Grammar Implementation with Lexicalized Tree Adjoining Grammars and Frame Semantics

Frame semantics

Laura Kallmeyer, Timm Lichte, Rainer Osswald & Simon Petitjean

University of Düsseldorf

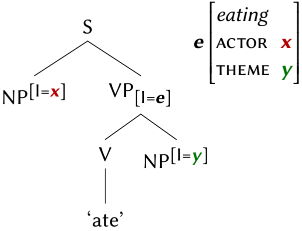
DGfS CL Fall School, September 14, 2017

SFB 991

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Reminder

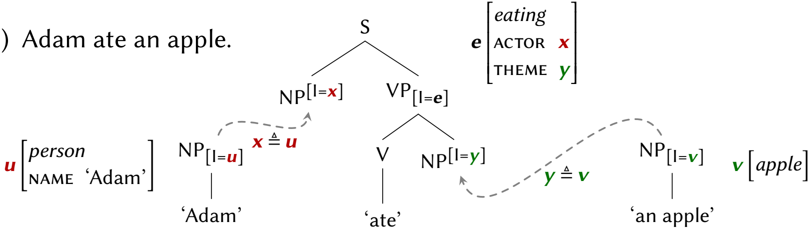
(1) Adam ate an apple.



The overall story

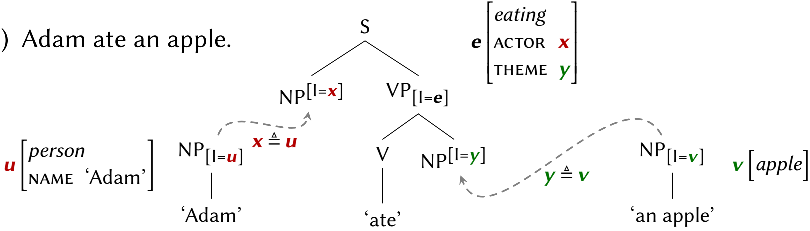
Reminder

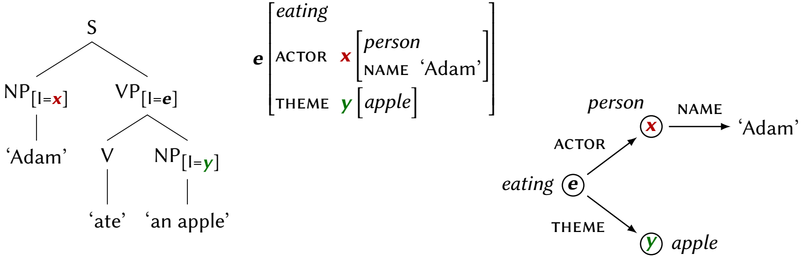
(1) Adam ate an apple.



Reminder

(1) Adam ate an apple.





Outline of today's course

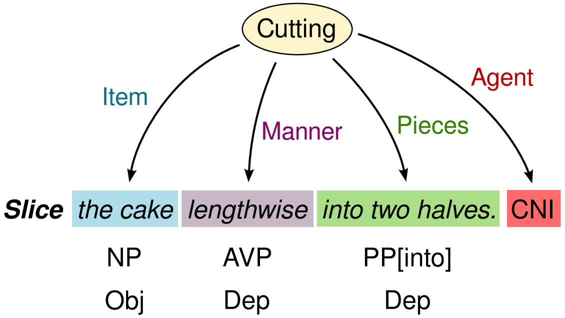
1. Introduction to frame semantics
   - Frames in the sense of Fillmore and Barsalou
   - Frames according to this course
2. Formalization of frames
   - Attribute-value descriptions and formulas
   - Formal definition of frames
   - Frames as models
   - Subsumption and unification
   - Attribute-value constraints
3. Further topics
   - Frames versus feature structures
   - Type constraints versus type hierarchy
4. Frame semantics: extensions
5. Summary and outlook

Outline of today's course

1. Introduction to frame semantics
   - Frames in the sense of Fillmore and Barsalou
   - Frames according to this course
2. Formalization of frames
   - Attribute-value descriptions and formulas
   - Formal definition of frames
   - Frames as models
   - Subsumption and unification
   - Attribute-value constraints
3. Further topics
   - Frames versus feature structures
   - Type constraints versus type hierarchy
4. Frame semantics: extensions
5. Summary and outlook

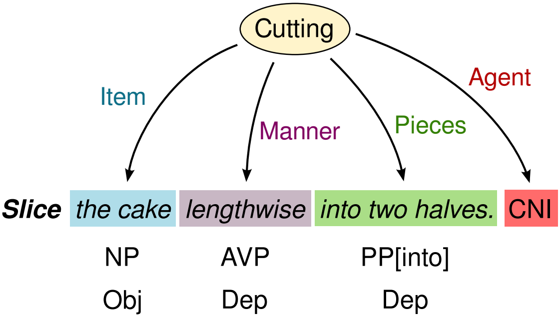
Frames according to Fillmore/FrameNet [framenet.icsi.berkeley.edu]

The 'Cutting' frame, annotated:



Frames according to Fillmore/FrameNet [framenet.icsi.berkeley.edu]

The ‘Cutting’ frame, annotated:



The FrameNet database:

- > 1200 frames
- > 13000 lexical units (= word senses)

Frames according to Fillmore/FrameNet [framenet.icsi.berkeley.edu]

Cutting frame

Definition: An [Agent] cuts an [Item] into [Pieces] using an [Instrument] (which may or may not be expressed).

Core frame elements:

Agent	The [Agent] is the person cutting the [Item] into [Pieces].
Item	The item which is being cut into [Pieces].
Pieces	The [Pieces] are the parts of the original [Item] which are the result of the slicing.

Non-core frame elements:

Instrument	The [Instrument] with which the [Item] is being cut into [Pieces].
Manner	[Manner] in which the [Item] is being cut into [Pieces].
Result	The [Result] of the [Item] being sliced into [Pieces]. (extrathematic)
In addition: Means, Purpose, Place, Time	

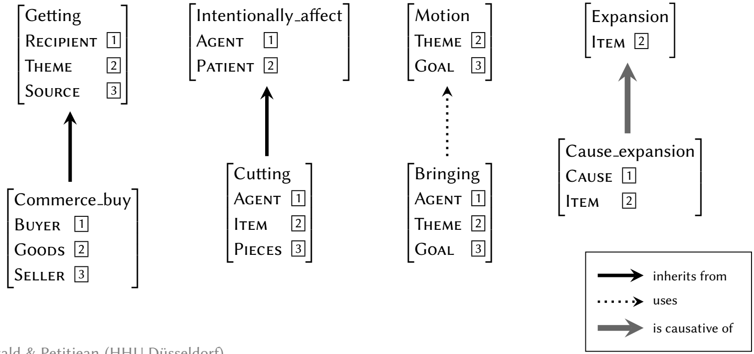
Lexical units: *carve*, *chop*, *cube*, *cut*, *dice*, *fillet*, *mince*, *pare*, *slice*

Frames according to Fillmore/FrameNet [framenet.icsi.berkeley.edu]

Frame-to-frame relations in FrameNet

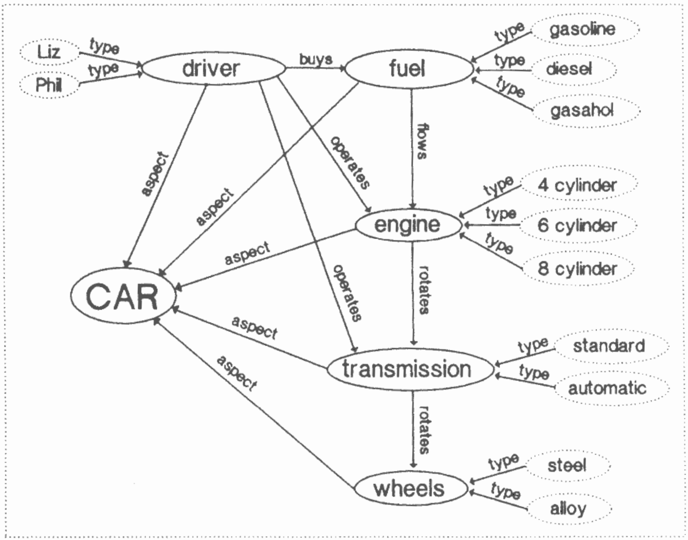
- Generalization relations: ‘inherits from’, ‘is perspective on’, ‘uses’
- Event structure relations: ‘is subframe of’, ‘precedes’
- Systematic relations: ‘is causative of’, ‘is inchoative of’

Examples



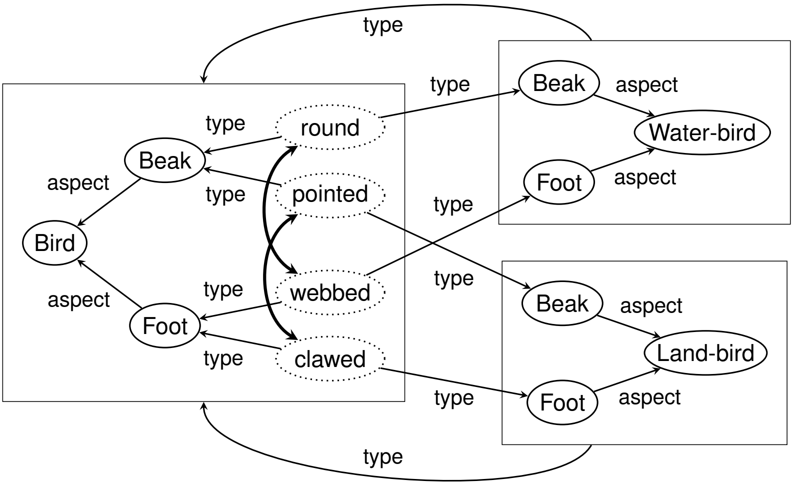
Frames according to Barsalou

[Barsalou 1992:30]



Frames according to Barsalou

[Gamerschlag et al. 2014:6]



Introduction to frame semantics

Frames according to this course [Kallmeyer/Osswald 2013; Osswald/Van Valin 2014]

- A representation format for **rich lexical** and **constructional content**.
- Can nicely capture **semantic composition** and **decomposition**.
- Can be formalized as **generalized feature structures** with **types**, **relations** and **node labels**.

Introduction to frame semantics

Frames according to this course [Kallmeyer/Osswald 2013; Osswald/Van Valin 2014]

- A representation format for **rich lexical** and **constructional content**.
- Can nicely capture **semantic composition** and **decomposition**.
- Can be formalized as **generalized feature structures** with **types**, **relations** and **node labels**.

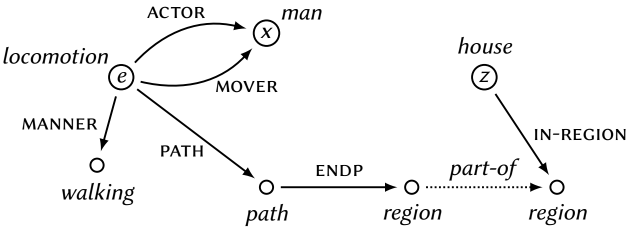
Basic assumptions

- **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 via **attributes** (from some "**base**" node).

  ⤳ inherently **structured representations** (models);
  composition by **unification** (under **constraints**)

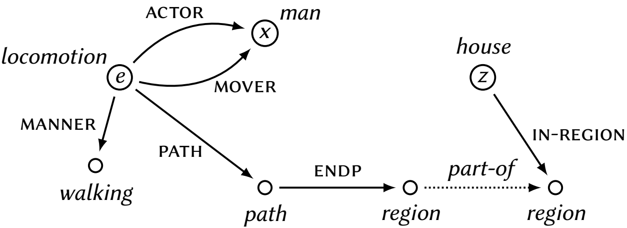
Frames according to this course

Example



Frames according to this course

Example

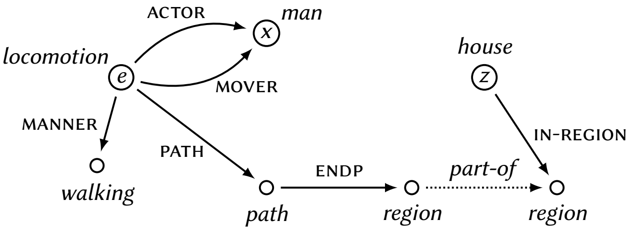


Ingredients

- Attributes (funct. relations): ACTOR, MOVER, PATH, MANNER, IN-REGION, …

Frames according to this course

Example

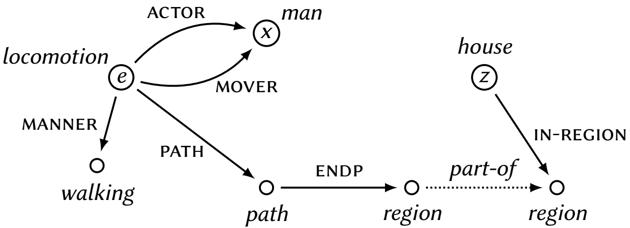


Ingredients

- Attributes (funct. relations): ACTOR, MOVER, PATH, MANNER, IN-REGION, …
- Type symbols: *locomotion*, *man*, *path*, *walking*, *region*, …

Frames according to this course

Example

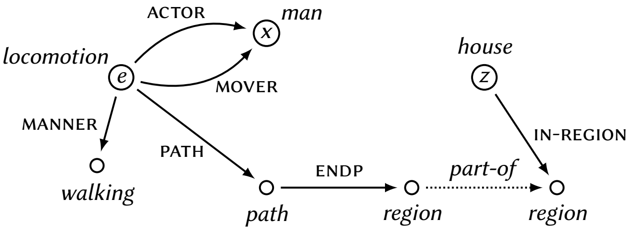


Ingredients

- Attributes (funct. relations): ACTOR, MOVER, PATH, MANNER, IN-REGION, …
- Type symbols: *locomotion*, *man*, *path*, *walking*, *region*, …
- Proper relations: *part-of*

Frames according to this course

Example



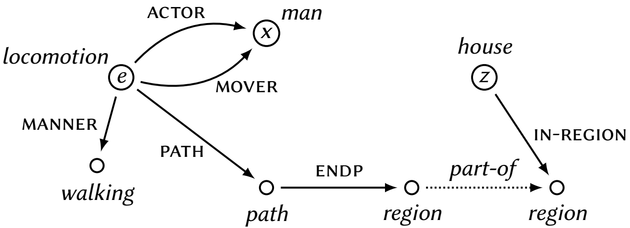
Ingredients

- Attributes (funct. relations): ACTOR, MOVER, PATH, MANNER, IN-REGION, …
- Type symbols: *locomotion*, *man*, *path*, *walking*, *region*, …
- Proper relations: *part-of*
- Node labels (variables): e, x, z

Introduction to frame semantics

Frames according to this course

Example



Ingredients

- Attributes (funct. relations): ACTOR, MOVER, PATH, MANNER, IN-REGION, …
- Type symbols: *locomotion*, *man*, *path*, *walking*, *region*, …
- Proper relations: *part-of*
- Node labels (variables): *e*, *x*, *z*

Core property

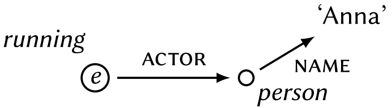
- Every node is reachable from some labeled "base" node via attributes.

Example

(2) Anna ran

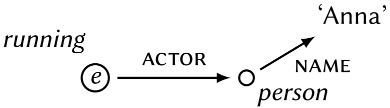
Example

(2) Anna ran



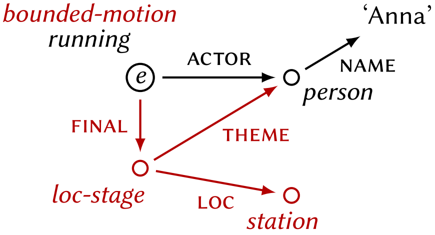
Example

(2) Anna ran to the station.



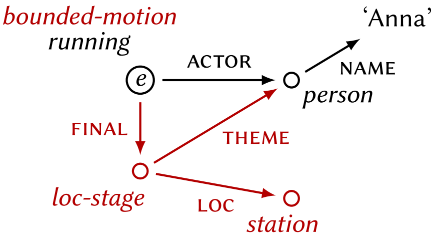
Example

(2) Anna ran to the station.



Example

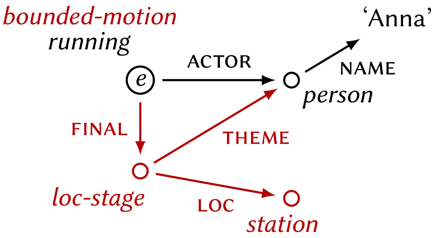
(2) Anna ran to the station.



$$e \begin{bmatrix} \textit{running} \wedge \textit{bounded-motion} \\ \text{ACTOR} \; \boxed{1} \begin{bmatrix} \textit{person} \\ \text{NAME} \; \text{`Anna'} \end{bmatrix} \\ \text{FINAL} \; \begin{bmatrix} \textit{loc-stage} \\ \text{THEME} \; \boxed{1} \\ \text{LOC} \; [\textit{station}] \end{bmatrix} \end{bmatrix}$$

Example

(2) Anna ran to the station.



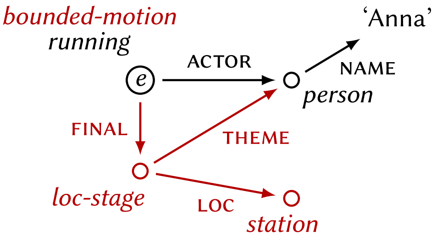
$$e\begin{bmatrix} \textit{running} \wedge \textit{bounded-motion} \\ \text{ACTOR}\ \boxed{1}\begin{bmatrix}\textit{person}\\ \text{NAME}\ \text{`Anna'}\end{bmatrix} \\ \text{FINAL}\ \begin{bmatrix}\textit{loc-stage}\\ \text{THEME}\ \boxed{1}\\ \text{LOC}\ [\textit{station}]\end{bmatrix}\end{bmatrix}$$

Attribute-value logic

$$e \cdot (\textit{running} \wedge \textit{bounded-motion} \wedge \text{ACTOR} : (\textit{person} \wedge \text{NAME} \triangleq \text{`Anna'})$$
$$\text{ACTOR} \doteq \text{FINAL THEME} \wedge \text{FINAL} : (\textit{loc-stage} \wedge \text{LOC} : \textit{station}))$$

Example

(2) Anna ran to the station.



$$e\begin{bmatrix} \textit{running} \wedge \textit{bounded-motion} & \\ \text{ACTOR} & \boxed{1}\begin{bmatrix} \textit{person} & \\ \text{NAME} & \text{`Anna'} \end{bmatrix} \\ \text{FINAL} & \begin{bmatrix} \textit{loc-stage} & \\ \text{THEME} & \boxed{1} \\ \text{LOC} & [\textit{station}] \end{bmatrix} \end{bmatrix}$$

Attribute-value logic

$$e \cdot (\textit{running} \wedge \textit{bounded-motion} \wedge \text{ACTOR} : (\textit{person} \wedge \text{NAME} \triangleq \text{`Anna'}) \\ \text{ACTOR} \doteq \text{FINAL THEME} \wedge \text{FINAL} : (\textit{loc-stage} \wedge \text{LOC} : \textit{station}))$$

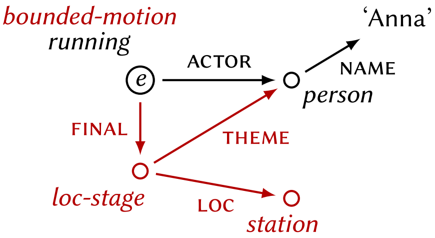
Translation into first-order logic

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

Introduction to frame semantics

Example

(2) Anna ran to the station.



$$e \begin{bmatrix} \textit{running} \wedge \textit{bounded-motion} & \\ \textsc{actor} & \boxed{1} \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{'Anna'} \end{bmatrix} \\ \textsc{final} & \begin{bmatrix} \textit{loc-stage} & \\ \textsc{theme} & \boxed{1} \\ \textsc{loc} & [\textit{station}] \end{bmatrix} \end{bmatrix}$$

Attribute-value logic

$$e \cdot (\textit{running} \wedge \textit{bounded-motion} \wedge \textsc{actor} : (\textit{person} \wedge \textsc{name} \triangleq \text{'Anna'}) \wedge$$
$$\textsc{actor} \doteq \textsc{final theme} \wedge \textsc{final} : (\textit{loc-stage} \wedge \textsc{loc} : \textit{station}))$$

Constraints

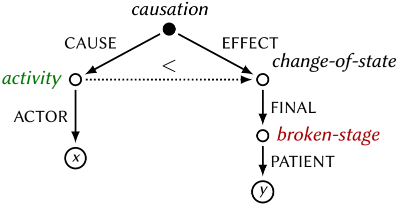
$$\textit{running} \Rrightarrow \textit{activity} \quad (\text{short for } \forall e(\textit{running}(e) \rightarrow \textit{activity}(e))),$$
$$\textit{loc-stage} \Rrightarrow \textsc{theme} : \top \wedge \textsc{loc} : \top, \quad \ldots$$

Example Lexical decomposition templates [Rappaport Hovav/Levin 1998]

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

Example Lexical decomposition templates [Rappaport Hovav/Levin 1998]

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



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

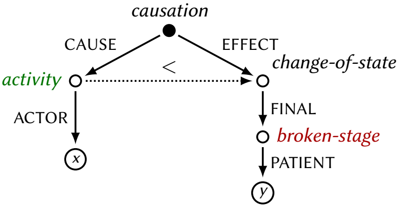
$\text{CAUSE} < \text{EFFECT}$

Description in attribute-value logic

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

Example Lexical decomposition templates [Rappaport Hovav/Levin 1998]

(3) [[*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

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

Translation into first-order logic

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

Outline of today's course

1. Introduction to frame semantics
   - Frames in the sense of Fillmore and Barsalou
   - Frames according to this course
2. Formalization of frames
   - Attribute-value descriptions and formulas
   - Formal definition of frames
   - Frames as models
   - Subsumption and unification
   - Attribute-value constraints
3. Further topics
   - Frames versus feature structures
   - Type constraints versus type hierarchy
4. Frame semantics: extensions
5. Summary and outlook

Attribute-value descriptions

Vocabulary / Signature

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

Attribute-value descriptions

Vocabulary / Signature

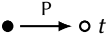
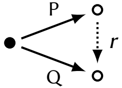
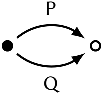
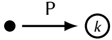
Attr	attributes (= dyadic functional relation symbols)		
Rel	(proper) relation symbols		
Type	type symbols (= monadic predicates)		
Nname	node names ("nominals")	} Nlabel	node labels
Nvar	node variables		

Primitive attribute-value descriptions (pAVDesc)

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

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

Semantics

$\text{P} : t$		$\begin{bmatrix} \text{P} & [t] \end{bmatrix}$	$[\text{P}, \text{Q}] : r$		$\begin{bmatrix} \text{P} & \boxed{1} \\ \text{Q} & \boxed{2} \end{bmatrix}$ $r(\boxed{1}, \boxed{2})$
$\text{P} \doteq \text{Q}$		$\begin{bmatrix} \text{P} & \boxed{1} \\ \text{Q} & \boxed{1} \end{bmatrix}$	$\text{P} \triangleq k$		$\begin{bmatrix} \text{P} & k[\;] \end{bmatrix}$

Attribute-value formulas

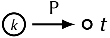
Primitive attribute-value formulas (pAVForm)

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

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

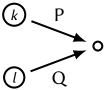
Semantics

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



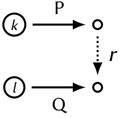
$k \begin{bmatrix} \mathrm{P} & [t] \end{bmatrix}$

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



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

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



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

Attribute-value formulas

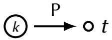
Primitive attribute-value formulas (pAVForm)

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

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

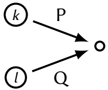
Semantics

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



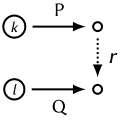
$k \left[\text{P} \; [t] \right]$

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



$k \left[\text{P} \; \boxed{1} \right]$
$l \left[\text{Q} \; \boxed{1} \right]$

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



$k \left[\text{P} \; \boxed{1} \right]$
$l \left[\text{Q} \; \boxed{2} \right]$
$r(\boxed{1}, \boxed{2})$

Formal definitions (fairly standard)

Set/universe of “nodes” V

Attribute-value formulas

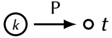
Primitive attribute-value formulas (pAVForm)

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

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

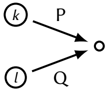
Semantics

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



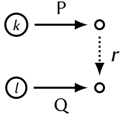
$k\begin{bmatrix}\text{P} & [t]\end{bmatrix}$

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



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

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



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

Formal definitions (fairly standard)

Set/universe of "nodes"	V
Interpretation function	$\mathcal{I} : \text{Attr} \to [V \rightharpoonup V]$,

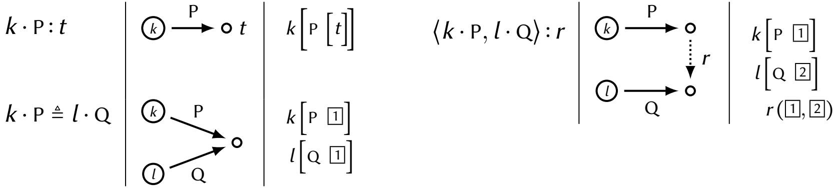
Attribute-value formulas

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, $p, q, p_i \in \mathrm{Attr}^*$, $k, l, k_i \in$ Nlabel)

Semantics



Formal definitions (fairly standard)

Set/universe of "nodes"	V
Interpretation function	$\mathcal{I}$: Attr $\rightarrow [V \rightharpoonup V]$, Type $\rightarrow \wp(V)$,

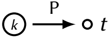
Primitive attribute-value formulas (pAVForm)

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

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

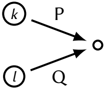
Semantics

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



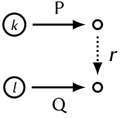
$k \left[\text{P} \left[t \right] \right]$

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



$k \left[\text{P} \; \boxed{1} \right]$
$l \left[\text{Q} \; \boxed{1} \right]$

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



$k \left[\text{P} \; \boxed{1} \right]$
$l \left[\text{Q} \; \boxed{2} \right]$
$r(\boxed{1}, \boxed{2})$

Formal definitions (fairly standard)

Set/universe of "nodes"	V
Interpretation function	$\mathcal{I}$: Attr $\rightarrow [V \rightharpoonup V]$, Type $\rightarrow \wp(V)$,
	Rel $\rightarrow \bigcup_n \wp(V^n)$,

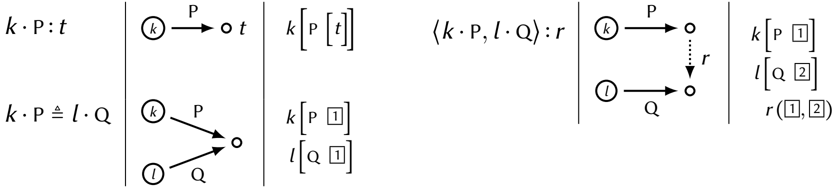
Attribute-value formulas

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, $p, q, p_i \in$ Attr*, $k, l, k_i \in$ Nlabel)

Semantics



Formal definitions (fairly standard)

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

Attribute-value formulas

Primitive attribute-value formulas (pAVForm)

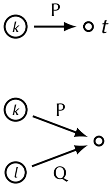
$$k \cdot p : t \;\mid\; k \cdot p \triangleq l \cdot q \;\mid\; \langle k_1 \cdot p_1, \ldots, k_n \cdot p_n \rangle : r$$

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

Semantics

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

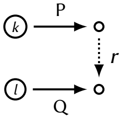
$k \cdot \text{P} \triangleq l \cdot \text{Q}$



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

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

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



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

Formal definitions (fairly standard)

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

Satisfaction of AV descriptions and formulas

Formal definitions (cont'd)

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

Satisfaction of AV descriptions and formulas

Formal definitions (cont'd)

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

Satisfaction of primitive **descriptions**

$\langle V, \mathcal{I}, g \rangle, v \vDash t$ iff $v \in \mathcal{I}(t)$

Satisfaction of AV descriptions and formulas

Formal definitions (cont'd)

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

Satisfaction of primitive **descriptions**

$\langle V, \mathcal{I}, g \rangle, v \vDash t$ iff $v \in \mathcal{I}(t)$

$\langle V, \mathcal{I}, g \rangle, v \vDash p : t$ iff $\mathcal{I}(p)(v) \in \mathcal{I}(t)$

Satisfaction of AV descriptions and formulas

Formal definitions (cont'd)

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

Satisfaction of primitive **descriptions**

$\langle V, \mathcal{I}, g\rangle, v \vDash t$ iff $v \in \mathcal{I}(t)$

$\langle V, \mathcal{I}, g\rangle, v \vDash p : t$ iff $\mathcal{I}(p)(v) \in \mathcal{I}(t)$

$\langle V, \mathcal{I}, g\rangle, v \vDash p \doteq q$ iff $\mathcal{I}(p)(v) = \mathcal{I}(q)(v)$

Satisfaction of AV descriptions and formulas

Formal definitions (cont'd)

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

Satisfaction of primitive **descriptions**

$\langle V, \mathcal{I}, g\rangle, v \vDash t$ iff $v \in \mathcal{I}(t)$

$\langle V, \mathcal{I}, g\rangle, v \vDash p : t$ iff $\mathcal{I}(p)(v) \in \mathcal{I}(t)$

$\langle V, \mathcal{I}, g\rangle, v \vDash p \doteq q$ iff $\mathcal{I}(p)(v) = \mathcal{I}(q)(v)$

$\langle V, \mathcal{I}, g\rangle, v \vDash [p_1, \ldots, p_n] : r$ iff $\langle \mathcal{I}(p_1)(v), \ldots, \mathcal{I}(p_n)(v)\rangle \in \mathcal{I}(r)$

Formal definitions (cont'd)

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

Satisfaction of primitive **descriptions**

$\langle V, \mathcal{I}, g \rangle, v \models t$ iff $v \in \mathcal{I}(t)$

$\langle V, \mathcal{I}, g \rangle, v \models p : t$ iff $\mathcal{I}(p)(v) \in \mathcal{I}(t)$

$\langle V, \mathcal{I}, g \rangle, v \models p \doteq q$ iff $\mathcal{I}(p)(v) = \mathcal{I}(q)(v)$

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

$\langle V, \mathcal{I}, g \rangle, v \models p \triangleq k$ iff $\mathcal{I}(p)(v) = \mathcal{I}_g(k)$ ($k \in$ Nlabel)

Satisfaction of AV descriptions and formulas

Formal definitions (cont'd)

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

Satisfaction of primitive **descriptions**

$\langle V, \mathcal{I}, g\rangle, v \vDash t$	iff $v \in \mathcal{I}(t)$
$\langle V, \mathcal{I}, g\rangle, v \vDash p : t$	iff $\mathcal{I}(p)(v) \in \mathcal{I}(t)$
$\langle V, \mathcal{I}, g\rangle, v \vDash p \doteq q$	iff $\mathcal{I}(p)(v) = \mathcal{I}(q)(v)$
$\langle V, \mathcal{I}, g\rangle, v \vDash [p_1, \ldots, p_n] : r$	iff $\langle \mathcal{I}(p_1)(v), \ldots, \mathcal{I}(p_n)(v)\rangle \in \mathcal{I}(r)$
$\langle V, \mathcal{I}, g\rangle, v \vDash p \triangleq k$	iff $\mathcal{I}(p)(v) = \mathcal{I}_g(k)$ ($k \in$ Nlabel)

Satisfaction of primitive **formulas**

$\langle V, \mathcal{I}, g\rangle \vDash k \cdot p : t$ iff $\mathcal{I}(p)(\mathcal{I}_g(k)) \in \mathcal{I}(t)$

Satisfaction of AV descriptions and formulas

Formal definitions (cont'd)

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

Satisfaction of primitive **descriptions**

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

Satisfaction of primitive **formulas**

$$\begin{array}{ll}
\langle V, \mathcal{I}, g\rangle \vDash k \cdot p : t & \text{iff } \mathcal{I}(p)(\mathcal{I}_g(k)) \in \mathcal{I}(t) \\
\langle V, \mathcal{I}, g\rangle \vDash k \cdot p \triangleq l \cdot q & \text{iff } \mathcal{I}(p)(\mathcal{I}_g(k)) = \mathcal{I}(q)(\mathcal{I}_g(l))
\end{array}$$

Satisfaction of AV descriptions and formulas

Formal definitions (cont'd)

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

Satisfaction of primitive **descriptions**

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

Satisfaction of primitive **formulas**

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

Satisfaction of AV descriptions and formulas

Formal definitions (cont'd)

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

Satisfaction of primitive **descriptions**

$\langle V, \mathcal{I}, g \rangle, v \vDash t$ iff $v \in \mathcal{I}(t)$

$\langle V, \mathcal{I}, g \rangle, v \vDash p : t$ iff $\mathcal{I}(p)(v) \in \mathcal{I}(t)$

$\langle V, \mathcal{I}, g \rangle, v \vDash p \doteq q$ iff $\mathcal{I}(p)(v) = \mathcal{I}(q)(v)$

$\langle V, \mathcal{I}, g \rangle, v \vDash [p_1, \ldots, p_n] : r$ iff $\langle \mathcal{I}(p_1)(v), \ldots, \mathcal{I}(p_n)(v) \rangle \in \mathcal{I}(r)$

$\langle V, \mathcal{I}, g \rangle, v \vDash p \triangleq k$ iff $\mathcal{I}(p)(v) = \mathcal{I}_g(k)$ ($k \in$ Nlabel)

Satisfaction of primitive **formulas**

$\langle V, \mathcal{I}, g \rangle \vDash k \cdot p : t$ iff $\mathcal{I}(p)(\mathcal{I}_g(k)) \in \mathcal{I}(t)$

$\langle V, \mathcal{I}, g \rangle \vDash k \cdot p \triangleq l \cdot q$ iff $\mathcal{I}(p)(\mathcal{I}_g(k)) = \mathcal{I}(q)(\mathcal{I}_g(l))$

$\langle V, \mathcal{I}, g \rangle \vDash \langle k_1 \cdot p_1, \ldots, k_n \cdot p_n \rangle : r$ iff $\langle \mathcal{I}(p_1)(\mathcal{I}_g(k_1)), \ldots, \mathcal{I}_g(p_n)(\mathcal{I}(k_n)) \rangle \in \mathcal{I}(r)$

Satisfaction of **Boolean combinations** as usual.

Frames defined

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,

Frames defined

Frame F over $\langle$Attr, Type, Rel, Nname, 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 $k \in$ Nlabel (= Nname $\cup$ Nvar) and

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

Frames defined

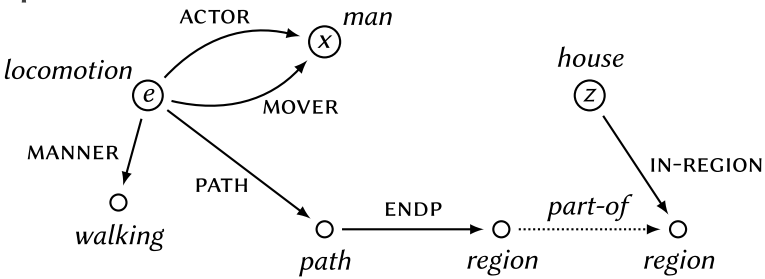
Frame F over $\langle$Attr, Type, Rel, Nname, Nvar$\rangle$:

$F = \langle V, \mathcal{I}, g \rangle$, 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 $k \in$ Nlabel (= Nname $\cup$ Nvar) and

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

Example



Frames as models of AV formulas

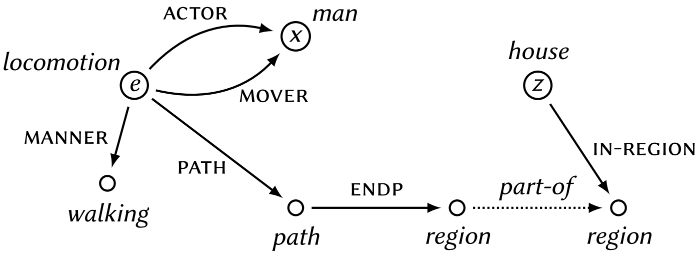
A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an AV formula ϕ iff $F \vDash \phi$.

Frames as models of AV formulas

A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an AV formula ϕ iff $F \vDash \phi$.

Example

F =

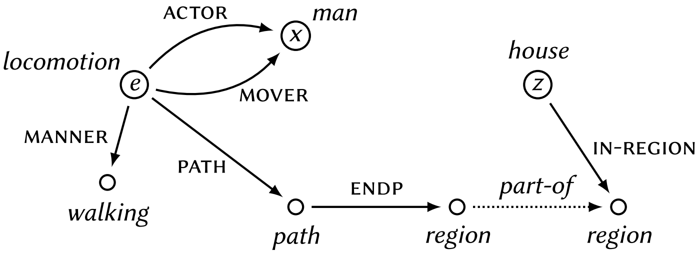


Frames as models of AV formulas

A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an AV formula ϕ iff $F \vDash \phi$.

Example

$F =$

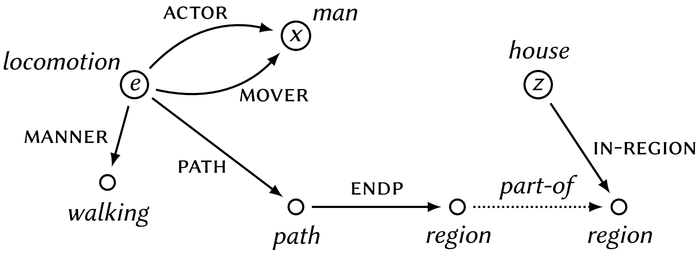


$F \vDash$

Frames as models of AV formulas

A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an AV formula ϕ iff $F \vDash \phi$.

Example

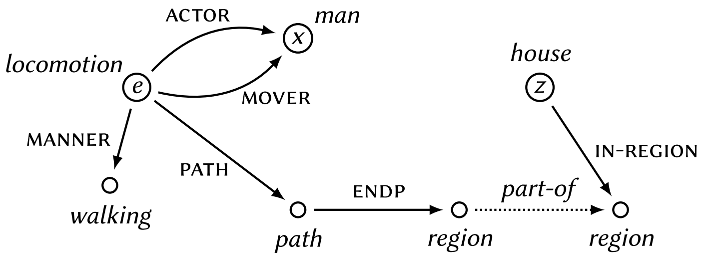
$F =$ 

$F \vDash e \cdot locomotion$

Frames as models of AV formulas

A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an AV formula ϕ iff $F \vDash \phi$.

Example

$F =$ 

$F \vDash e \cdot locomotion$

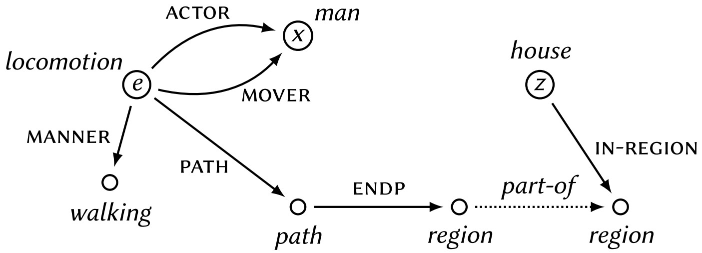
$F \vDash e \cdot (locomotion \wedge \text{ACTOR} : man)$

Frames as models of AV formulas

A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an AV formula ϕ iff $F \models \phi$.

Example

$F =$



$F \models e \cdot \textit{locomotion}$

$F \models e \cdot (\textit{locomotion} \wedge \text{ACTOR} : \textit{man})$

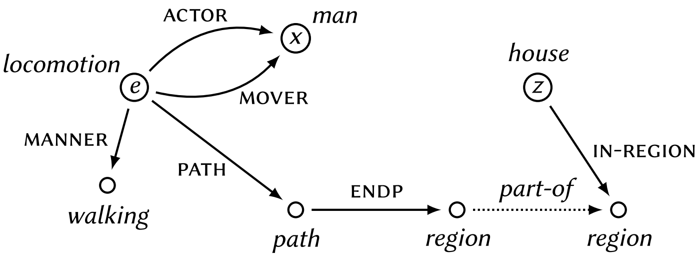
$F \models e \cdot (\textit{locomotion} \wedge \text{ACTOR} \triangleq x)$

Frames as models of AV formulas

A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an AV formula ϕ iff $F \vDash \phi$.

Example

$F =$



$F \vDash e \cdot \textit{locomotion}$

$F \vDash e \cdot (\textit{locomotion} \wedge \textsc{actor} : \textit{man})$

$F \vDash e \cdot (\textit{locomotion} \wedge \textsc{actor} \triangleq x)$

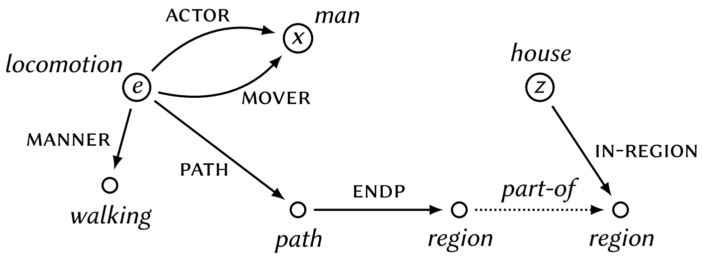
$F \vDash x \cdot \textit{man}$

Frames as models of AV formulas

A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an AV formula ϕ iff $F \vDash \phi$.

Example

$F =$



$F \vDash e \cdot locomotion$

$F \vDash e \cdot (locomotion \wedge \text{ACTOR} : man)$

$F \vDash e \cdot (locomotion \wedge \text{ACTOR} \triangleq x)$

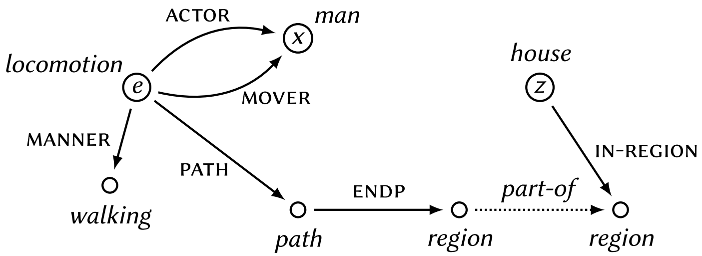
$F \vDash x \cdot man \wedge z \cdot house$

Frames as models of AV formulas

A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an AV formula ϕ iff $F \vDash \phi$.

Example

F =



$F \vDash e \cdot locomotion$

$F \vDash e \cdot (locomotion \wedge \text{ACTOR} : man)$

$F \vDash e \cdot (locomotion \wedge \text{ACTOR} \triangleq x)$

$F \vDash x \cdot man \wedge z \cdot house$

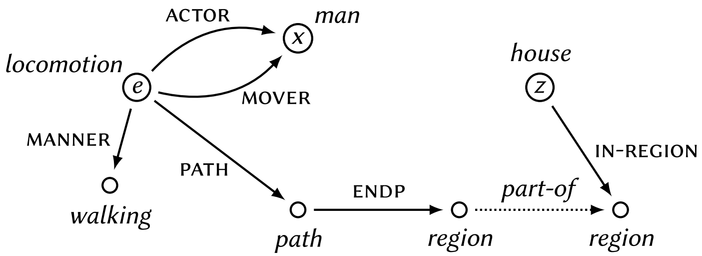
$F \vDash e \cdot (\text{ACTOR} \triangleq \text{MOVER})$

Frames as models of AV formulas

A frame $F = \langle V, \mathcal{I}, g \rangle$ is a **model** of an AV formula ϕ iff $F \vDash \phi$.

Example

$F =$



$F \vDash e \cdot locomotion$

$F \vDash e \cdot (locomotion \wedge \textsc{actor} : man)$

$F \vDash e \cdot (locomotion \wedge \textsc{actor} \triangleq x)$

$F \vDash x \cdot man \wedge z \cdot house$

$F \vDash e \cdot (\textsc{actor} \triangleq \textsc{mover})$

$F \vDash \langle e \cdot \textsc{path endp}, z \cdot \textsc{in-region} \rangle : part\text{-}of$

Subsumption and unification

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, $f \in$ Attr, $v \in V_1$,

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

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

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

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

Subsumption and unification

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, $f \in \text{Attr}$, $v \in V_1$,

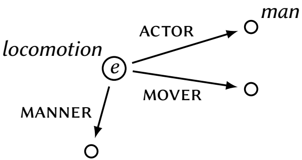
(ii) $h(\mathcal{I}_1(t)) \subseteq \mathcal{I}_2(t)$, for $t \in \text{Type}$

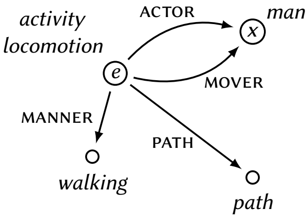
(iii) $h(\mathcal{I}_1(r)) \subseteq \mathcal{I}_2(r)$, for $r \in \text{Rel}$

(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





Subsumption and unification

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** $h : F_1 \rightarrow F_2$, i.e., a function $h : V_1 \rightarrow V_2$ such that

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

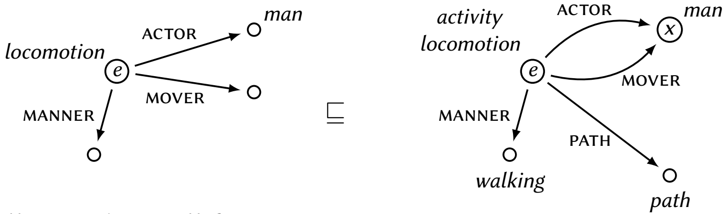
(ii) $h(\mathcal{I}_1(t)) \subseteq \mathcal{I}_2(t)$, for $t \in \text{Type}$

(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



Subsumption and unification

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, $f \in$ Attr, $v \in V_1$,

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

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

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

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

Intuition

F_1 subsumes F_2 ($F_1 \sqsubseteq F_2$) iff F_2 is **at least as informative** as F_1.

Subsumption and unification

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

Unification

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

Subsumption and unification

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

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.

Frames as minimal models

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.

Frames as minimal models

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.

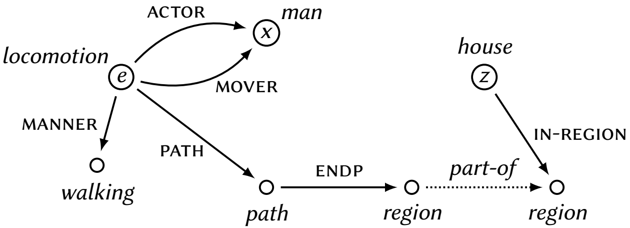
(ii) Every finite conjunction of primitive attribute-value formulas has a minimal frame model.

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



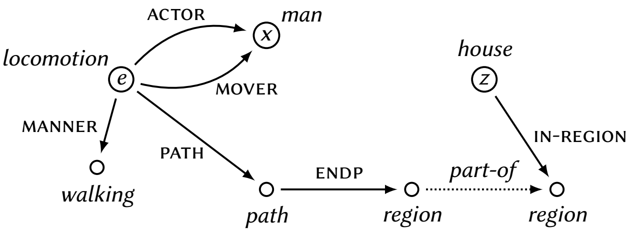
Frames as minimal models

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



$$e \cdot (\textit{locomotion} \;\wedge\; \text{MANNER} : \textit{walking} \;\wedge\; \text{ACTOR} \triangleq x$$
$$\wedge\; \text{MOVER} \doteq \text{ACTOR} \;\wedge\; \text{PATH} : (\textit{path} \;\wedge\; \text{ENDP} : \textit{region}))$$
$$\wedge\; \langle e \cdot \text{PATH ENDP},\; z \cdot \text{IN-REGION} \rangle : \textit{part-of} \;\wedge\; x \cdot \textit{man}$$

Attribute-value constraints

Constraints (general format) $\forall\phi,\ \phi \in \text{AVDesc}$

Attribute-value constraints

Constraints (general format) $\forall\phi,\ \phi \in$ AVDesc

$\langle V, \mathcal{I}, g \rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g \rangle, v \vDash \phi$ for every $v \in V$

Attribute-value constraints

Constraints (general format) $\forall\phi,\ \phi \in \text{AVDesc}$

$\langle V, \mathcal{I}, g \rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g \rangle, v \vDash \phi$ for every $v \in V$

Notation:

$\phi \Rrightarrow \psi$ for $\forall(\phi \rightarrow \psi)$

Attribute-value constraints

Constraints (general format) $\forall\phi,\ \phi \in \text{AVDesc}$

$\langle V, \mathcal{I}, g \rangle \models \forall\phi$ iff $\langle V, \mathcal{I}, g \rangle, v \models \phi$ for every $v \in V$

Notation:

$\phi \Rrightarrow \psi$ for $\forall(\phi \rightarrow \psi)$

Horn constraints:

$\phi_1 \wedge \ldots \wedge \phi_n \Rrightarrow \psi \qquad (\phi_i \in \text{pAVDesc} \cup \{\top\},\ \psi \in \text{pAVDesc} \cup \{\bot\})$

Attribute-value constraints

Constraints (general format) $\forall\phi, \ \phi \in \text{AVDesc}$

$\langle V, \mathcal{I}, g \rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g \rangle, v \vDash \phi$ for every $v \in V$

Notation:

$\phi \Rrightarrow \psi$ for $\forall(\phi \rightarrow \psi)$

Horn constraints:

$\phi_1 \wedge \ldots \wedge \phi_n \Rrightarrow \psi \qquad (\phi_i \in \text{pAVDesc} \cup \{\top\}, \ \psi \in \text{pAVDesc} \cup \{\bot\})$

Examples

$\textit{activity} \Rrightarrow \textit{event}$

$\textit{causation} \wedge \textit{activity} \Rrightarrow \bot$

$\text{AGENT} : \top \Rrightarrow \text{AGENT} \doteq \text{ACTOR}$

$\textit{activity} \Rrightarrow \text{ACTOR} : \top$

$\textit{activity} \wedge \textit{motion} \Rrightarrow \text{ACTOR} \doteq \text{MOVER}$

Attribute-value constraints

Constraints (general format) $\forall\phi,\ \phi \in \text{AVDesc}$

$\langle V, \mathcal{I}, g\rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g\rangle, v \vDash \phi$ for every $v \in V$

Notation:

$\phi \Rrightarrow \psi$ for $\forall(\phi \to \psi)$

Horn constraints:

$\phi_1 \wedge \ldots \wedge \phi_n \Rrightarrow \psi \qquad (\phi_i \in \text{pAVDesc} \cup \{\top\},\ \psi \in \text{pAVDesc} \cup \{\bot\})$

Examples

activity $\Rrightarrow$ *event* (every activity is an event)

causation $\wedge$ *activity* $\Rrightarrow \bot$

AGENT : $\top \Rrightarrow$ AGENT $\doteq$ ACTOR

activity $\Rrightarrow$ ACTOR : $\top$

activity $\wedge$ *motion* $\Rrightarrow$ ACTOR $\doteq$ MOVER

Attribute-value constraints

Constraints (general format) $\forall\phi,\ \phi \in \text{AVDesc}$

$\langle V, \mathcal{I}, g\rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g\rangle, v \vDash \phi$ for every $v \in V$

Notation:

$\phi \Rrightarrow \psi$ for $\forall(\phi \to \psi)$

Horn constraints:

$\phi_1 \wedge \ldots \wedge \phi_n \Rrightarrow \psi$ $(\phi_i \in \text{pAVDesc} \cup \{\top\},\ \psi \in \text{pAVDesc} \cup \{\bot\})$

Examples

$\textit{activity} \Rrightarrow \textit{event}$ (every activity is an event)

$\textit{causation} \wedge \textit{activity} \Rrightarrow \bot$ (there is nothing which is both a causation and an activity)

$\text{AGENT} : \top \Rrightarrow \text{AGENT} \doteq \text{ACTOR}$

$\textit{activity} \Rrightarrow \text{ACTOR} : \top$

$\textit{activity} \wedge \textit{motion} \Rrightarrow \text{ACTOR} \doteq \text{MOVER}$

Attribute-value constraints

Constraints (general format) $\forall\phi,\ \phi \in \text{AVDesc}$

$\langle V, \mathcal{I}, g\rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g\rangle, v \vDash \phi$ for every $v \in V$

Notation:

$\phi \Rrightarrow \psi$ for $\forall(\phi \rightarrow \psi)$

Horn constraints:

$\phi_1 \wedge \ldots \wedge \phi_n \Rrightarrow \psi$ $(\phi_i \in \text{pAVDesc} \cup \{\top\},\ \psi \in \text{pAVDesc} \cup \{\bot\})$

Examples

$\textit{activity} \Rrightarrow \textit{event}$	(every activity is an event)
$\textit{causation} \wedge \textit{activity} \Rrightarrow \bot$	(there is nothing which is both a causation and an activity)
$\text{AGENT} : \top \Rrightarrow \text{AGENT} \doteq \text{ACTOR}$	(every agent is also an actor)
$\textit{activity} \Rrightarrow \text{ACTOR} : \top$	
$\textit{activity} \wedge \textit{motion} \Rrightarrow \text{ACTOR} \doteq \text{MOVER}$	

Attribute-value constraints

Constraints (general format) $\forall\phi,\ \phi \in \text{AVDesc}$

$\langle V, \mathcal{I}, g\rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g\rangle, v \vDash \phi$ for every $v \in V$

Notation:

$\phi \Rrightarrow \psi$ for $\forall(\phi \rightarrow \psi)$

Horn constraints:

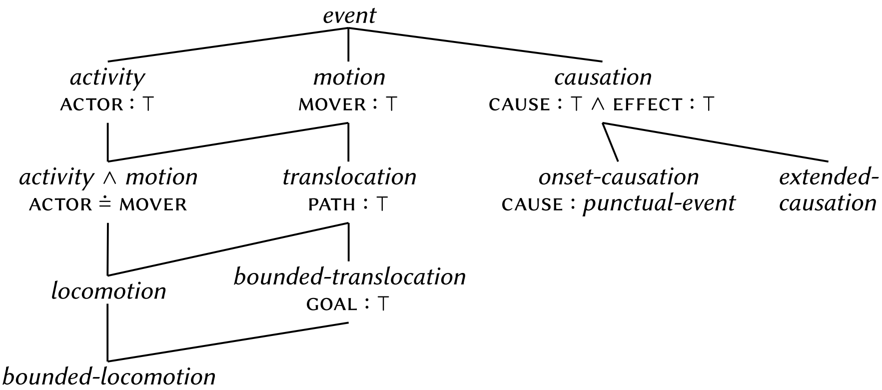
$\phi_1 \wedge \ldots \wedge \phi_n \Rrightarrow \psi \qquad (\phi_i \in \text{pAVDesc} \cup \{\top\},\ \psi \in \text{pAVDesc} \cup \{\bot\})$

Examples

$\mathit{activity} \Rrightarrow \mathit{event}$	(every activity is an event)
$\mathit{causation} \wedge \mathit{activity} \Rrightarrow \bot$	(there is nothing which is both a causation and an activity)
$\text{AGENT} : \top \Rrightarrow \text{AGENT} \doteq \text{ACTOR}$	(every agent is also an actor)
$\mathit{activity} \Rrightarrow \text{ACTOR} : \top$	(every activity has an actor)
$\mathit{activity} \wedge \mathit{motion} \Rrightarrow \text{ACTOR} \doteq \text{MOVER}$	...

Attribute-value constraints

Graphical presentation of constraints



Caveat: Reading convention required!

Attribute-value constraints

Further examples [Babonnaud et al. 2016]

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

Attribute-value constraints

Further examples

[Babonnaud et al. 2016]

book ⇛ *info-carrier*

book ● ⇝ *book*, *info-carrier* ●

Further examples [Babonnaud et al. 2016]

book $\Rightarrow$ *info-carrier*

book ● $\rightsquigarrow$ *book*, *info-carrier* ●

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

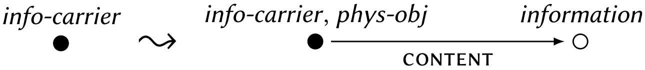
Attribute-value constraints

Further examples

[Babonnaud et al. 2016]

book ⇛ *info-carrier*

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

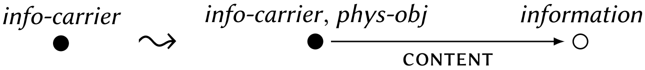


Further examples

[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*

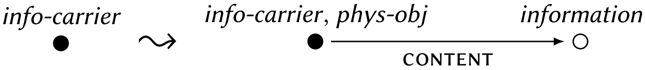
Attribute-value constraints

Further examples

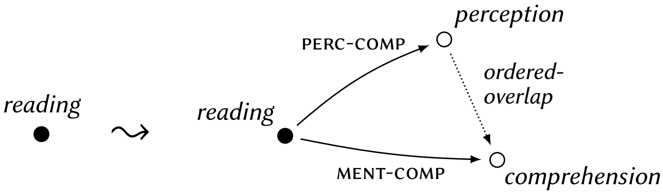
[Babonnaud et al. 2016]

$$\mathit{book} \Rrightarrow \mathit{info\text{-}carrier}$$

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



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



Unification under constraints

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.

Unification under constraints

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: $k_1 \cdot \phi_1 \wedge \ldots \wedge k_n \cdot \phi_n \rightarrow l \cdot \psi$)

Unification under constraints

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: $k_1 \cdot \phi_1 \wedge \ldots \wedge k_n \cdot \phi_n \rightarrow l \cdot \psi$)

Example

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

Unification under constraints

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: $k_1 \cdot \phi_1 \wedge \ldots \wedge k_n \cdot \phi_n \rightarrow l \cdot \psi$)

Example

$$e\begin{bmatrix} \textit{eating} & \\ \textsc{actor} & x \\ \textsc{theme} & y \end{bmatrix} \sqcup \quad u\begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{`Adam'} \end{bmatrix} \sqcup \quad x \triangleq u$$

Unification under constraints

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: $k_1 \cdot \phi_1 \wedge \ldots \wedge k_n \cdot \phi_n \rightarrow l \cdot \psi$)

Example

$$e\begin{bmatrix} \textit{eating} & \\ \textsc{actor} & x \\ \textsc{theme} & y \end{bmatrix} \sqcup u\begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{'Adam'} \end{bmatrix} \sqcup x \triangleq u = e\begin{bmatrix} \textit{eating} & \\ \textsc{actor} & \begin{matrix} x \\ u \end{matrix}\begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{'Adam'} \end{bmatrix} \\ \textsc{theme} & y \end{bmatrix}$$

Unification under constraints

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: $k_1 \cdot \phi_1 \wedge \ldots \wedge k_n \cdot \phi_n \rightarrow l \cdot \psi$)

Example

$$e\begin{bmatrix} \textit{eating} & \\ \text{ACTOR} & x \\ \text{THEME} & y \end{bmatrix} \sqcup u\begin{bmatrix} \textit{person} & \\ \text{NAME} & \text{`Adam'} \end{bmatrix} \sqcup x \triangleq u = e\begin{bmatrix} \textit{eating} & \\ \text{ACTOR} & \begin{matrix} x \\ u \end{matrix}\begin{bmatrix} \textit{person} & \\ \text{NAME} & \text{`Adam'} \end{bmatrix} \\ \text{THEME} & y \end{bmatrix}$$

A general view on semantic processing

Semantic processing as the **incremental construction** of **minimal** (**frame**) **models** (by unification under constraints) based on the input, the context, and background knowledge (lexicon, . . .).

Outline of today's course

1. Introduction to frame semantics
   - Frames in the sense of Fillmore and Barsalou
   - Frames according to this course
2. Formalization of frames
   - Attribute-value descriptions and formulas
   - Formal definition of frames
   - Frames as models
   - Subsumption and unification
   - Attribute-value constraints
3. Further topics
   - Frames versus feature structures
   - Type constraints versus type hierarchy
4. Frame semantics: extensions
5. Summary and outlook

Frames versus feature structures

- **Feature structures** have a designated **root** node from which each other node is reachable via an attribute path, and they have no relations.

Frames versus feature structures

- **Feature structures** have a designated **root** node from which each other node is reachable via an attribute path, and they have no relations. $\rightsquigarrow$ $|\mathrm{Nvar}| = 1$, $\mathrm{Rel} = \emptyset$.

Frames versus feature structures

- **Feature structures** have a designated **root** node from which each other node is reachable via an attribute path, and they have no relations. $\rightsquigarrow$ $|\mathrm{Nvar}| = 1$, $\mathrm{Rel} = \emptyset$.

- **Typed feature structures**

Frames versus feature structures

- **Feature structures** have a designated **root** node from which each other node is reachable via an attribute path, and they have no relations. $\leadsto$ $|\text{Nvar}| = 1$, $\text{Rel} = \emptyset$.

- **Typed feature structures**

  $\text{Nname} = \emptyset$

Frames versus feature structures

- **Feature structures** have a designated **root** node from which each other node is reachable via an attribute path, and they have no relations. $\leadsto$ $|\mathrm{Nvar}| = 1$, $\mathrm{Rel} = \varnothing$.
- **Typed feature structures**

  $\mathrm{Nname} = \varnothing$
- **Untyped feature structures**

Frames versus feature structures

- **Feature structures** have a designated **root** node from which each other node is reachable via an attribute path, and they have no relations. $\rightsquigarrow$ $|\text{Nvar}| = 1$, $\text{Rel} = \emptyset$.

- **Typed feature structures**

  $\text{Nname} = \emptyset$

- **Untyped feature structures**

  $\text{Type} = \emptyset$; named nodes have no attributes.

Type constraints versus type hierarchy

Type constraints

(Horn) constraints consisting only of type symbols (and $\top$ and $\bot$)

Type constraints versus type hierarchy

Type constraints

(Horn) constraints consisting only of type symbols (and $\top$ and $\bot$)

Type hierarchy generated by type constraints

$\approx$ single node models which satisfy all constraints, ordered by (inverse) subsumption

Type constraints versus type hierarchy

Type constraints

(Horn) constraints consisting only of type symbols (and $\top$ and $\bot$)

Type hierarchy generated by type constraints

$\approx$ single node models which satisfy all constraints, ordered by (inverse) subsumption

Example

activity $\Rrightarrow$ *event*

motion $\Rrightarrow$ *event*

locomotion $\Rrightarrow$ *activity*

locomotion $\Rrightarrow$ *motion*

Type constraints versus type hierarchy

Type constraints

(Horn) constraints consisting only of type symbols (and $\top$ and $\bot$)

Type hierarchy generated by type constraints

$\approx$ single node models which satisfy all constraints, ordered by (inverse) subsumption

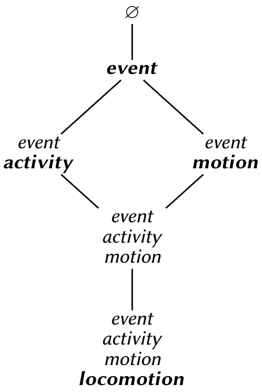
Example

activity $\Rrightarrow$ *event*

motion $\Rrightarrow$ *event*

locomotion $\Rrightarrow$ *activity*

locomotion $\Rrightarrow$ *motion*



Outline of today's course

Frame semantics: extensions

Issue How to represent quantification, negation, intensionality, etc. within frame semantics?

Frame semantics: extensions

Issue How to represent quantification, negation, intensionality, etc. within frame semantics?

(4) a. Every man kissed some woman.
  b. The king of France is not bald.
  c. Adam thinks he has understood what frame semantics is about.

Frame semantics: extensions

Issue How to represent quantification, negation, intensionality, etc. within frame semantics?

(4) a. Every man kissed some woman.
    b. The king of France is not bald.
    c. Adam thinks he has understood what frame semantics is about.

Possible approaches

1. Use an **attribute-value language with quantifiers** and build formulas instead of models. [≈ Kallmeyer/Osswald/Pogodalla 2016]

Frame semantics: extensions

Issue How to represent quantification, negation, intensionality, etc. within frame semantics?

(4) a. Every man kissed some woman.
b. The king of France is not bald.
c. Adam thinks he has understood what frame semantics is about.

Possible approaches

1. Use an **attribute-value language with quantifiers** and build formulas instead of models. [≈ Kallmeyer/Osswald/Pogodalla 2016]
2. Keep frames as basic semantic representations and evaluate **quantification over the domain of frames**. [≈ Muskens 2013]

Frame semantics: extensions

Issue How to represent quantification, negation, intensionality, etc. within frame semantics?

(4) a. Every man kissed some woman.
b. The king of France is not bald.
c. Adam thinks he has understood what frame semantics is about.

Possible approaches

1. Use an **attribute-value language with quantifiers** and build formulas instead of models. [≈ 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)

$\forall\phi$, $\exists\phi$ ($\phi \in$ AVDesc)

Frame semantics: extensions

Attribute-value formulas with quantifiers (qAVForm)

$\forall\phi$, $\exists\phi$ ($\phi \in$ AVDesc)

$\langle V, \mathcal{I}, g \rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g \rangle, v \vDash \phi$ for every $v \in V$

Frame semantics: extensions

Attribute-value formulas with quantifiers (qAVForm)

$\forall\phi$, $\exists\phi$ ($\phi \in$ AVDesc)

$\langle V, \mathcal{I}, g\rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g\rangle, v \vDash \phi$ for every $v \in V$

$\langle V, \mathcal{I}, g\rangle \vDash \exists\phi$ iff $\langle V, \mathcal{I}, g\rangle, v \vDash \phi$ for some $v \in V$

Frame semantics: extensions

Attribute-value formulas with quantifiers (qAVForm)

$\forall\phi,\ \exists\phi\ \ (\phi \in \text{AVDesc})$ $\qquad$ $\boldsymbol{\forall} x\, \alpha,\ \boldsymbol{\exists} x\, \alpha\ \ (\alpha \in \text{AVForm} \cup \text{qAVForm})$

$\langle V, \mathcal{I}, g \rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g \rangle, v \vDash \phi$ for every $v \in V$

$\langle V, \mathcal{I}, g \rangle \vDash \exists\phi$ iff $\langle V, \mathcal{I}, g \rangle, v \vDash \phi$ for some $v \in V$

Attribute-value formulas with quantifiers (qAVForm)

$\forall\phi$, $\exists\phi$ ($\phi \in$ AVDesc) $\quad$ $\boldsymbol{\forall} x\,\alpha$, $\boldsymbol{\exists} x\,\alpha$ ($\alpha \in$ AVForm $\cup$ qAVForm)

$\langle V, \mathcal{I}, g\rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g\rangle, v \vDash \phi$ for every $v \in V$

$\langle V, \mathcal{I}, g\rangle \vDash \exists\phi$ iff $\langle V, \mathcal{I}, g\rangle, v \vDash \phi$ for some $v \in V$

For $x \notin \text{dom}(g)$ (= set of variables for which g is defined):

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

Frame semantics: extensions

Attribute-value formulas with quantifiers (qAVForm)

$\forall\phi,\ \exists\phi\ \ (\phi \in \text{AVDesc})$ $\qquad$ $\boldsymbol{\forall} x\,\alpha,\ \boldsymbol{\exists} x\,\alpha\ \ (\alpha \in \text{AVForm} \cup \text{qAVForm})$

$\langle V, \mathcal{I}, g\rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g\rangle, v \vDash \phi$ for every $v \in V$

$\langle V, \mathcal{I}, g\rangle \vDash \exists\phi$ iff $\langle V, \mathcal{I}, g\rangle, v \vDash \phi$ for some $v \in V$

For $x \notin \text{dom}(g)$ (= set of variables for which g is defined):

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

$\langle V, \mathcal{I}, g\rangle \vDash \boldsymbol{\exists} x\,\alpha$ iff $\langle V, \mathcal{I}, g'\rangle \vDash \alpha$ for some assignment g' with …

Attribute-value formulas with quantifiers (qAVForm)

$\forall\phi,\ \exists\phi\ \ (\phi \in \text{AVDesc})\qquad \boldsymbol{\forall} x\,\alpha,\ \boldsymbol{\exists} x\,\alpha\ \ (\alpha \in \text{AVForm} \cup \text{qAVForm})$

$\langle V, \mathcal{I}, g \rangle \vDash \forall\phi$ iff $\langle V, \mathcal{I}, g \rangle, v \vDash \phi$ for every $v \in V$

$\langle V, \mathcal{I}, g \rangle \vDash \exists\phi$ iff $\langle V, \mathcal{I}, g \rangle, v \vDash \phi$ for some $v \in V$

For $x \notin \text{dom}(g)$ (= set of variables for which g is defined):

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

$\langle V, \mathcal{I}, g \rangle \vDash \boldsymbol{\exists} x\,\alpha$ iff $\langle V, \mathcal{I}, g' \rangle \vDash \alpha$ for some assignment g' with …

Note: $\forall\phi\ \equiv\ \boldsymbol{\forall} x(x \cdot \phi),\quad \exists\phi\ \equiv\ \boldsymbol{\exists} x(x \cdot \phi)$ (with x not occurring in ϕ)

Frame semantics: extensions

Examples

(5) Every dog barked.

Examples

(5) Every dog barked.

$\forall x(x \cdot \mathit{dog} \rightarrow \exists(\mathit{barking} \wedge \textsc{agent} \triangleq x))$

Frame semantics: extensions

Examples

(5) Every dog barked.

$\forall x(x \cdot dog \rightarrow \exists(barking \wedge \textsc{agent} \triangleq x))$

corresponding first-order formula: $\forall x(dog(x) \rightarrow \exists e(barking(e) \wedge \textsc{agent}(e, x)))$

Frame semantics: extensions

Examples

(5) Every dog barked.

$\forall x(x \cdot dog \rightarrow \exists(barking \wedge \textsc{agent} \triangleq x))$

corresponding first-order formula: $\forall x(dog(x) \rightarrow \exists e(barking(e) \wedge \textsc{agent}(e, x)))$

(6) Every man kissed some woman.

Examples

(5) Every dog barked.

$\forall x(x \cdot \mathit{dog} \rightarrow \exists(\mathit{barking} \wedge \textsc{agent} \triangleq x))$

corresponding first-order formula: $\forall x(\mathit{dog}(x) \rightarrow \exists e(\mathit{barking}(e) \wedge \textsc{agent}(e, x)))$

(6) Every man kissed some woman.

$\forall x(x \cdot \mathit{man} \rightarrow \exists y(y \cdot \mathit{woman} \wedge \exists(\mathit{kissing} \wedge \textsc{agent} \triangleq x \wedge \textsc{theme} \triangleq y))$

Examples

(5) Every dog barked.

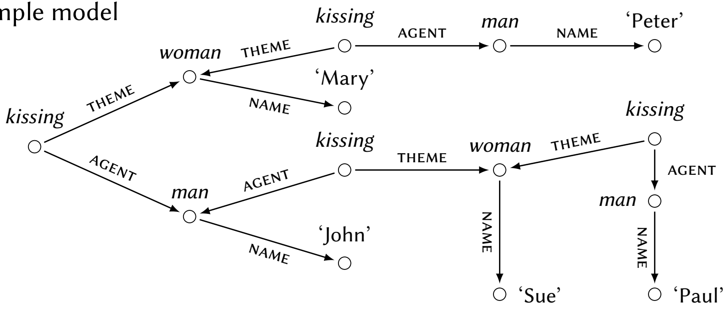
$\forall x(x \cdot dog \rightarrow \exists(barking \wedge \textsc{agent} \triangleq x))$

corresponding first-order formula: $\forall x(dog(x) \rightarrow \exists e(barking(e) \wedge \textsc{agent}(e, x)))$

(6) Every man kissed some woman.

$\forall x(x \cdot man \rightarrow \exists y(y \cdot woman \wedge \exists(kissing \wedge \textsc{agent} \triangleq x \wedge \textsc{theme} \triangleq y))$

Example model



Frame semantics: extensions

Examples

(7) Every man walked into some house.

$$\forall x(x \cdot \mathit{man} \;\rightarrow\; \exists z(z \cdot \mathit{house} \;\wedge\; \exists(\mathit{locomotion} \;\wedge\; \textsc{manner} : \mathit{walking}$$
$$\wedge\; \textsc{actor} \triangleq x \;\wedge\; \textsc{mover} \doteq \textsc{actor}$$
$$\wedge\; \textsc{goal} \triangleq z \;\wedge\; \textsc{path} : (\mathit{path} \;\wedge\; \textsc{endp} : \mathit{region})$$
$$\wedge\; [\textsc{path endp},\; \textsc{goal in-region}] : \mathit{part\text{-}of})))$$

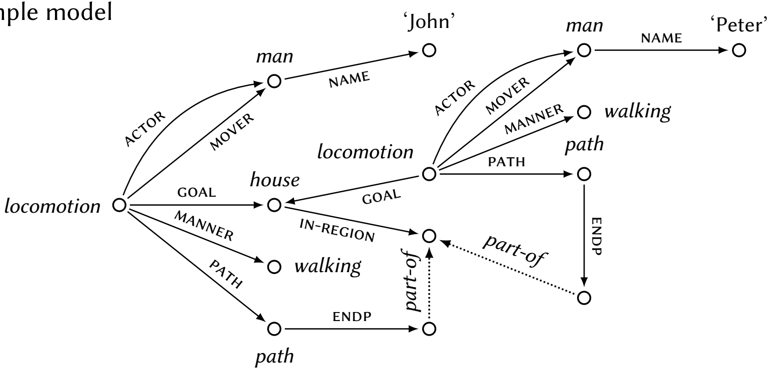
Frame semantics: extensions

Examples

(7) Every man walked into some house.

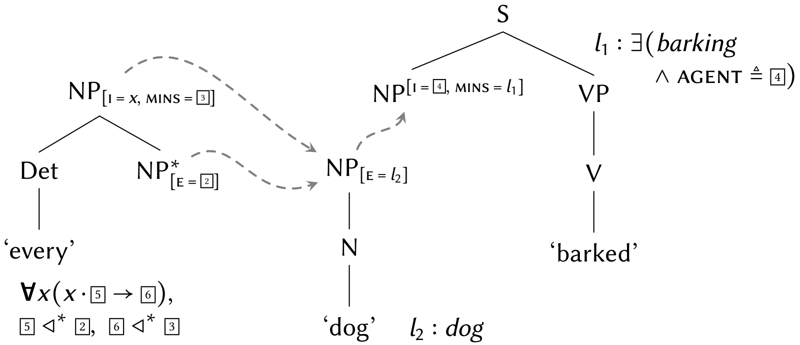
$$\forall x(x \cdot man \;\rightarrow\; \exists z(z \cdot house \;\wedge\; \exists(locomotion \;\wedge\; \textsc{manner} : walking$$
$$\wedge\; \textsc{actor} \triangleq x \;\wedge\; \textsc{mover} \doteq \textsc{actor}$$
$$\wedge\; \textsc{goal} \triangleq z \;\wedge\; \textsc{path} : (path \;\wedge\; \textsc{endp} : region)$$
$$\wedge\; [\textsc{path endp},\; \textsc{goal in-region}] : part\text{-}of)))$$

Example model



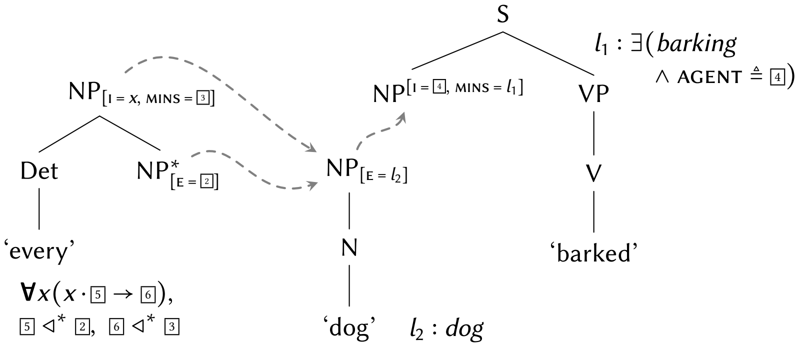
AV logic with quantifiers + underspecification ("hole semantics")

(8) Every dog barked.



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

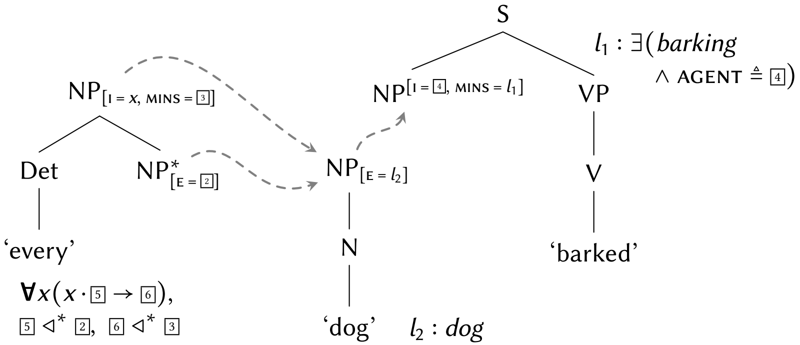
(8) Every dog barked.



$\rightsquigarrow$ $\forall x(x \cdot \boxed{5} \rightarrow \boxed{6})$, $l_2 : dog$, $l_1 : \exists(barking \wedge \textsc{agent} \triangleq x)$, $\boxed{5} \lhd^* l_2$, $\boxed{6} \lhd^* l_1$

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

(8) Every dog barked.



$\rightsquigarrow\ \forall x(x \cdot \boxed{5} \rightarrow \boxed{6}),\ l_2 : dog,\ l_1 : \exists(barking \wedge \textsc{agent} \triangleq x),\ \boxed{5} \lhd^* l_2,\ \boxed{6} \lhd^* l_1$

$\rightsquigarrow\ \forall x(x \cdot dog \rightarrow \exists(barking \wedge \textsc{agent} \triangleq x))$

Outline of today's course

Summary & outlook

Summary

- Motivation and background to frame semantics
- Attribute-value logic as a tailored logic for specifying frames
- Frames as minimal models of attribute-value formulas
- Possible ways to express quantification in frames semantics

Summary & outlook

Summary

- Motivation and background to frame semantics
- Attribute-value logic as a tailored logic for specifying frames
- Frames as minimal models of attribute-value formulas
- Possible ways to express quantification in frames semantics

Tomorrow

- Combining LTAG with frame semantics
- Elementary constructions as elementary trees with semantic frames
- Linguistic applications
- Looking ahead to the factorization of elementary constructions in the metagrammar.

References

Babonnaud, William, Laura Kallmeyer & Rainer Osswald. 2016. Polysemy and coercion – a frame-based approach using LTAG and Hybrid Logic. In Maxime Amblard, Philippe de Groote, Sylvain Pogodalla & Christian Retoré (eds.), **Logical Aspects of Computational Linguistics, 9th International Conference** Lecture Notes in Artificial Intelligence 10054, 18–33. Berlin: Springer.

Barsalou, Lawrence W. 1992. Frames, concepts, and conceptual fields. In Adrienne Lehrer & Eva Feder Kittay (eds.), **Frames, fields, and contrasts**, 21–74. Hillsdale, NJ: Lawrence Erlbaum Associates.

Carpenter, Bob. 1992. **The logic of typed feature structures** Cambridge Tracts in Theoretical Computer Science 32. Cambridge: Cambridge University Press.

Hegner, Stephen J. 1994. Properties of Horn clauses in feature-structure logic. In C. J. Rupp, Michael A. Rosner & Rod L. Johnson (eds.), **Constraints, language and computation**, 111–147. San Diego, CA: Academic Press.

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

Kallmeyer, Laura, Rainer Osswald & Sylvain Pogodalla. 2016. *For*-adverbials and aspectual interpretation: An LTAG analysis using hybrid logic and frame semantics. In Christopher Piñón (ed.), **Empirical issues in syntax and semantics EISS**, vol. 11, .

Muskens, Rheinard. 2013. Data semantics and linguistic semantics. In Maria Aloni, Michael Franke & Floris Roelofsen (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, Rainer. 1999. Semantics for attribute-value theories. In Paul Dekker (ed.), **Proceedings of the twelfth amsterdam colloquium**, 199–204. Amsterdam: University of Amsterdam, ILLC.

References (cont.)

Osswald, Rainer & Robert D. Van Valin, Jr. 2014. FrameNet, frame structure, and the syntax-semantics interface. In Thomas Gamerschlag, Doris Gerland, Rainer Osswald & Wiebke Petersen (eds.), **Frames and concept types** (Studies in Linguistics and Philosophy 94), 125–156. Dordrecht: Springer.

Rappaport Hovav, Malka & Beth Levin. 1998. Building verb meanings. In Miriam Butt & Wilhelm Geuder (eds.), **The projection of arguments: Lexical and compositional factors**, 97–134. Stanford, CA: CSLI Publications.

Rounds, William C. 1997. Feature logics. In Johan van Benthem & Alice ter Meulen (eds.), **Handbook of logic and language**, 475–533. Amsterdam: North-Holland.