial

- Frame semantics
  - Relation to semantic decomposition approaches
  - Formalization of frames and frames descriptions
  - Subsumption and unification
- Recap of Lexicalized Tree Adjoining Grammars (LTAG)
  - Basic definitions; feature-structure based TAG
  - Key properties of LTAG: extended domain of locality, etc.
  - Tree families and lexical anchoring
- Specification of elementary trees in the metagrammar
- Combing LTAG and frame semantics
  - Elementary constructions
  - Specification of constructions in the metagrammar
  - Examples (caused motion, Dative alternation)
- XMG implementation of frame semantics
- Frame semantics and quantification
- Further topics
Frame semantics: introduction

- Frames à la Fillmore/FrameNet

![Diagram showing Frame semantics with categories Item, Manner, Agent, Pieces, and example sentence: Slice the cake lengthwise into two halves. CNI.]}
Frame semantics: introduction

- Frames à la Fillmore/FrameNet

  ![Diagram](image)

  - **Slice** the cake lengthwise into two halves.
  - NP: the cake
  - AVP: lengthwise
  - PP[into]: two halves
  - CNI

- Frames (possibly embedded attribute-value structures with constraints) as a general format of conceptual representation

  [Barsalou 1992; Löbner 2014]
Frame semantics: introduction

- Frames à la Fillmore/FrameNet

![Diagram showing a frame for cutting a cake](image)

- Frames (possibly embedded attribute-value structures with constraints) as a general format of conceptual representation

  [Barsalou 1992; Löbner 2014]

- Decompositional frame semantics

  [Kallmeyer/Osswald 2013; Osswald/Van Valin 2014]
Semantic frames are commonly depicted as graphs with labeled nodes and edges, where nodes correspond to entities (individuals, events, ... ) and edges to functional (or non-functional) relations between these entities.
Semantic frames are commonly depicted as **graphs** with labeled nodes and edges, where **nodes** correspond to entities (individuals, events, ...) and **edges** to functional (or non-functional) relations between these entities.

Frames in this sense can be formalized as generalized feature structures with types, relations and node labels.
Example Lexical decomposition templates [Rappaport Hovav/Levin 1998]

(2) \[[x \text{ ACT}] \text{ CAUSE } [\text{ BECOME } [y \text{ BROKEN}]]\]
Frame semantics: introduction

Example  Lexical decomposition templates

(2) \([[[x \text{ ACT}] \text{ CAUSE } [\text{ BECOME } [y \text{ BROKEN}]]]]\)

Description in attribute-value logic

\[
\text{causation} \wedge \text{CAUSE} : \text{activity} \wedge \text{CAUSE} \text{ ACTOR } \triangleq x \\
\wedge \text{EFFECT} (\text{change-of-state} \wedge \text{FINAL} : (\text{broken-stage} \wedge \text{PATIENT } \triangleq y)) \\
\wedge \text{CAUSE} < \text{EFFECT}
\]
Frame semantics: introduction

Example  Lexical decomposition templates  

(2) \[[[x \text{ ACT}] \text{ CAUSE} [\text{ BECOME} [y \text{ BROKEN}]]]\]

Description in attribute-value logic

\[\text{causation} \land \text{CAUSE: activity} \land \text{CAUSE ACTOR} \triangleq x \land \text{EFFECT (change-of-state} \land \text{FINAL: (broken-stage} \land \text{PATIENT} \triangleq y)) \land \text{CAUSE < EFFECT}\]

Translation into first-order logic

\[\lambda e \exists e' \exists e'' \exists s (\text{causation}(e) \land \text{CAUSE}(e, e') \land \text{EFFECT}(e, e'') \land e' < e'' \land \text{activity}(e') \land \text{ACTOR}(e', x) \land \text{change-of-state}(e'') \land \text{FINAL}(e'', s) \land \text{broken-stage}(s) \land \text{PATIENT}(s, y))\]
Frame semantics: introduction

Basic assumptions

- **Attributes** (features, functional roles/relations) play a central role in the organization of semantic and conceptual knowledge and representation. [Barsalou 1992; Löbner 2014]

- Semantic components (participants, subevents) can be (recursively) addressed by attributes.
  - inherently **structured representations** (models);
    composition by **unification** (under constraints)

- Semantic processing may be seen as the **incremental construction** of **minimal (frame) models** based on the input, the context, and background knowledge (lexicon, ...).
Frame semantics: introduction

Example

(3) Anna ran to the station.

Attribute-value logic

\[ e \cdot (\text{running} \land \text{bounded-motion} \land \text{ACTOR} \not= \text{FINAL} \land \text{THEME} \land \text{LOC} : \text{station}) \]

Translation into first-order logic

\[ \exists x \exists s \exists y (\text{running}(e) \land \text{bounded-motion}(e) \land \text{ACTOR}(e, x) \land \text{FINAL}(e, s) \land \text{loc-stage}(s) \land \text{THEME}(s, x) \land \text{LOC}(s, y) \land \text{station}(y)) \]

Constraints

\[ \text{running} \Rightarrow \text{activity} \quad \text{(short for } \forall e(\text{running}(e) \rightarrow \text{activity}(e))) \]
\[ \text{loc-stage} \Rightarrow \text{THEME} : T \land \text{LOC} : T, \quad \ldots \]
Frame semantics: formalization

Vocabulary / Signature

Attr attributes (= dyadic functional relation symbols)
Rel (proper) relation symbols
Type type symbols (= monadic predicates)
Nname node names ("nominals") \}
Nvar node variables \} Nlabel

Primitive attribute-value descriptions (pAVDesc)

\[ t \mid p : t \mid p \equiv q \mid [p_1, \ldots, p_n] : r \mid p \triangleq k \]

(t \in \text{Type}, \ r \in \text{Rel}, \ p, q, p_i \in \text{Attr}^*, \ k \in \text{Nlabel})

Semantics

\[
\begin{align*}
P : t & \quad \begin{array} {c}
P \rightarrow t \end{array} \quad \begin{bmatrix} P \ t \end{bmatrix} \\
P \equiv Q & \quad \begin{array} {c}
P \rightarrow Q \end{array} \quad \begin{bmatrix} P \ Q \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
[P, Q] : r & \quad \begin{array} {c}
P \rightarrow Q \\
\quad \rightarrow \end{array} \quad \begin{bmatrix} P & Q \end{bmatrix} \quad r(1, 2) \\
P \triangleq k & \quad \begin{array} {c}
P \rightarrow k \end{array} \quad \begin{bmatrix} P & k \end{bmatrix}
\end{align*}
\]
Frame semantics: formalization

Primitive attribute-value formulas (pAVForm)

\[ k \cdot p : t \mid k \cdot p \triangleq l \cdot q \mid \langle k_1 \cdot p_1, \ldots, k_n \cdot p_n \rangle : r \]

\( t \in \text{Type}, \ r \in \text{Rel}, \ p, q, p_i \in \text{Attr}^*, \ k, l, k_i \in \text{Nlabel} \)

Semantics

\[
\begin{align*}
    k \cdot P : t & \quad \xrightarrow{P} t \quad k [P \ t] \quad \langle k \cdot P, l \cdot Q \rangle : r \quad k [P \ 1] \\
    k \cdot P \triangleq l \cdot Q & \quad \xrightarrow{P} \quad k [P \ 1] \quad l [Q \ 1] \quad r (1, 2)
\end{align*}
\]
Frame semantics: formalization

Primitive attribute-value formulas (pAVForm)

\[
k \cdot p : t \mid k \cdot p \triangleq l \cdot q \mid \langle k_1 \cdot p_1, \ldots, k_n \cdot p_n \rangle : r
\]

\[(t \in \text{Type}, \ r \in \text{Rel}, \ p, q, p_i \in \text{Attr}^*, \ k, l, k_i \in \text{Nlabel})\]

Semantics

\[
k \cdot P : t \quad k[ P \ t ] \quad \langle k \cdot P, l \cdot Q \rangle : r \quad k[ P \ 1 ] \quad l[ Q \ 2 ] \quad r(1, 2)
\]

Formal definitions (fairly standard)

Set/universe of “nodes” \( V \)

Interpretation function \( \mathcal{I} : \text{Attr} \rightarrow [V ightarrow V], \ 	ext{Type} \rightarrow \wp( V ), \ 	ext{Rel} \rightarrow \bigcup_n \wp( V^n ), \ 	ext{Nname} \rightarrow V \)

(Partial) variable assignment \( g : \text{Nvar} \rightarrow V \)
Frame semantics: formalization

Formal definitions (cont’d)

Abbreviation: \( I_g(k) = v \) for \( k \in \text{Nlabel} \) iff \( I(k) = v \) if \( k \in \text{Nname} \) and \( g(k) = v \) if \( k \in \text{Nvar} \) (\( g(k) \) defined)

Satisfaction of primitive descriptions

\[
\langle V, I, g \rangle, v \models t \quad \text{iff} \quad v \in I(t)
\]

\[
\langle V, I, g \rangle, v \models p : t \quad \text{iff} \quad I(p)(v) \in I(t)
\]

\[
\langle V, I, g \rangle, v \models p \doteq q \quad \text{iff} \quad I(p)(v) = I(q)(v)
\]

\[
\langle V, I, g \rangle, v \models [p_1, \ldots, p_n] : r \quad \text{iff} \quad \langle I(p_1)(v), \ldots, I(p_n)(v) \rangle \in I(r)
\]

\[
\langle V, I, g \rangle, v \models p \doteq k \quad \text{iff} \quad I(p)(v) = I_g(k) \quad (k \in \text{Nlabel})
\]

Satisfaction of primitive formulas

\[
\langle V, I, g \rangle \models k \cdot p : t \quad \text{iff} \quad I(p)(I_g(k)) \in I(t)
\]

\[
\langle V, I, g \rangle \models k \cdot p \doteq l \cdot q \quad \text{iff} \quad I(p)(I_g(k)) = I(q)(I_g(l))
\]

\[
\langle V, I, g \rangle \models \langle k_1 \cdot p_1, \ldots, k_n \cdot p_n \rangle : r \quad \text{iff} \quad \langle I(p_1)(I_g(k_1)), \ldots, I_g(p_n)(I(k_n)) \rangle \in I(r)
\]

Satisfaction of Boolean combinations as usual.
Frame semantics: formalization

Frame $F$ over $\langle \text{Attr}, \text{Type}, \text{Rel}, \text{Nname}, \text{Nvar} \rangle$:

$F = \langle V, \mathcal{I}, g \rangle$, with $V$ finite, such that every node $v \in V$ is reachable from some labeled node $w \in V$ via an attribute path, i.e.,

(i) $w = \mathcal{I}_g(k)$ for some $k \in \text{Nlabel} (= \text{Nname} \cup \text{Nvar})$ and
(ii) $v = \mathcal{I}(p)(w)$ for some $p \in \text{Attr}^*$.

Example

A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an attribute-value formula $\phi$ iff $F \models \phi$. 
Frame semantics: formalization

Subsumption

\( F_1 = \langle V_1, \mathcal{I}_1, g_1 \rangle \) subsumes \( F_2 = \langle V_2, \mathcal{I}_2, g_2 \rangle \) \((F_1 \sqsubseteq F_2)\) iff there is a (necessarily unique) morphism \( h : F_1 \to F_2 \), i.e., a function \( h : V_1 \to V_2 \) such that

(i) \( \mathcal{I}_2(f)(h(v)) = h(\mathcal{I}_1(f)(v)) \), if \( \mathcal{I}_1(f)(v) \) is defined, \( f \in \text{Attr}, v \in V_1 \),
(ii) \( h(\mathcal{I}_1(t)) \subseteq \mathcal{I}_2(t) \), for \( t \in \text{Type} \)
(iii) \( h(\mathcal{I}_1(r)) \subseteq \mathcal{I}_2(r) \), for \( r \in \text{Rel} \)
(iv) \( h(\mathcal{I}_1(n)) = \mathcal{I}_2(n) \), for \( n \in \text{Nname} \)
(v) \( h(g_1(x)) = g_2(x) \), for \( x \in \text{Nvar} \), if \( g_1(x) \) is defined

Example

```
    e
    \[\text{locomotion}\]  \[\text{ACTOR}\]  \[\text{man}\]
    \[\text{MANNER}\]  \[\text{MOVER}\]

\[\sqsubseteq\]

    x
    \[\text{activity}\]  \[\text{locomotion}\]  \[\text{ACTOR}\]  \[\text{man}\]
    \[\text{MANNER}\]  \[\text{MOVER}\]
    \[\text{PATH}\]  \[\text{path}\]
```
Frame semantics: formalization

Subsumption

\[ F_1 = \langle V_1, \mathcal{I}_1, g_1 \rangle \textbf{ subsumes } F_2 = \langle V_2, \mathcal{I}_2, g_2 \rangle (F_1 \subseteq F_2) \text{ iff there is a (necessarily unique) morphism } h : F_1 \rightarrow F_2, \text{ i.e., a function } h : V_1 \rightarrow V_2 \text{ such that}\]

(i) \( \mathcal{I}_2(f)(h(v)) = h(\mathcal{I}_1(f)(v)), \) if \( \mathcal{I}_1(f)(v) \) is defined, \( f \in \text{Attr}, v \in V_1, \)

(ii) \( h(\mathcal{I}_1(t)) \subseteq \mathcal{I}_2(t), \) for \( t \in \text{Type} \)

(iii) \( h(\mathcal{I}_1(r)) \subseteq \mathcal{I}_2(r), \) for \( r \in \text{Rel} \)

(iv) \( h(\mathcal{I}_1(n)) = \mathcal{I}_2(n), \) for \( n \in \text{Nname} \)

(v) \( h(g_1(x)) = g_2(x), \) for \( x \in \text{Nvar}, \) if \( g_1(x) \) is defined

Unification

Least upper bound \( F_1 \cup F_2 \) of \( F_1 \) and \( F_2 \) w.r.t. subsumption.
Frame semantics: formalization

Subsumption

\[ F_1 = \langle V_1, \mathcal{I}_1, g_1 \rangle \textbf{ subsumes } F_2 = \langle V_2, \mathcal{I}_2, g_2 \rangle \ (F_1 \sqsubseteq F_2) \text{ iff there is a (necessarily unique) morphism } h : F_1 \rightarrow F_2, \text{ i.e., a function } h : V_1 \rightarrow V_2 \text{ such that} \]

(i) \( \mathcal{I}_2(f)(h(v)) = h(\mathcal{I}_1(f)(v)) \), if \( \mathcal{I}_1(f)(v) \) is defined, \( f \in \text{Attr}, v \in V_1 \),

(ii) \( h(\mathcal{I}_1(t)) \subseteq \mathcal{I}_2(t) \), for \( t \in \text{Type} \)

(iii) \( h(\mathcal{I}_1(r)) \subseteq \mathcal{I}_2(r) \), for \( r \in \text{Rel} \)

(iv) \( h(\mathcal{I}_1(n)) = \mathcal{I}_2(n) \), for \( n \in \text{Nname} \)

(v) \( h(g_1(x)) = g_2(x) \), for \( x \in \text{Nvar} \), if \( g_1(x) \) is defined

Unification

Least upper bound \( F_1 \sqcup F_2 \) of \( F_1 \) and \( F_2 \) w.r.t. subsumption.

Theorem (Frame unification) \cite{Hegner 1994}

The worst case time-complexity of frame unification is almost linear in the number of nodes.
Frames as minimal models of attribute-value formulas

(i) Every frame is the minimal model (w.r.t. subsumption) of a finite conjunction of primitive attribute-value formulas.

(ii) Every finite conjunction of primitive attribute-value formulas has a minimal frame model.

Example

\[
e \cdot (\text{locomotion} \land \text{MANNER: walking} \land \text{ACTOR} \triangleq x \\
\land \text{MOVER} \triangleq \text{ACTOR} \land \text{PATH:} (\text{path} \land \text{ENDP: region})) \\
\land \langle e \cdot \text{PATH ENDP}, z \cdot \text{IN-REGION} \rangle : \text{part-of} \land x \cdot \text{man}
\]
Frame semantics: formalization

Attribute-value constraints

General format: \( \forall \phi, \phi \in \text{AVDesc} \ (\langle V, \mathcal{I}, g \rangle \models \forall \phi \text{ iff } \langle V, \mathcal{I}, g \rangle, v \models \phi \) for every \( v \in V \)

Notation: \( \phi \Rightarrow \psi \) for \( \forall (\phi \rightarrow \psi) \)

Horn constraints: \( \phi_1 \land \ldots \land \phi_n \Rightarrow \psi \ (\phi_i \in \text{pAVDesc} \cup \{T\}, \psi \in \text{pAVDesc} \cup \{\bot\}) \)

Examples

\textit{activity} \Rightarrow \textit{event}
\textit{causation} \land \textit{activity} \Rightarrow \bot
\text{AGENT} : T \Rightarrow \text{AGENT} \neq \text{ACTOR}
\textit{activity} \Rightarrow \text{ACTOR} : T
\textit{activity} \land \textit{motion} \Rightarrow \text{ACTOR} \neq \text{MOVER}
\ldots
Frame semantics: formalization

Attribute-value constraints

General format: $\forall \phi, \phi \in AVDesc \iff \forall \phi \text{ iff } \langle V, \mathcal{I}, g \rangle$, $\forall \phi \text{ for every } v \in V$

Notation: $\phi \Rightarrow \psi$ for $\forall (\phi \rightarrow \psi)$

Horn constraints: $\phi_1 \land \ldots \land \phi_n \Rightarrow \psi$ ($\phi_i \in pAVDesc \cup \{\top\}$, $\psi \in pAVDesc \cup \{\bot\}$)

Examples

- $\text{activity} \Rightarrow \text{event}$
- $\text{causation} \land \text{activity} \Rightarrow \bot$
- $\text{AGENT} : \top \Rightarrow \text{AGENT} \neq \text{ACTOR}$
- $\text{activity} \Rightarrow \text{ACTOR} : \top$
- $\text{activity} \land \text{motion} \Rightarrow \text{ACTOR} \neq \text{MOVER}$
- $\ldots$
Frame semantics: formalization

Attribute-value constraints

General format: \( \forall \phi, \phi \in \text{AVDesc} \ (\langle V, I, g \rangle \models \forall \phi \iff \langle V, I, g \rangle, v \models \phi \text{ for every } v \in V) \)

Notation: \( \phi \Rightarrow \psi \) for \( \forall (\phi \rightarrow \psi) \)

Horn constraints: \( \phi_1 \land \ldots \land \phi_n \Rightarrow \psi \) \((\phi_i \in \text{pAVDesc} \cup \{\top\}, \psi \in \text{pAVDesc} \cup \{\bot\})\)

Examples

- \( \text{activity} \Rightarrow \text{event} \)
- \( \text{causation} \land \text{activity} \Rightarrow \bot \)
- \( \text{AGENT} : \top \Rightarrow \text{AGENT} \neq \text{ACTOR} \)
- \( \text{activity} \Rightarrow \text{ACTOR} : \top \)
- \( \text{activity} \land \text{motion} \Rightarrow \text{ACTOR} \neq \text{MOVER} \)
- \( \ldots \)

Theorem (Frame unification under Horn constraints) \[\approx \text{Hegner 1994}\]

The worst case time-complexity of frame unification under a finite set of labeled Horn constraints is almost linear in the number of nodes.

(Labeled Horn constraint: \( k_1 \cdot \phi_1 \land \ldots \land k_n \cdot \phi_n \rightarrow l \cdot \psi \))
Frame semantics: formalization

Further examples

\[ \text{book} \Rightarrow \text{info-carrier} \]}
Frame semantics: formalization

Further examples

\[ \text{book} \Rightarrow \text{info-carrier} \quad \text{book} \quad \rightsquigarrow \quad \text{book, info-carrier} \]

[Babonnaud et al. 2016]
Frame semantics: formalization

Further examples

\[ \text{book} \Rightarrow \text{info-carrier} \]

\[ \text{book} \sim \text{book, info-carrier} \]

\[ \text{info-carrier} \Rightarrow \text{phys-obj} \wedge \text{CONTENT : information} \]
Frame semantics: formalization

Further examples

\[ book \Rightarrow info-carrier \]

\[ book \sim book, info-carrier \]

\[ info-carrier \Rightarrow phys-obj \land CONTENT: information \]

\[ info-carrier \sim info-carrier, phys-obj \]

\[ information \]

\[ CONTENT \]
Frame semantics: formalization

Further examples

\[ \text{book} \Rightarrow \text{info-carrier} \]

\[ \text{book} \sim \text{book, info-carrier} \]

\[ \text{info-carrier} \Rightarrow \text{phys-obj} \land \text{CONTENT} : \text{information} \]

\[ \text{info-carrier} \sim \text{info-carrier, phys-obj} \]

\[ \text{information} \rightarrow \text{CONTENT} \]

\[ \text{reading} \Rightarrow \text{PERC-COMP} : \text{perception} \land \text{MENT-COMP} : \text{comprehension} \]

\[ \land \left[ \text{PERC-COMP, MENT-COMP} : \text{ordered-overlap} \right] \]

[Babonnaud et al. 2016]
Frame semantics: formalization

Further examples

\[ book \Rightarrow info-carrier \]
\[ info-carrier \Rightarrow phys-obj \land \text{CONTENT} : \text{information} \]
\[ reading \Rightarrow \text{PERC-COMP} : \text{perception} \land \text{MENT-COMP} : \text{comprehension} \land \left[ \text{PERC-COMP}, \text{MENT-COMP} \right] : \text{ordered-overlap} \]
Lexicalized Tree Adjoining Grammars (LTAG): recap

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).

\[
\begin{align*}
S &\quad NP \quad VP \\
&\quad NP \quad V \quad NP \\
&\quad Adv \quad VP^* \\
&\quad ‘always’ \quad ‘ate’ \quad ‘an apple’
\end{align*}
\]

\[
\begin{align*}
S &\quad NP \quad VP \\
&\quad VP \quad Adv \quad NP \\
&\quad ‘always’ \quad ‘ate’ \quad ‘an apple’
\end{align*}
\]
Lexicalized Tree Adjoining Grammars (LTAG): recap

Feature-structure based TAG (FTAG)  
[Vijay-Shanker/Joshi 1988]

Each node has a top and a bottom feature structure:

- The top feature structure provide information about what the node presents within the surrounding structure.
- The bottom feature structure provide information about what the tree below the node represents.

In the final derived tree, top and bottom must be unified.

Operations on feature structures under substitution:

- The top of the root of the new initial tree unifies with the top of the substitution node.

Operations on feature structures under adjunction:

- The top of the root of the new auxiliary tree unifies with the top of the adjunction site; the bottom of the foot of the new tree unifies with the bottom of the adjunction site.
Lexicalized Tree Adjoining Grammars (LTAG): recap

Example

S
\[\begin{array}{c}
\text{NP}^{[\text{AGR}= 1]} \\
\text{VP}^{[\text{AGR}= 1, \text{MODE}= \text{ind}]} \\
\text{V}^{[\text{MODE}= \text{ind}]} \\
\text{VP}^{[\text{AGR}= 2, \text{PERS}= 3, \text{NUM}= \text{sg}]} \\
\text{VP}^{[\text{MODE}= \text{ger}]} \\
\end{array}\]
\[\begin{array}{c}
\text{NP}^{[\text{AGR}= \text{pers} = 3, \text{NUM}= \text{sg}]} \\
\text{‘John’} \\
\end{array}\]
Lexicalized Tree Adjoining Grammars (LTAG): recap

Example

\[
\begin{align*}
S & \rightarrow NP \rightarrow VP \\
NP & \rightarrow \text{John} \\
VP & \rightarrow \text{is} \rightarrow \text{VP} \\
VP & \rightarrow \text{is} \rightarrow \text{VP} \\
VP & \rightarrow \text{singing} \rightarrow \text{VP} \\
\end{align*}
\]

Result of derivation:
Lexicalized Tree Adjoining Grammars (LTAG): recap

Example

S

NP[AGR=1]

VP[AGR=1, MODE=ind]

V

‘singing’

NP[AGR=[PERS=3, NUM=sg]]

‘John’

VP[AGR=[PERS=3, NUM=sg], MODE=ger]

‘is’

VP*[MODE=ger]

After top-bottom unifications:

S

NP[AGR=1]

VP[AGR=[PERS=3, NUM=sg], MODE=ind]

V

‘singing’

‘John’

VP[MODE=ger]
Lexicalized Tree Adjoining Grammars (LTAG): recap

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.
Lexicalized Tree Adjoining Grammars (LTAG): recap

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]
Lexicalized Tree Adjoining Grammars (LTAG): recap

“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]
Tree families

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

Example  unanchored tree family for transitive verbs
Lexicalized Tree Adjoining Grammars (LTAG): recap

Tree families

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

Example  unanchored tree family for transitive verbs

Options for the specification/generation of tree families:

- Transformation rules applied to base trees (e.g., metarules in XTAG)
- Classes of tree constraints (“metagrammar”, XMG system)
XMG (eXtensible MetaGrammar) [e.g., Crabbé et al. 2013]

- A framework for specifying (the elementary structures of) tree based grammars by means of a **declarative** language (e.g., by dominance and precedence constraints)

- The specifications are organized into **classes** that can be **reused** (“imported”) by other classes.

- Classes may contain descriptions from different **dimensions**, and the XMG system can be extended in this respect, e.g., by a dimension of **frame** descriptions.

- An XMG **compiler** generates the elementary structures of a grammar from a metagrammar.
LTAG & metagrammar specification

Example

Class *CanSubj*

\[
S \\
NP \prec VP \\
V\diamond
\]

Class *ExtrSubj*

\[
S \\
NP[WH=yes] \prec^* S \\
NP \prec VP \\
\varepsilon \quad V\diamond
\]

Class *Subj*

\[
CanSubj \lor ExtSubj
\]

Class *ActV*

\[
\forall p[VOICE=active] \\
V\diamond
\]

Class *ByObj*

\[
\forall p[VOICE=passive] \\
V\diamond \prec^* PP \\
P \prec NP \\
by
\]

Class *PassV*

\[
\forall p[VOICE=passive] \\
V\diamond
\]

Class *Transitive*

\[
((Subj \land ActV) \lor ByObj \lor PassV) \land ((DirObj \land ActV) \lor (Subj \land PassV))
\]
Next step:

Add (frame) semantics to all components and link syntax to semantics.
LTAG & frame semantics

Overall architecture (syntax + semantics)

- metagrammar classes + AV constraints
  - compilation
  - unanchored families of constructions
  - lexical selection
  - lexical entries (+ frame semantics)
  - LTAG + frames
LTAG & frame semantics

Elements of the syntax-semantics interface

- **Elementary construction:**
  - elementary tree
  - + semantic frame
  - + linking of frame node variables to index features in the tree

- Specification in the metagrammar:
  - classes of tree constraints
  - + sets of attribute-value constraints
  - + linking of variables to index features

Note: Regularities about **argument linking** are expressed in the metagrammar. [Kallmeyer/Lichte/Osswald/Petitjean 2016]

- **Semantic composition** $\approx$ frame unification via identification of index variables during substitution and adjunction.
LTAG & frame semantics

Example (intransitive) directed motion construction

(4) John walked into the house.
**Lexical anchoring**

* morph entry + lemma entry + Constraints:

  - `morph entry`:
    - ‘walks’
    - pos: V

  - `lemma entry`:
    - walk:
      - FAM: n0V, ...
    - Syn1:
      - \[
          \begin{bmatrix}
            \text{AGR} = & \begin{bmatrix}
              \text{PERS} = 3 \\
              \text{NUM} = \text{sg}
            \end{bmatrix}
          \end{bmatrix}
        \]
      - lemma: walk
    - Syn2:
      - \[
          \begin{bmatrix}
            \text{E} = e_0
          \end{bmatrix}
        \]
    - Sem:
      - \[
          e_0 \begin{bmatrix}
            \text{locomotion} \\
            \text{MANNER walking}
          \end{bmatrix}
        \]

  - Constraints:
    - `locomotion ⇒ activity ∧ translocation`
    - `translocation ⇒ motion ∧ PATH : path`
    - `activity ⇒ ACTOR : T`
    - `motion ⇒ MOVER : T`
    - `activity ∧ motion ⇒ ACTOR ≠ MOVER`

**Diagram**

- S \rightarrow V_{[AGR = \ldots, E = e_0]}
  - ‘walks’
  - e_0 \rightarrow e
  - e \rightarrow e_{[activity]}
  - V_{[E = e]}

- S \rightarrow NP_{[i = x]} \rightarrow VP_{[E = e]}
  - ‘walks’
  - V_{[AGR = \ldots, E = e]}
  - \(e_0 \rightarrow e\)}
Example (transitive/causative) directed motion construction

(5) Mary threw/kicked/rolled the ball into the room.

Unanchored construction ($n0Vn1pp(dir)$):

$$
\text{S} \\
\text{NP}[I=x] \\
\text{VP}[E=e] \\
\text{VP}[E=e, \text{PATH}=p] \\
\text{VP}[E=e, \text{PATH}=p] \\
\text{V} \diamond [E=e] \\
\text{NP}[I=y] \\
\text{PP}[I=z, E=e']
$$

$$
\begin{array}{c}
\text{causation} \\
\text{CAUSE} \\
\text{ACTOR } x \\
\text{THEME } y \\
\text{EFFECT} \\
\text{MOVER } y \\
\text{GOAL } z \\
\text{PATH } p
\end{array}
$$
**Example**  (transitive/causative) directed motion construction

(5) Mary threw/kicked/rolled the ball into the room.

Unanchored construction \((n0Vn1pp(dir)):\)

![Diagram of unanchored construction]

(Partial) lexical entry for ‘threw’:

![Partial lexical entry diagram]
Metagrammar classes (syntax + semantics)

Class \( n0Vn1 \)

- Class \( n0V \)
  - Class \( Subj \)
    - \( S \)
      - \( NP[AGR=□, I=x] \prec VP[AGR=□] \)
      - \( V[E=e] \)
      - \( e[event, ACTOR \ x] \)
  - Class \( VSpine \)
    - \( VP[AGR=□] \)
    - \( V[AGR=□] \)

Class \( DirObj \)

- \( VP \)
  - \( V[AGR=□] \)
  - \( NP[AGR=□]\)
  - \( e[event, THEME \ x] \lor \ldots \)
Metagrammar classes (syntax + semantics)

Class \textit{DirPrepObj} \\
export: e, x \\
\[ \text{VP}_{[\text{PATH}=p]} \] \\
\[ \text{VP}_{[\text{PATH}=p]} \prec \text{PP}_{[\text{i.sc}=x, \text{e.sc}=e]} \] \\
\[ \text{V} \bowtie e \]
\[ \begin{bmatrix} \text{bounded-translocation} \\
\text{GOAL } x \\
\text{PATH } p \end{bmatrix} \]

Class \textit{n0Vpp(dir)} \\
identities: \( C_1.e = C_2.e \)

Class \textit{n0Vn1pp(dir)} \\
identities: \( C_1.e = e, C_2.x = z, C_2.e = e' \)
The Dative alternation (sketch)

(6) a. John sent Mary the book. [double object construction]
b. John sent the book to Mary. [prepositional object construction]
The Dative alternation (sketch)

(6) a. John sent Mary the book. [double object construction]
    b. John sent the book to Mary. [prepositional object construction]

a) \[ \begin{array}{c}
S \\
\text{NP}[I=x] & \text{VP}[E=e] \\
\text{V}\diamond[E=e] & \text{NP}[I=z] & \text{NP}[I=y] \\
\end{array} \]

b) \[ \begin{array}{c}
S \\
\text{NP}[I=x] & \text{VP}[E=e] \\
\text{VP}[E=e] & \text{PP}[PREP=to, I=z, E=e'] \\
\text{V}\diamond[E=e] & \text{NP}[I=y] \\
\end{array} \]
Examples

<syn>-dimension of class \textit{Subj}

```
class Subj
...
<syn>{
    node ?S [cat=s];
    node ?SUBJ [cat=np,
                  top=[i=?1]];
    node ?VP [cat=vp,bot=[e=?0]];
    node ?V (mark=anchor)
             [cat=v,top=[e=?0]];
    ?SUBJ>>?VP
}
```
XMG implementation of frame semantics

Examples

[syn]-dimension + [frame]-dimension of class \( Subj \)

```plaintext
class Subj
...
<syn>{
    node ?S [cat=s];
    node ?SUBJ [cat=np,
                top=[i=?1]];
    node ?VP [cat=vp,bot=[e=?0]];
    node ?V (mark=anchor)
            [cat=v,top=[e=?0]];
    ?SUBJ>>?VP
}
<frame>{
    ?0[event,
        actor:?1]
}
```
XMG implementation of frame semantics

Examples

Specification of frames:

```
<frame>
?0[causation,
  actor:?1,
  theme:?2,
  cause:[activity,
    actor:?1,
    theme:?2],
  effect:?4[mover:?2,
    goal:?3]
}
```
XMG implementation of frame semantics

Examples

Specification of frames:

```plaintext
<frame>
?0[causation,
    actor:?1,
    theme:?2,
    cause:[activity,
        actor:?1,
        theme:?2],
    effect:?4[mover:?2,
        goal:?3]
}
```

Specification of attribute-value constraints:

```plaintext
frame_constraints = {
    activity -> event, activity -> [actor:+],
    motion -> event, motion -> [mover:+],
    causation -> event, causation -> [cause:+, effect:+],
    locomotion -> activity motion
}
Frame semantics: extensions

Obvious issue
   What about sentence level semantics, quantification, intensionality, and all these things?
Frame semantics: extensions

Obvious issue

What about sentence level semantics, quantification, intensionality, and all these things?

Possible approaches

1. Use an attribute-value language with quantifiers (or, e.g., Hybrid Logic), and build formulas instead of models.  
  [e.g., Kallmeyer/Osswald/Pogodalla 2016]

2. Keep frames as basic semantic representations and evaluate quantification over the domain of frames.  
  [≈ Muskens 2013]

3. Try to retain the idea of minimal model building and consider frame types as proper entities of the model/universe.
Frame semantics: extensions

Attribute-value formulas with quantifiers (qAVForm)

\[ \forall \phi, \exists \phi \ (\phi \in \text{AVDesc}) \quad \forall x \alpha, \exists x \alpha \ (\alpha \in \text{AVForm} \cup \text{qAVForm}) \]

\[ \langle V, \mathcal{I}, g \rangle \models \forall \phi \ \text{iff} \ \langle V, \mathcal{I}, g \rangle, v \models \phi \ \text{for every} \ v \in V \]
\[ \langle V, \mathcal{I}, g \rangle \models \exists \phi \ \text{iff} \ \langle V, \mathcal{I}, g \rangle, v \models \phi \ \text{for some} \ v \in V \]

For \( x \notin \text{dom}(g) \):

\[ \langle V, \mathcal{I}, g \rangle \models \forall x \alpha \ \text{iff} \ \langle V, \mathcal{I}, g' \rangle \models \alpha \ \text{for every assignment} \ g' \ \text{with} \]
\[ \text{dom}(g') = \text{dom}(g) \cup \{x\} \ \text{and} \]
\[ g(v) = g'(v) \ \text{for all} \ v \in \text{dom}(g) \]

\[ \langle V, \mathcal{I}, g \rangle \models \exists x \alpha \ \text{iff} \ \langle V, \mathcal{I}, g' \rangle \models \alpha \ \text{for some assignment} \ g' \ \text{with} \ldots \]

Note: \( \forall \phi \equiv \forall x(x \cdot \phi), \ \exists \phi \equiv \exists x(x \cdot \phi) \) (with \( x \) not occurring in \( \phi \))
Frame semantics: extensions

Example

(7) Every man walked into some house.

\[ \forall x (x \cdot man \rightarrow \exists z (z \cdot house \land \exists (locomotion \land \text{MANNER: walking} \\
\land \text{ACTOR} \triangleq x \land \text{MOVER} \triangleq \text{ACTOR} \\
\land \text{GOAL} \triangleq z \land \text{PATH} : (path \land \text{ENDP: region} \\
\land \text{[PATH ENDP, GOAL IN-REGION] : part-of} ))) \]
Frame semantics: extensions

Example

(7) Every man walked into some house.

\[ \forall x (x \cdot \text{man} \rightarrow \exists z (z \cdot \text{house} \land \exists (\text{locomotion} \land \text{MANNER: walking} \land \text{ACTOR} \triangleq x \land \text{MOVER} \triangleq \text{ACTOR} \land \text{GOAL} \triangleq z \land \text{PATH:} (\text{path} \land \text{ENDP: region}) \land [\text{PATH ENDP, GOAL IN-REGION} : \text{part-of}))) \]

Example model

[Diagram of a frame structure representing the sentence.(7): Nodes for 'John', 'Peter', 'man', 'locomotion', 'house', 'walking', 'path', 'ENDP', with connections indicating their relationships and properties such as actor, mover, goal, path, and part-of.]
Frame semantics: extensions

**AV logic with quantifiers + underspecification** ("hole semantics")

(8) Every dog barked.

\[
\forall x (x \cdot 5 \rightarrow 6), \quad 5 \triangleq^* 2, \quad 6 \triangleq^* 3
\]

NP\[i = x, \text{MINS } = 3\] → NP\[i = 4, \text{MINS } = l_1\] → NP\[i = l_2\] → S → VP → V

NP\[i = l_2\] → N → 'dog' → l_2 : dog

l_1 : \exists (\text{barking} \land \text{AGENT } \triangleq 4)
Frame semantics: extensions

AV logic with quantifiers + underspecification (“hole semantics”)

(8) Every dog barked.

\[ \forall x (x \cdot 5 \rightarrow 6), \quad [5 \triangleleft^* 2], \quad [6 \triangleleft^* 3] \]

\[ \sim \forall x (x \cdot 5 \rightarrow 6), \quad l_2 : dog, \quad l_1 : \exists (barking \land \text{AGENT} = x), \quad [5 \triangleleft^* l_2], \quad [6 \triangleleft^* l_1] \]
Frame semantics: extensions

**AV logic with quantifiers + underspecification** ("hole semantics")

(8) Every dog barked.

\[
\forall x (x \cdot 5 \rightarrow 6), \quad 5 \triangleleft^* 2, \quad 6 \triangleleft^* 3
\]

\[
\sim \quad \forall x (x \cdot 5 \rightarrow 6), \quad l_2 : \text{dog}, \quad l_1 : \exists (\text{barking} \land \text{AGENT} \triangleq x), \quad 5 \triangleleft^* l_2, \quad 6 \triangleleft^* l_1
\]

\[
\sim \quad \forall x (x \cdot \text{dog} \rightarrow \exists (\text{barking} \land \text{AGENT} \triangleq x))
\]
Frame semantics: extensions

Alternative: Hybrid Logic + underspecification (“hole semantics”)

(9) Every dog barked.

\[
\begin{align*}
S & \quad l_1 : \exists(barking \land \langle \text{AGENT} \rangle x), \quad [5] \bowtie^* l_2, [6] \bowtie^* l_1 \\
\quad \text{NP} \quad l_2 : \text{dog} & \quad V \\
\quad \text{NP} \quad NP_{[E = 2]} & \quad \text{VP} \\
\quad \text{NP} \quad NP_{[I = x, \text{MINS} = \square]} & \quad \text{Det} \\
\quad 'every' & \quad \text{N} \\
\quad \forall(\downarrow x. [5 \rightarrow [6]), \quad [5] \bowtie^* 2, [6] \bowtie^* 3 \\
\end{align*}
\]

[from Kallmeyer/Osswald/Pogodalla 2016]
Frame semantics: extensions

Alternative: Hybrid Logic + underspecification ("hole semantics")

(10) Peter knocked at the door for ten minutes.

\[ \exists (\downarrow e.\text{nonbounded} \land \langle \text{DURATION} \rangle \text{ten-minutes} \land \forall (\langle \text{segment-of} \rangle e \rightarrow \text{knocking} \land \langle \text{AGENT} \rangle i \land \langle \text{PATIENT} \rangle j)) \land \forall i (\langle \text{person} \rangle \land \langle \text{NAME} \rangle \text{Peter}) \land \forall j (\langle \text{door} \rangle) \]

[from Kallmeyer/Osswald/Pogodalla 2016]
Application: Formalization of Role and Reference Grammar

Role and Reference Grammar (RRG): [e.g., Van Valin 2005, 2010]

A non-transformational grammatical theory, inspired by typological concerns, which makes use of syntactic templates and lexical decomposition structures, among others.

Aspects of the formalization

Modified tree operations because of flat syntactic structures:
- Wrapping substitution and sister adjunction.
  [Kallmeyer/Osswald/Van Valin 2013; Osswald/Kallmeyer, to appear]
- Semantic frames instead of semantic templates.
- Argument linking rules as constraints in the metagrammar.
  [Kallmeyer/Lichte/Osswald/Petitjean 2016]
Application: Formalization of Role and Reference Grammar

Role and Reference Grammar (RRG): [e.g., Van Valin 2005, 2010]

A non-transformational grammatical theory, inspired by typological concerns, which makes use of syntactic templates and lexical decomposition structures, among others.

Aspects of the formalization

- Modified tree operations because of flat syntactic structures: Wrapping substitution and sister adjunction.
  [Kallmeyer/Osswald/Van Valin 2013; Osswald/Kallmeyer, to appear]

- Semantic frames instead of semantic templates.

- Argument linking rules as constraints in the metagrammar.
  [Kallmeyer/Lichte/Osswald/Petitjean 2016]
Thank you very much for your attention!
References I


