Frame-Semantic Composition at the Syntax-Semantics Interface

Rainer Osswald

Heinrich-Heine-Universität Düsseldorf

(joint work with Laura Kallmeyer)

Workshop on Logic and Algorithms in Computational Linguistics 2017
(LACompLing2017)

Stockholm, August 16–19, 2017

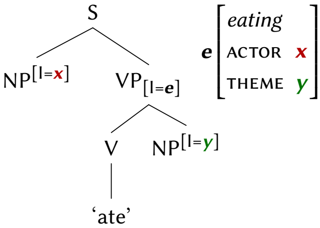
Department of Mathematics, Stockholm University

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

CRC 991

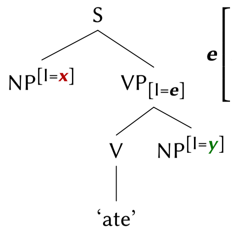
A simple example

(1) Adam ate an apple.



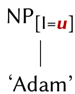
A simple example

(1) Adam ate an apple.



$$\boldsymbol{e}\begin{bmatrix} \textit{eating} & \\ \text{ACTOR} & \boldsymbol{x} \\ \text{THEME} & \boldsymbol{y} \end{bmatrix}$$

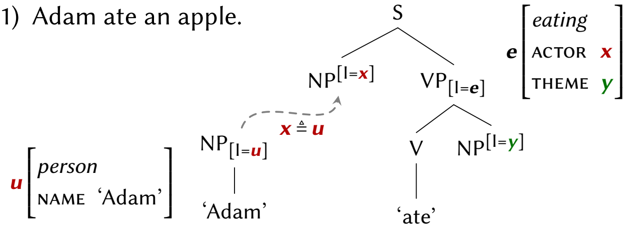
$$\boldsymbol{u}\begin{bmatrix} \textit{person} & \\ \text{NAME} & \text{'Adam'} \end{bmatrix}$$



Introduction

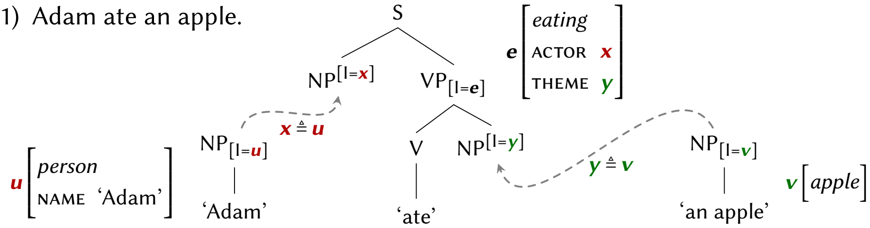
A simple example

(1) Adam ate an apple.



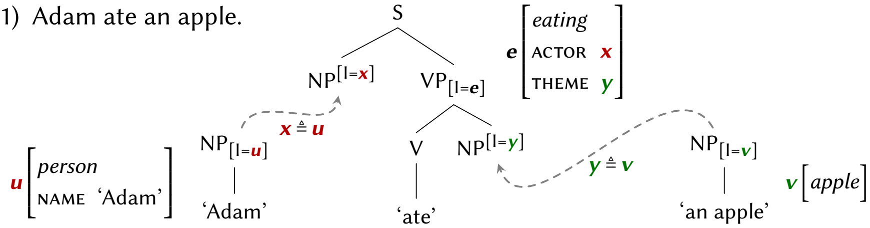
A simple example

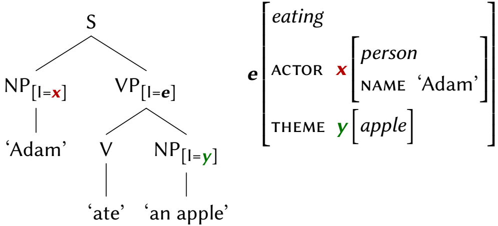
(1) Adam ate an apple.



A simple example

(1) Adam ate an apple.

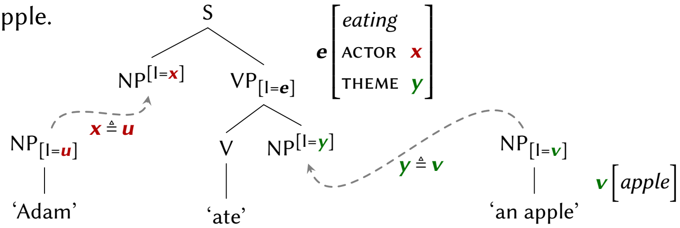


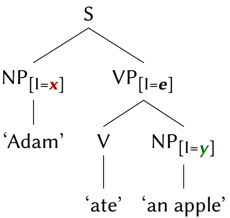


A simple example

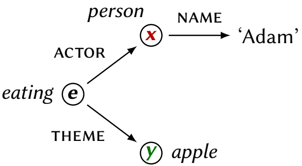
(1) Adam ate an apple.

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





$$e\begin{bmatrix}\textit{eating} & & \\ \textsc{actor} & x & \begin{bmatrix}\textit{person}\\ \textsc{name}\ \text{‘Adam’}\end{bmatrix}\\ \textsc{theme} & y & \begin{bmatrix}\textit{apple}\end{bmatrix}\end{bmatrix}$$



Introduction

A model of the syntax-semantics interface

- Semantic composition (≈ unification) is triggered by syntactic composition (≈ substitution and adjunction).
- Semantic representations are linked to entire elementary trees.
  (A further decomposition is possible in the “metagrammar”.)
- Interface features relate nodes in the syntactic tree to components in the semantic representation.

A model of the syntax-semantics interface

- Semantic composition (≈ unification) is triggered by syntactic composition (≈ substitution and adjunction).
- Semantic representations are linked to entire elementary trees. (A further decomposition is possible in the “metagrammar”.)
- Interface features relate nodes in the syntactic tree to components in the semantic representation.

Main components of the framework

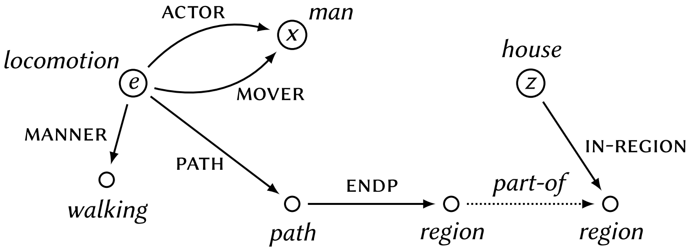
[Kallmeyer/Osswald 2013]

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]

Decompositional Frame Semantics

Frames as a way to represent rich lexical and constructional content.

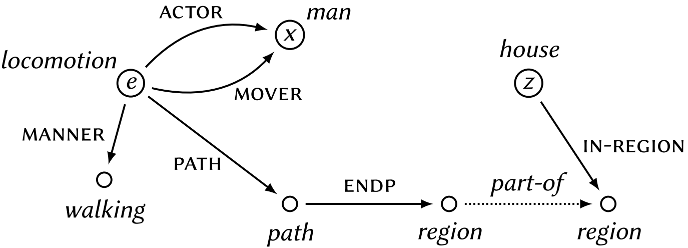
- Semantic frames are commonly depicted as **graphs** with labeled nodes and edges, where **nodes** correspond to entities (individuals, events, ...) and **edges** to functional (or non-functional) relations between these entities.



Decompositional Frame Semantics

Frames as a way to represent rich lexical and constructional content.

- 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.

Decompositional Frame Semantics

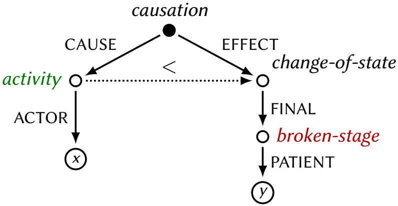
Example Lexical decomposition templates [Rappaport Hovav/Levin 1998]

(2) [[*x* ACT] CAUSE [BECOME [*y* *BROKEN*]]]

Decompositional Frame Semantics

Example Lexical decomposition templates [Rappaport Hovav/Levin 1998]

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



$$\begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & \begin{bmatrix}\textit{activity} & \\ \text{EFFECTOR} & x\end{bmatrix} \\ \text{EFFECT} & \begin{bmatrix}\textit{change-of-state} & \\ \text{FINAL} & \begin{bmatrix}\textit{broke-stage} & \\ \text{PATIENT} & y\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

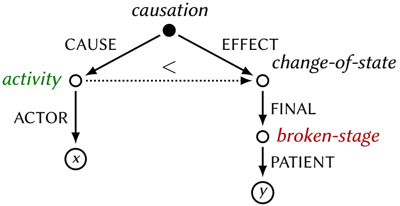
$\text{CAUSE} < \text{EFFECT}$

Description in attribute-value logic

$$\textit{causation} \wedge \text{CAUSE} : \textit{activity} \wedge \text{CAUSE ACTOR} \triangleq x \wedge \text{CAUSE PATIENT} \triangleq y$$
$$\wedge\ \text{EFFECT}\,(\textit{change-of-state} \wedge \text{FINAL} : \textit{broken-stage})$$
$$\wedge\ \text{CAUSE} < \text{EFFECT}$$

Example Lexical decomposition templates [Rappaport Hovav/Levin 1998]

(2) [[x ACT] CAUSE [BECOME [y *BROKEN*]]]



$$\begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & \begin{bmatrix}\textit{activity} & \\ \text{EFFECTOR} & x\end{bmatrix} \\ \text{EFFECT} & \begin{bmatrix}\textit{change-of-state} & \\ \text{FINAL} & \begin{bmatrix}\textit{broke-stage} & \\ \text{PATIENT} & y\end{bmatrix}\end{bmatrix}\end{bmatrix}$$
$\text{CAUSE} < \text{EFFECT}$

Description in attribute-value logic

$$\textit{causation} \wedge \text{CAUSE} : \textit{activity} \wedge \text{CAUSE ACTOR} \triangleq x \wedge \text{CAUSE PATIENT} \triangleq y$$
$$\wedge\ \text{EFFECT}\,(\textit{change-of-state} \wedge \text{FINAL} : \textit{broken-stage})$$
$$\wedge\ \text{CAUSE} < \text{EFFECT}$$

Translation into first-order logic

$$\lambda e \exists e' \exists e'' \exists s (\textit{causation}(e) \wedge \text{CAUSE}(e, e') \wedge \text{EFFECT}(e, e'') \wedge e' < e'' \wedge$$
$$\textit{activity}(e') \wedge \text{ACTOR}(e', x) \wedge \textit{change-of-state}(e'') \wedge$$
$$\text{FINAL}(e'', s) \wedge \textit{broken-stage}(s) \wedge \text{PATIENT}(s, y))$$

Decompositional Frame Semantics

Basic assumptions

- **Attributes** (features, functional roles/relations) play a central role in the organization of semantic and conceptual knowledge and representation. [Barsalou 1992; Petersen 2007; Löbner 2014]

- Semantic components (participants, subevents) can be (recursively) addressed by **attributes**.

  $\rightsquigarrow$ inherently **structured representations** (models);
  composition by **unification** (under constraints)

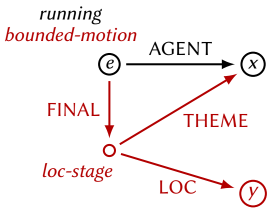
- Semantic processing may be seen as the **incremental construction** of **minimal** (**frame**) **models** based on the input, the context, and background knowledge (lexicon, …).

Decompositional Frame Semantics

Example

(3) Anna ran to the station.

$$e\begin{bmatrix} \textit{running} \wedge \textit{bounded-motion} & \\ \text{AGENT} & \boxed{1}\ x \\ \text{FINAL} & \begin{bmatrix} \textit{loc-stage} & \\ \text{THEME} & \boxed{1} \\ \text{LOC} & y \end{bmatrix} \end{bmatrix}$$



Attribute-value logic

$$e \cdot (\textit{running} \wedge \textit{bounded-motion} \wedge \text{ACTOR} \triangleq x \wedge$$
$$\text{FINAL} : \textit{loc-stage} \wedge \text{FINAL THEME} \doteq \text{ACTOR} \wedge \text{FINAL LOC} \triangleq y)$$

Translation into first-order logic

$$\textit{running}(e) \wedge \textit{bounded-motion}(e) \wedge \text{ACTOR}(e, x) \wedge$$
$$\exists s(\text{FINAL}(e, s) \wedge \textit{loc-stage}(s) \wedge \text{THEME}(s, x) \wedge \text{LOC}(s, y))$$

Constraints

$\textit{running} \Rightarrow \textit{activity}$ (short for $\forall e(\textit{running}(e) \rightarrow \textit{activity}(e))$),

$\textit{loc-stage} \Rightarrow \text{THEME} : \top \wedge \text{LOC} : \top, \quad \ldots$

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

Nname, Nvar } Nlabel

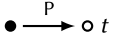
Primitive attribute-value descriptions (pAVDesc)

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

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

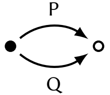
Semantics

P : t



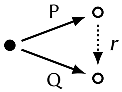
$\begin{bmatrix} \text{P} & t \end{bmatrix}$

P $\doteq$ Q



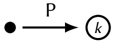
$\begin{bmatrix} \text{P} & \boxed{1} \\ \text{Q} & \boxed{1} \end{bmatrix}$

$[\text{P}, \text{Q}] : r$



$\begin{bmatrix} \text{P} & \boxed{1} \\ \text{Q} & \boxed{2} \end{bmatrix}$
$r\,(\boxed{1}, \boxed{2})$

P $\triangleq$ k



$\begin{bmatrix} \text{P} & k\,[\ ] \end{bmatrix}$

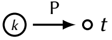
Primitive attribute-value formulas (pAVForm)

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

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

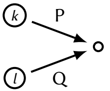
Semantics

$k \cdot \mathrm{P} : t$



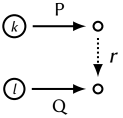
$$k\begin{bmatrix}\mathrm{P} & t\end{bmatrix}$$

$k \cdot \mathrm{P} \triangleq l \cdot \mathrm{Q}$



$$k\begin{bmatrix}\mathrm{P} & \boxed{1}\end{bmatrix} \quad l\begin{bmatrix}\mathrm{Q} & \boxed{1}\end{bmatrix}$$

$\langle k \cdot \mathrm{P}, l \cdot \mathrm{Q} \rangle : r$



$$k\begin{bmatrix}\mathrm{P} & \boxed{1}\end{bmatrix} \quad l\begin{bmatrix}\mathrm{Q} & \boxed{2}\end{bmatrix} \quad r(\boxed{1}, \boxed{2})$$

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

$k \cdot \mathrm{P} : t$

$$\begin{tikzcd} k \arrow[r, "\mathrm{P}"] & \circ\ t \end{tikzcd}$$

$$k\begin{bmatrix} \mathrm{P} & t \end{bmatrix}$$

$k \cdot \mathrm{P} \triangleq l \cdot \mathrm{Q}$

$$\begin{tikzcd} k \arrow[dr, "\mathrm{P}"] & \\ & \circ \\ l \arrow[ur, "\mathrm{Q}"'] & \end{tikzcd}$$

$$k\begin{bmatrix} \mathrm{P} & \boxed{1} \end{bmatrix} \quad l\begin{bmatrix} \mathrm{Q} & \boxed{1} \end{bmatrix}$$

$\langle k \cdot \mathrm{P}, l \cdot \mathrm{Q} \rangle : r$

$$\begin{tikzcd} k \arrow[r, "\mathrm{P}"] & \circ \arrow[d, dotted, "r"] \\ l \arrow[r, "\mathrm{Q}"'] & \circ \end{tikzcd}$$

$$k\begin{bmatrix} \mathrm{P} & \boxed{1} \end{bmatrix} \quad l\begin{bmatrix} \mathrm{Q} & \boxed{2} \end{bmatrix} \quad r(\boxed{1}, \boxed{2})$$

Formal definitions (fairly standard)

Set/universe of "nodes"	V
Interpretation function	$I :\ \text{Attr} \to [V \rightharpoonup V],\ \ \text{Type} \to \wp(V),$
	$\text{Rel} \to \bigcup_n \wp(V^n),\ \ \text{Nname} \to V$
(Partial) variable assignment	$g : \text{Nvar} \rightharpoonup V$

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 descriptions

$$
\begin{array}{ll}
\langle V, I, g\rangle, v \vDash t & \text{iff } v \in I(t) \\
\langle V, I, g\rangle, v \vDash p : t & \text{iff } I(p)(v) \vDash t \\
\langle V, I, g\rangle, v \vDash p \doteq q & \text{iff } I(p)(v) = I(q)(v) \\
\langle V, I, g\rangle, v \vDash [p_1, \ldots, p_n] : r & \text{iff } \langle I(p_1)(v), \ldots, I(p_n)(v)\rangle \in I(r) \\
\langle V, I, g\rangle, v \vDash p \triangleq k & \text{iff } I(p)(v) = I_g(k) \quad (k \in \text{Nlabel})
\end{array}
$$

Satisfaction of formulas

$$
\begin{array}{ll}
\langle V, I, g\rangle \vDash k \cdot p : t & \text{iff } I(p)(I_g(k)) \in I(t) \\
\langle V, I, g\rangle \vDash k \cdot p \triangleq l \cdot q & \text{iff } I(p)(I_g(k)) = I(q)(I_g(l)) \\
\langle V, I, g\rangle \vDash \langle k_1 \cdot p_1, \ldots, k_n \cdot p_n\rangle : r & \text{iff } \langle I(p_1)(I_g(k_1)), \ldots, I_g(p_n)(I(k_n))\rangle \in I(r)
\end{array}
$$

Satisfaction of Boolean combinations as usual.

Frame semantics: formalization

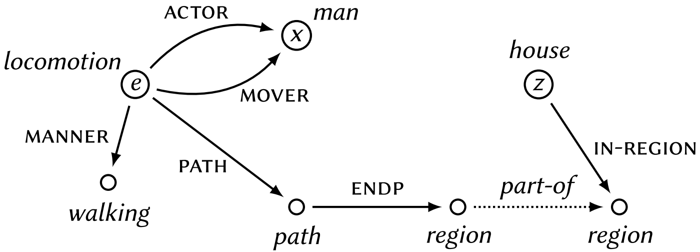
Frame F over $\langle \text{Attr}, \text{Type}, \text{Rel}, \text{Nname}, \text{Nvar} \rangle$:

$F = \langle V, I, g \rangle$, 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) $w = I_g(k)$ for some $k \in \text{Nlabel}$ and

(ii) $v = I(p)(w)$ for some $p \in \text{Attr}^*$.

Example



Frame semantics: formalization

Subsumption

$F_1 = \langle V_1, I_1, g_1 \rangle$ **subsumes** $F_2 = \langle V_2, I_2, g_2 \rangle$ ($F_1 \sqsubseteq F_2$) 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) $I_2(f)(h(v)) = h(I_1(f)(v))$, if $I_1(f)(v)$ is defined, $f \in \mathrm{Attr}$, $v \in V_1$,

(ii) $h(I_1(t)) \subseteq I_2(t)$, for $t \in \mathrm{Type}$

(iii) $h(I_1(r)) \subseteq I_2(r)$, for $r \in \mathrm{Rel}$

(iv) $h(I_1(n)) = I_2(n)$, for $n \in \mathrm{Nname}$

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

Frame semantics: formalization

Subsumption

$F_1 = \langle V_1, I_1, g_1 \rangle$ **subsumes** $F_2 = \langle V_2, I_2, g_2 \rangle$ ($F_1 \sqsubseteq F_2$) 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) $I_2(f)(h(v)) = h(I_1(f)(v))$, if $I_1(f)(v)$ is defined, $f \in \text{Attr}$, $v \in V_1$,

(ii) $h(I_1(t)) \subseteq I_2(t)$, for $t \in \text{Type}$

(iii) $h(I_1(r)) \subseteq I_2(r)$, for $r \in \text{Rel}$

(iv) $h(I_1(n)) = I_2(n)$, 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.

Frame semantics: formalization

Subsumption

$F_1 = \langle V_1, I_1, g_1 \rangle$ **subsumes** $F_2 = \langle V_2, I_2, g_2 \rangle$ ($F_1 \sqsubseteq F_2$) 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) $I_2(f)(h(v)) = h(I_1(f)(v))$, if $I_1(f)(v)$ is defined, $f \in \text{Attr}$, $v \in V_1$,

(ii) $h(I_1(t)) \subseteq I_2(t)$, for $t \in \text{Type}$

(iii) $h(I_1(r)) \subseteq I_2(r)$, for $r \in \text{Rel}$

(iv) $h(I_1(n)) = I_2(n)$, 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) [Hegner 1994]

The worst case time-complexity of frame unification is almost linear in the number of nodes.

Frame semantics: formalization

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.

Frame semantics: formalization

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.

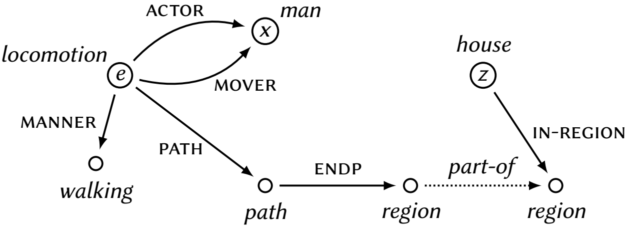
Frame semantics: formalization

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



$$
\begin{aligned}
&e \cdot (\mathit{locomotion} \wedge \textsc{manner}\!:\mathit{walking} \wedge \textsc{actor} \triangleq x \wedge \\
&\quad \textsc{mover} \doteq \textsc{actor} \wedge \textsc{path}\!:(\mathit{path} \wedge \textsc{endp}\!:\mathit{region})) \wedge \\
&\langle e \cdot \textsc{path endp},\ z \cdot \textsc{in-region} \rangle : \mathit{part\text{-}of} \wedge x \cdot \mathit{man}
\end{aligned}
$$

Frame semantics: formalization

Attribute-value constraints

General format: $\forall\phi,\ \phi \in \text{AVDesc}$

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

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

Examples

$\mathit{activity} \Rightarrow \mathit{event}$
$\mathit{causation} \wedge \mathit{activity} \Rightarrow \bot$
$\textsc{agent} : \top \Rightarrow \textsc{agent} \doteq \textsc{actor}$
$\mathit{activity} \Rightarrow \textsc{actor} : \top$
$\mathit{activity} \wedge \mathit{motion} \Rightarrow \textsc{actor} \doteq \textsc{mover}$

Frame semantics: formalization

Attribute-value constraints

General format: $\forall \phi,\ \phi \in \text{AVDesc}$

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

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

Examples

$\textit{activity} \Rightarrow \textit{event}$
$\textit{causation} \wedge \textit{activity} \Rightarrow \bot$
$\textsc{agent} : \top \Rightarrow \textsc{agent} \doteq \textsc{actor}$
$\textit{activity} \Rightarrow \textsc{actor} : \top$
$\textit{activity} \wedge \textit{motion} \Rightarrow \textsc{actor} \doteq \textsc{mover}$

Theorem (Frame unification under Horn constraints) [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: $k_1 \cdot \phi_1 \wedge \ldots \wedge k_n \cdot \phi_n \rightarrow l \cdot \psi$)

Frame semantics: formalization

Further examples [Babonnaud et al. 2016]

book ⇒ *info-carrier*

Frame semantics: formalization

Further examples [Babonnaud et al. 2016]

book ⇒ *info-carrier*

$$\overset{\textit{book}}{\bullet} \rightsquigarrow \overset{\textit{book, info-carrier}}{\bullet}$$

Frame semantics: formalization

Further examples [Babonnaud et al. 2016]

book $\Rightarrow$ *info-carrier*

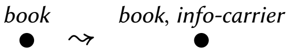
$\overset{book}{\bullet} \quad \rightsquigarrow \quad \overset{book,\ info\text{-}carrier}{\bullet}$

info-carrier $\Rightarrow$ *phys-obj* $\wedge$ CONTENT : *information*

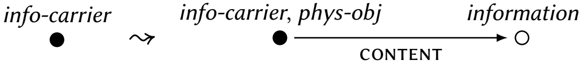
Frame semantics: formalization

Further examples [Babonnaud et al. 2016]

book ⇒ *info-carrier*



info-carrier ⇒ *phys-obj* ∧ CONTENT : *information*



Frame semantics: formalization

Further examples [Babonnaud et al. 2016]

$\textit{book} \Rrightarrow \textit{info-carrier}$

$$\overset{\textit{book}}{\bullet} \quad \rightsquigarrow \quad \overset{\textit{book, info-carrier}}{\bullet}$$

$\textit{info-carrier} \Rrightarrow \textit{phys-obj} \wedge \textsc{content} : \textit{information}$

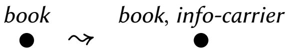
$$\overset{\textit{info-carrier}}{\bullet} \quad \rightsquigarrow \quad \overset{\textit{info-carrier, phys-obj}}{\bullet} \xrightarrow[\textsc{content}]{} \overset{\textit{information}}{\circ}$$

$\textit{reading} \Rrightarrow \textsc{perc-comp} : \textit{perception} \wedge \textsc{ment-comp} : \textit{comprehension}$
$\wedge \left[\textsc{perc-comp}, \textsc{ment-comp}\right] : \textit{ordered-overlap}$

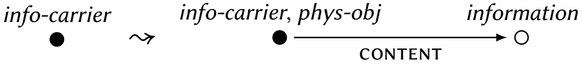
Frame semantics: formalization

Further examples [Babonnaud et al. 2016]

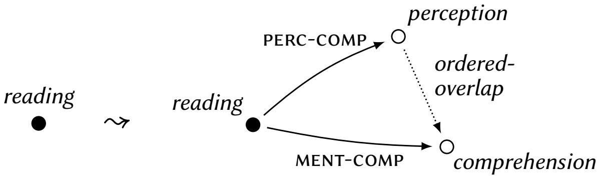
book ⇛ *info-carrier*



info-carrier ⇛ *phys-obj* ∧ CONTENT : *information*



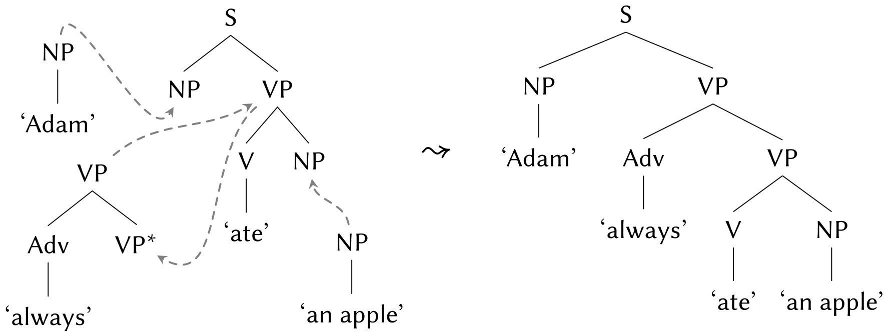
reading ⇛ PERC-COMP : *perception* ∧ MENT-COMP : *comprehension*
∧ [PERC-COMP, MENT-COMP] : *ordered-overlap*



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).



Lexicalized Tree Adjoining Grammars (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.

Lexicalized Tree Adjoining Grammars (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]

Lexicalized Tree Adjoining Grammars (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**"

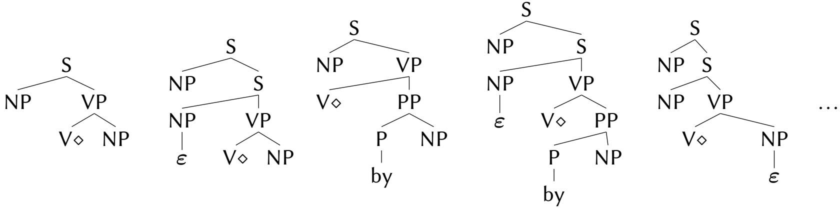
- Elementary trees can have complex semantic representations which are not necessarily derived compositionally (in the syntax) from smaller parts of the trees.

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

Tree families

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

Example unanchored family for transitive verbs

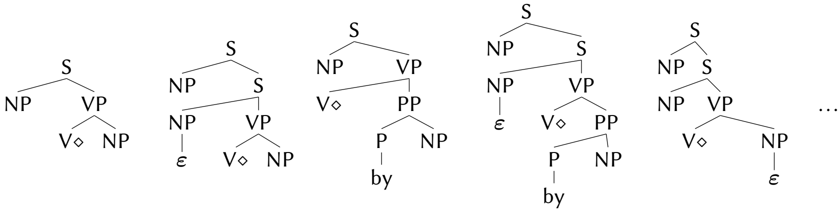


Lexicalized Tree Adjoining Grammars (LTAG)

Tree families

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

Example unanchored family for transitive verbs

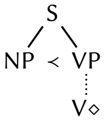


Metagrammar

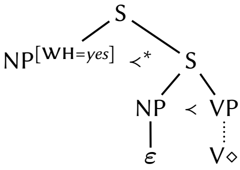
Modular characterization of elementary trees by a system of tree descriptions.

Decomposition/factorization in the metagrammar

Class *CanSubj*



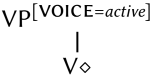
Class *ExtrSubj*



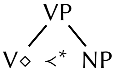
Class *Subj*

CanSubj ∨ *ExtSubj*

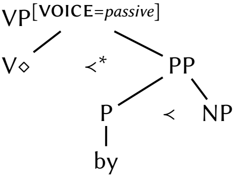
Class *ActV*



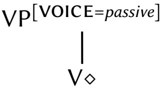
Class *DirObj*



Class *ByObj*

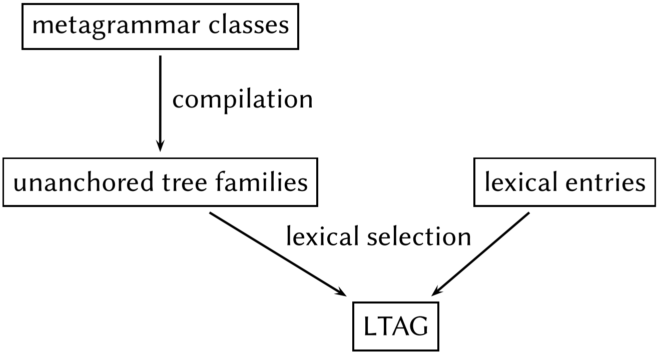


Class *PassV*



Lexicalized Tree Adjoining Grammars (LTAG)

Decomposition/factorization in the metagrammar

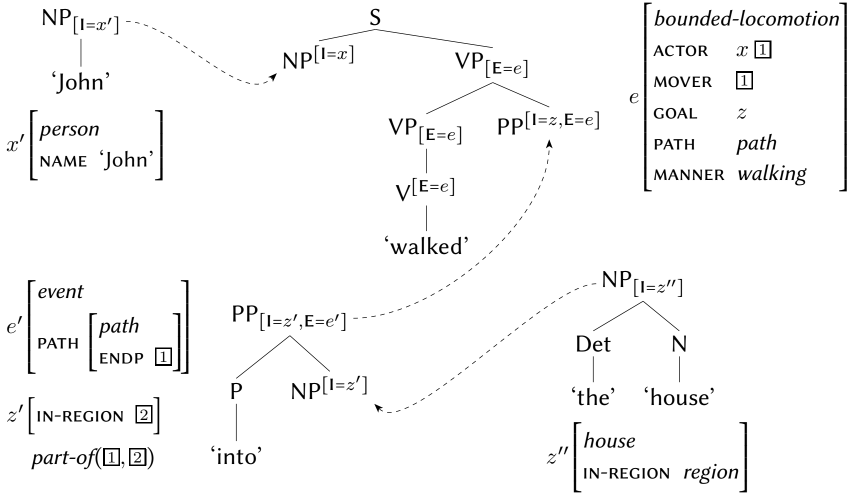


Advantage

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

Example

(4) John walked into the house.



LTAG and frames

Example (cont'd)

(5) a. John walked into the house.
  b. Mary kicked the ball into the room.

Metagrammar classes (syntax and semantics)

Class *n0Vpp(dir)*
identities: $C_1.e = C_2.e$

- Class C_1 =*n0V*
  export: e
  …
- Class C_2 =*DirPrepObj*
  export: e, z
  …

Class *n0Vn1pp(dir)*
identities: $C_1.e = e$, $C_2.z = z$, $C_2.e = e'$

- Class C_1 =*n0Vn1*
  export: e
  …
- Class C_2 =*DirPrepObj*
  export: e, z
  …

$$e\begin{bmatrix} \textit{causation} & \\ \text{ACTOR} & x\ \boxed{1} \\ \text{THEME} & y\ \boxed{2} \\ \text{CAUSE} & \begin{bmatrix} \textit{activity} & \\ \text{ACTOR} & \boxed{1} \\ \text{THEME} & \boxed{2} \end{bmatrix} \\ \text{EFFECT} & e'\begin{bmatrix} \text{MOVER} & \boxed{2} \\ \text{GOAL} & z \end{bmatrix} \end{bmatrix}$$

Implementation

XMG metagrammar compiler extended with semantic frame specifications

[Lichte/Petitjean 2015]

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?

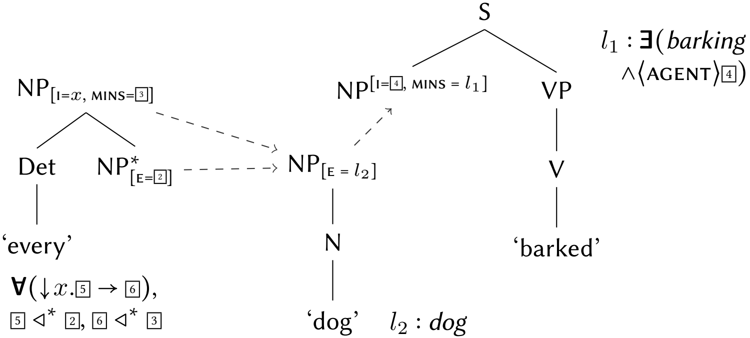
Possible approaches

1. Keep frames as basic semantic representations and evaluate quantification over the domain of frames. [≈ Muskens 2013]
2. Use an attribute-value language with quantifiers (e.g. Hybrid Logic), and build formulas instead of models. [e.g., Kallmeyer/Osswald/Pogodalla 2016]
3. Try to retain the idea of minimal model building and consider **frame types** as proper entities of the model/universe.

Frame semantics: extensions

Hybrid Logic + underspecification ("hole semantics")

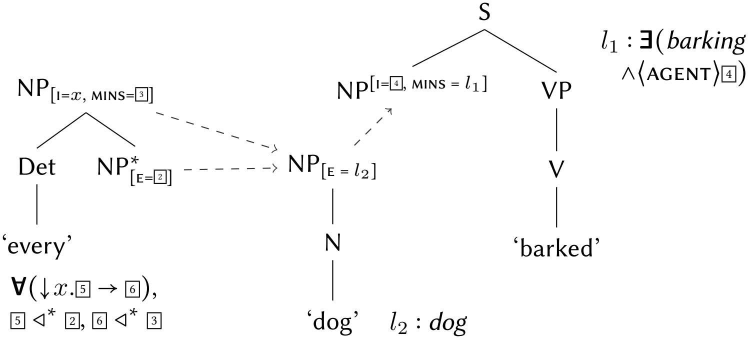
(6) Every dog barked.



Frame semantics: extensions

Hybrid Logic + underspecification (“hole semantics”)

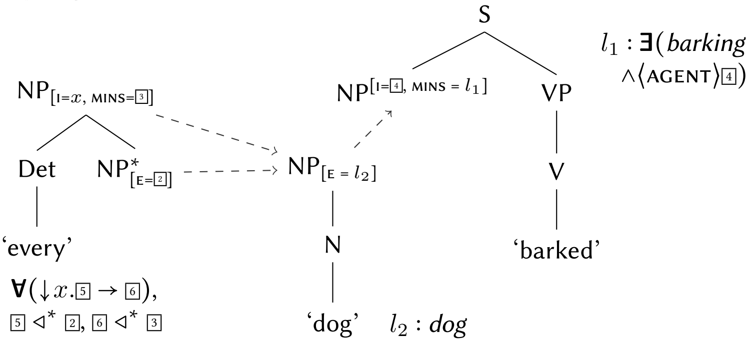
(6) Every dog barked.



$\rightsquigarrow\ \forall(\downarrow x.\boxed{5} \rightarrow \boxed{6}),\ l_2 : \mathit{dog},\ l_1 : \exists(\mathit{barking} \wedge \langle\textsc{agent}\rangle x),\ \boxed{5} \vartriangleleft^{*} l_2, \boxed{6} \vartriangleleft^{*} l_1$

Hybrid Logic + underspecification ("hole semantics")

(6) Every dog barked.

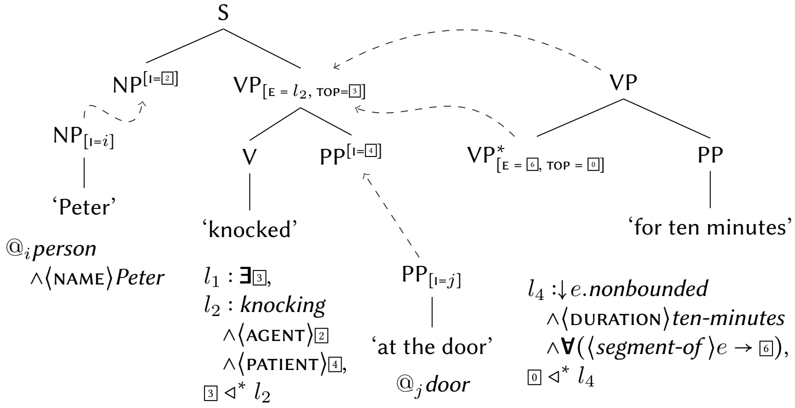


$\rightsquigarrow\ \forall(\downarrow x.\boxed{5} \to \boxed{6}),\ l_2 : \mathit{dog},\ l_1 : \exists(\mathit{barking} \wedge \langle\textsc{agent}\rangle x),\ \boxed{5} \lhd^* l_2, \boxed{6} \lhd^* l_1$

$\rightsquigarrow\ \forall(\downarrow x.\ \mathit{dog} \to \exists(\mathit{barking} \wedge \langle\textsc{agent}\rangle x))$

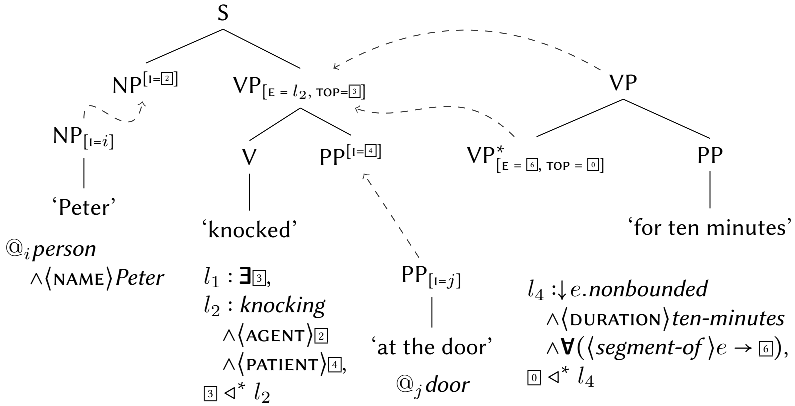
Hybrid Logic + underspecification ("hole semantics")

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



Hybrid Logic + **underspecification** (“hole semantics”)

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



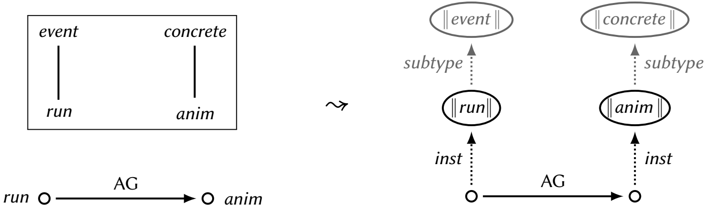
$$\rightsquigarrow \quad \exists(\downarrow e.\textit{nonbounded} \wedge \langle \textsc{duration}\rangle \textit{ten-minutes} \wedge \forall(\langle \textit{segment-of}\rangle e \rightarrow \textit{knocking} \wedge \langle \textsc{agent}\rangle i \wedge \langle \textsc{patient}\rangle j)) \wedge @_i(\textit{person} \wedge \langle \textsc{name}\rangle \textit{Peter}) \wedge @_j \textit{door}$$

Frame semantics: extensions

Frame types (sketch/work in progress)

Frame types (sketch/work in progress)

- **Types as elements of the universe/model**

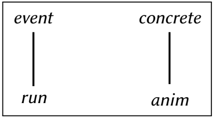


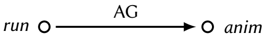
‖*event*‖, ‖*run*‖, etc.: type names (nominals)

Frame semantics: extensions

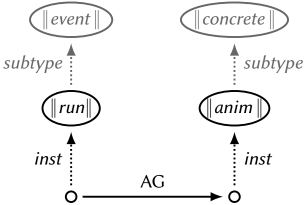
Frame types (sketch/work in progress)

- **Types as elements of the universe/model**





$\rightsquigarrow$

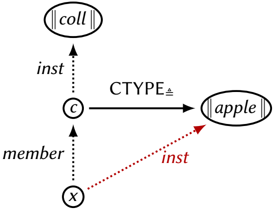


$\|event\|$, $\|run\|$, etc.: type names (nominals)

- **Types as values of attributes**

  Example: collections of elements of type T

  $c \cdot \text{CTYPE} \triangleq T \wedge x \; member \; c \rightarrow x \; inst \; T$



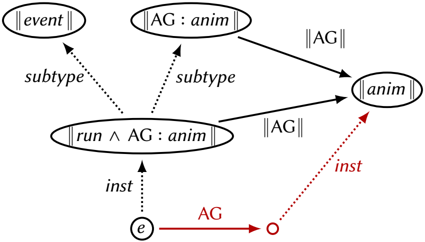
Frame types (sketch/work in progress)

- **Complex frame types**

  Introduce **frame types** like $\| \mathrm{P} : t \|$

  Frame types can have (canonical) attributes, e.g., $\| \mathrm{P} : t \| \cdot \| \mathrm{P} \| \triangleq \| t \|$

  $n \;\; inst \;\; \| \mathrm{P} : t \| \;\; \leftrightarrow \;\; n \cdot \mathrm{P} \;\; inst \;\; \| t \|$



Frame semantics: extensions

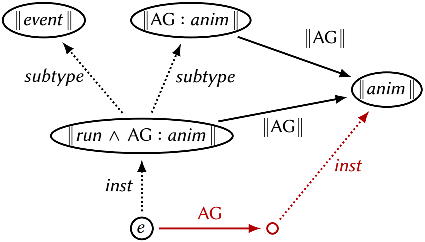
Frame types (sketch/work in progress)

- **Complex frame types**

  Introduce **frame types** like $\|P : t\|$

  Frame types can have (canonical) attributes, e.g., $\|P : t\| \cdot \|P\| \triangleq \|t\|$

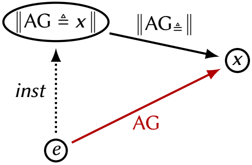
  $n \;\; inst \;\; \|P : t\| \;\leftrightarrow\; n \cdot P \;\; inst \;\; \|t\|$



- **Dependent frame types**

  $n \;\; inst \;\; \|P \triangleq x\| \;\leftrightarrow\; n \cdot P \triangleq x$

  ($\|P \triangleq x\|$ frame type "dependent" on x)



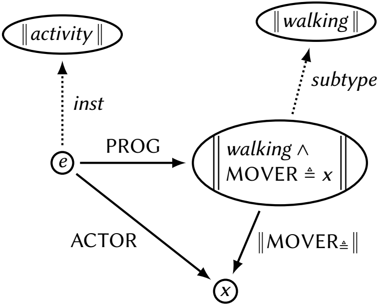
Frame semantics: extensions

Frame types (sketch/work in progress)

- Example: event progression

  $e \cdot \textsc{prog} \triangleq T \wedge\ e'\ \mathit{segment}\ e\ \rightarrow\ e'\ \mathit{inst}\ T$

$$e\begin{bmatrix} \mathit{activity} & \\ \textsc{actor} & x \\ \textsc{prog} & \left\| \begin{bmatrix} \mathit{walking} & \\ \textsc{mover} & x \end{bmatrix} \right\| \end{bmatrix}$$

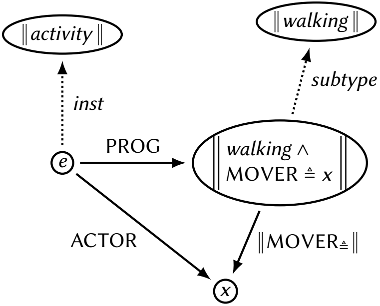


Frame types (sketch/work in progress)

- Example: event progression

$$e \cdot \textsc{prog} \triangleq T \wedge\ e'\ \mathit{segment}\ e\ \rightarrow\ e'\ \mathit{inst}\ T$$

$$e\begin{bmatrix} \textit{activity} & \\ \textsc{actor} & x \\ \textsc{prog} & \left\| \begin{bmatrix} \textit{walking} & \\ \textsc{mover} & x \end{bmatrix} \right\| \end{bmatrix}$$



- Example: scalar change

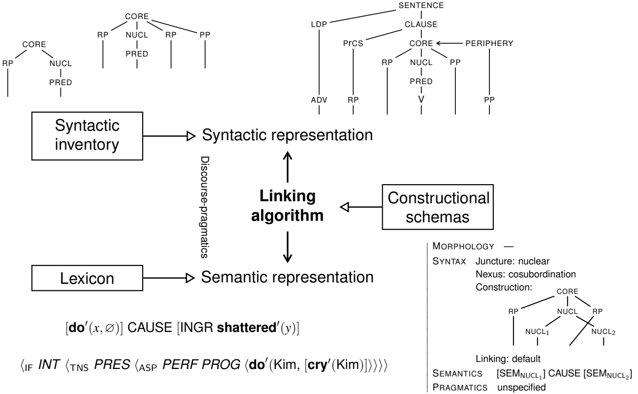
$$\begin{bmatrix} \textit{progression} & \\ \textsc{entity} & x \\ \textsc{prog} & \left\| \begin{bmatrix} \textit{incremental-change} & \\ \textsc{entity} & x \\ \textsc{initial} & \begin{bmatrix} \textit{stage} & \\ \textsc{entity} & x \\ \textsc{length} & \boxed{0} \end{bmatrix} \\ \textsc{final} & \begin{bmatrix} \textit{stage} & \\ \textsc{entity} & x \\ \textsc{length} & \boxed{1} \end{bmatrix} \end{bmatrix} \\ \textit{lesser}(\boxed{1}, \boxed{0}) \right\| \end{bmatrix}$$

Further ongoing work

Formalization of Role and Reference Grammar

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

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



Further ongoing work

Formalization of Role and Reference Grammar

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

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.

  [Osswald/Kallmeyer, to appear]

- Decompositional 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

Babonnaud, W., Kallmeyer, L., and Osswald, R. (2016). Polysemy and coercion – a frame-based approach using LTAG and Hybrid Logic. In Amblard, M. et al., eds., *Logical Aspects of Computational Linguistics, 9th International Conference*, 18–33. Springer.

Bangalore, S. and Joshi, A. K. (2010). Introduction. In Bangalore, S. and Joshi, A. K., eds., *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*, 1–31. MIT Press.

Barsalou, L. W. (1992). Frames, concepts, and conceptual fields. In Lehrer, A. and Kittay, E. F., eds., *Frames, Fields, and Contrasts*, 21–74. Lawrence Erlbaum.

Crabbé, B. and Duchier, D. (2005). Metagrammar redux. In Christiansen, H. et al., eds., *Constraint Solving and Language Processing*, 32–47. Springer.

Crabbé, B., Duchier, D., Gardent, C., Le Roux, J., and Parmentier, Y. (2013). XMG: eXtensible MetaGrammar. *Computational Linguistics*, 39(3):591–629.

Hegner, S. J. (1994). Properties of Horn clauses in feature-structure logic. In Rupp, C. J. et al., eds., *Constraints, Language and Computation*, 111–147. Academic Press.

Joshi, A. K. and Schabes, Y. (1997). Tree-adjoining grammars. In Rozenberg, G. and Salomaa, A., eds., *Handbook of Formal Languages. Vol. 3: Beyond Words*, 69–123. Springer.

Kallmeyer, L., Lichte, T., Osswald, R., and Petitjean, S. (2016a). 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, L. and Osswald, R. (2012). A frame-based semantics of the dative alternation in Lexicalized Tree Adjoining Grammars. In Piñón, C., ed., *Empirical Issues in Syntax and Semantics 9*, 167–184.

Kallmeyer, L. and Osswald, R. (2013). Syntax-driven semantic frame composition in Lexicalized Tree Adjoining Grammars. *Journal of Language Modelling*, 1(2):267–330.

Kallmeyer, L., Osswald, R., and Pogodalla, S. (2016). *For*-adverbials and aspectual interpretation: An LTAG analysis using Hybrid Logic and Frame Semantics. In Piñón, C., ed., *Empirical Issues in Syntax and Semantics 11*.

Löbner, S. (2014). Evidence for frames from human language. In Gamerschlag, T. et al., eds., *Frames and Concept Types*, 23–67. Springer.

Muskens, R. (2013). Data semantics and linguistic semantics. In Aloni, M. et al., eds., *The dynamic, inquisitive, and visionary life of ϕ, $?\phi$, and $\diamond\phi$. A festschrift for Jeroen Groenendijk, Martin Stokhof, and Frank Veltman*, 175–183.

Osswald, R. (1999). Semantics for attribute-value theories. In Dekker, P., ed., *Proceedings of the Twelfth Amsterdam Colloquium*, 199–204, Amsterdam. ILLC.

Osswald, R. and Kallmeyer, L. (to appear). Towards a formalization of Role and Reference Grammar. In Kailuweit, R. et al., eds., *Applying and Expanding Role and Reference Grammar*.

Osswald, R. and Van Valin, Jr., R. D. (2014). FrameNet, frame structure, and the syntax-semantics interface. In Gamerschlag, T. et al., eds., *Frames and Concept Types*, 125–156. Springer.

Petersen, W. (2007). Representation of concepts as frames. In Skilters, J. et al., eds., *The Baltic International Yearbook of Cognition, Logic and Communication*, volume 2, 151–170. University of Latvia.

Van Valin, Jr., R. D. (2005). *Exploring the Syntax-Semantics Interface*. Cambridge University Press.