Semantic Modeling with Frames

Rainer Osswald & Wiebke Petersen

Department of Linguistics and Information Science
Heinrich-Heine-Universität Düsseldorf

ESSLLI 2018

Introductory Course

Sofia University

06. 08. – 10. 08. 2018

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

SFB 991

Part 4

Formal foundations: extensions

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?

(1) 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?

(1) 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?

(1) 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?

(1) 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)

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

Frame semantics: extensions

Attribute-value formulas with quantifiers (qAVForm)

$\forall\phi, \ \exists\phi \quad (\phi \in \text{AVDesc})$

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

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

Frame semantics: extensions

Attribute-value formulas with quantifiers (qAVForm)

$\forall\phi$, $\exists\phi$ ($\phi \in$ AVDesc) $\boldsymbol{\forall} x\, \alpha$, $\boldsymbol{\exists} x\, \alpha$ ($\alpha \in$ 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 $v \in V$

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 $v \in \text{dom}(g)$

Frame semantics: extensions

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 $x \notin \mathrm{dom}(g)$ (= 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
$\mathrm{dom}(g') = \mathrm{dom}(g) \cup \{x\}$ and
$g(v) = g'(v)$ for all $v \in \mathrm{dom}(g)$

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

Frame semantics: extensions

Attribute-value formulas with quantifiers (qAVForm)

$\forall\phi$, $\exists\phi$ ($\phi \in$ AVDesc) $\qquad$ $\boldsymbol{\forall} x\, \alpha$, $\boldsymbol{\exists} x\, \alpha$ ($\alpha \in$ 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 $x \notin \mathrm{dom}(g)$ (= 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 $\mathrm{dom}(g') = \mathrm{dom}(g) \cup \{x\}$ and $g(v) = g'(v)$ for all $v \in \mathrm{dom}(g)$

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

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

Frame semantics: extensions

Examples

(2) Every dog barked.

Examples

(2) Every dog barked.

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

Frame semantics: extensions

Examples

(2) Every dog barked.

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

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

Examples

(2) Every dog barked.

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

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

(3) Every man kissed some woman.

Frame semantics: extensions

Examples

(2) Every dog barked.

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

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

(3) Every man kissed some woman.

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

Frame semantics: extensions

Examples

(2) Every dog barked.

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

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

(3) Every man kissed some woman.

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

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

Frame semantics: extensions

Examples

(2) Every dog barked.

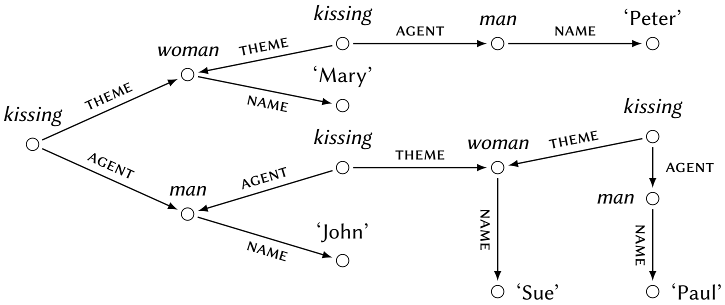
$\forall x(x \cdot \textit{dog} \rightarrow \exists(\textit{barking} \wedge \textsc{agent} \triangleq x))$

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

(3) Every man kissed some woman.

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

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



Frame semantics: extensions

Examples

(4) Every man walked into some house.

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

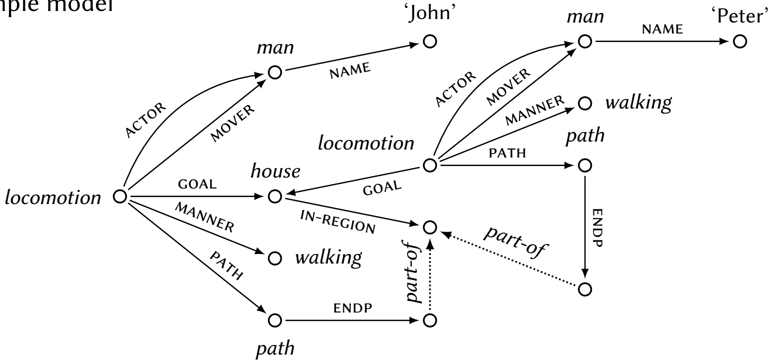
Frame semantics: extensions

Examples

(4) 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

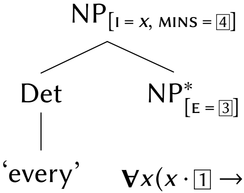


Frame semantics: extensions

An alternative approach to the syntax-semantics interface:

- elementary trees + frames $\rightsquigarrow$
  elementary trees + underspecified AV formulas with
  scope constraints
- unification of frames $\rightsquigarrow$ construction of AV formulas

Example



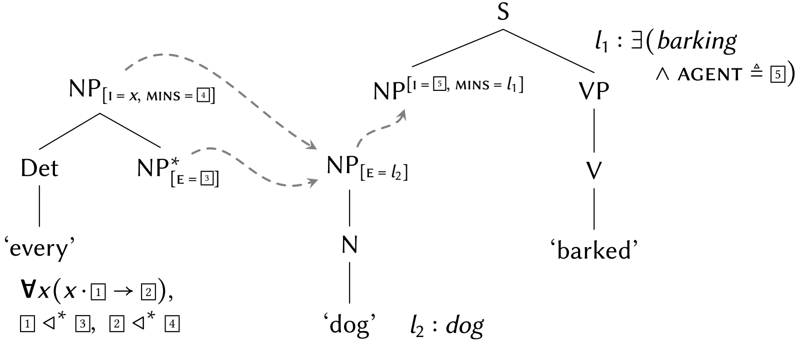
$\forall x(x \cdot \boxed{1} \rightarrow \boxed{2})$,
$\boxed{1} \lhd^* \boxed{3}$, $\boxed{2} \lhd^* \boxed{4}$

($h \lhd^* l$ means that expression l is a **subexpression** of h.)

Frame semantics: extensions

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

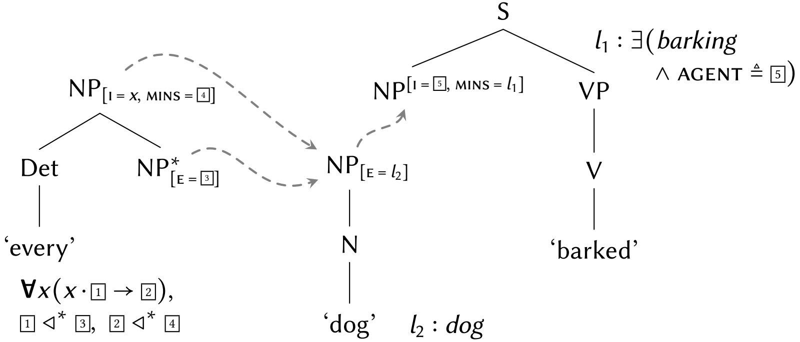
(5) Every dog barked.



Frame semantics: extensions

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

(5) Every dog barked.

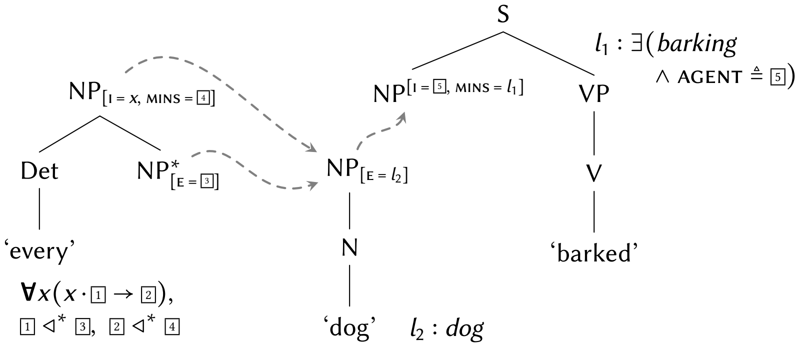


$\rightsquigarrow\ \forall x(x \cdot \boxed{1} \rightarrow \boxed{2}),\ l_2 : dog,\ l_1 : \exists(barking \wedge \text{AGENT} \triangleq x),\ \boxed{1} \triangleleft^{*} l_2,\ \boxed{2} \triangleleft^{*} l_1$

Frame semantics: extensions

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

(5) Every dog barked.

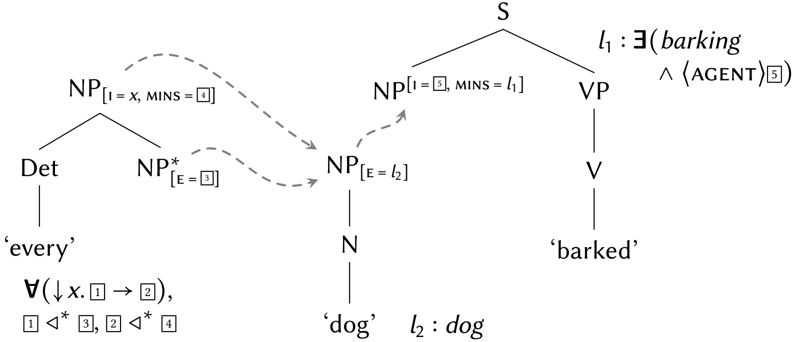


$\rightsquigarrow$ $\forall x(x \cdot [1] \rightarrow [2])$, $l_2 : dog$, $l_1 : \exists(barking \wedge \text{AGENT} \triangleq x)$, $[1] \triangleleft^* l_2$, $[2] \triangleleft^* l_1$

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

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

(6) Every dog barked.



$\rightsquigarrow\ \forall(\downarrow x.\, \boxed{1} \rightarrow \boxed{2}),\ l_2 : dog,\ l_1 : \exists(barking \wedge \langle \textsc{agent} \rangle x),\ \boxed{1} \triangleleft^* l_2, \boxed{2} \triangleleft^* l_1$

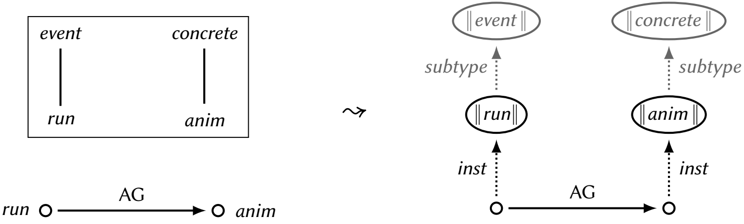
$\rightsquigarrow\ \forall(\downarrow x.\, dog \rightarrow \exists(barking \wedge \langle \textsc{agent} \rangle x))$

[from Kallmeyer/Osswald/Pogodalla 2016]

Frame types (sketch)

Frame types (sketch)

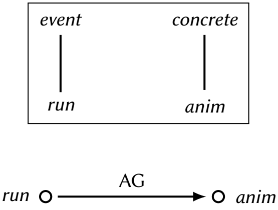
Types as elements of the universe/model



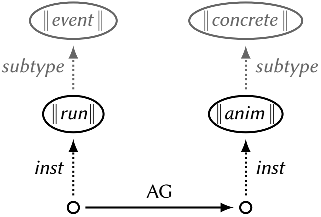
‖*event*‖, ‖*run*‖, etc.: type names (nominals)

Frame types (sketch)

Types as elements of the universe/model



$\rightsquigarrow$

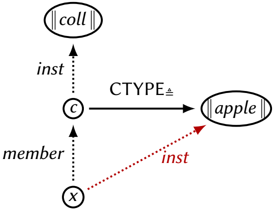


$\|event\|$, $\|run\|$, etc.: type names (nominals)

Types as values of attributes

Example: collections of elements of type T

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



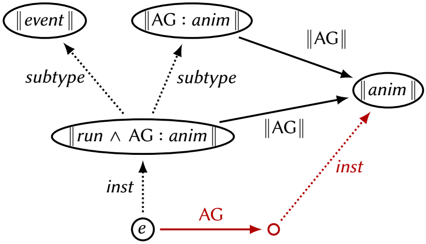
Frame types (sketch)

Complex frame types

Introduce **frame types** like $\|\textsc{p} : t\|$

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

$n \ \textit{inst} \ \|\textsc{p} : t\| \leftrightarrow n \cdot \textsc{p} \ \textit{inst} \ \|t\|$

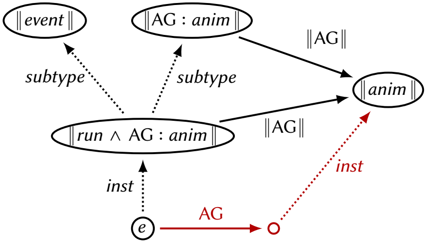


Complex frame types

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

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

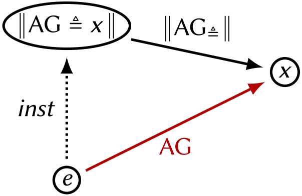
n *inst* $\|\text{P} : t\,\| \leftrightarrow n \cdot \text{P}$ *inst* $\|t\,\|$



Dependent frame types

n *inst* $\|\text{P} \triangleq x\| \leftrightarrow n \cdot \text{P} \triangleq x$

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

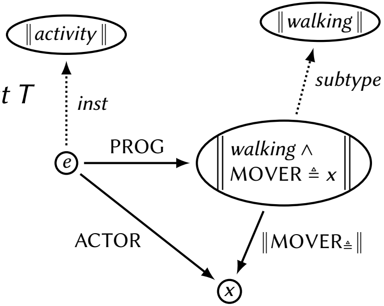


Frame types (sketch)

Example event progression

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

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

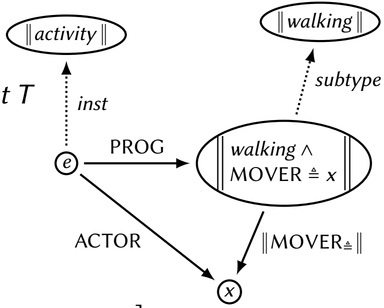


Frame types (sketch)

Example event progression

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

$$e\begin{bmatrix} \textit{activity} & \\ \text{ACTOR} & x \\ \text{PROG} & \left\| \begin{bmatrix} \textit{walking} & \\ \text{MOVER} & x \end{bmatrix} \right\| \end{bmatrix}$$



Example scalar change

$$\begin{bmatrix} \textit{progression} & \\ \text{ENTITY} & x \\ \text{PROG} & \left\| \begin{array}{l} \begin{bmatrix} \textit{incremental-change} & \\ \text{ENTITY} & x \\ \text{INITIAL} & \begin{bmatrix} \textit{stage} & \\ \text{ENTITY} & x \\ \text{LENGTH} & \boxed{0} \end{bmatrix} \\ \text{FINAL} & \begin{bmatrix} \textit{stage} & \\ \text{ENTITY} & x \\ \text{LENGTH} & \boxed{1} \end{bmatrix} \end{bmatrix} \\ \textit{lesser}(\boxed{1}, \boxed{0}) \end{array} \right\| \end{bmatrix}$$

Frame types (sketch)

Application atelic and telic interpretation of degree achievements

szárad (‘dry’)

$$
\begin{bmatrix}
\textit{progression} & \\
\textsc{entity} & x \\
\textsc{prog} & \left\| \begin{array}{l}
\begin{bmatrix}
\textit{incremental-change} & \\
\textsc{entity} & x \\
\textsc{init} & \begin{bmatrix} \textit{stage} & \\ \textsc{entity} & x \\ \textsc{moisture} & \boxed{1} \end{bmatrix} \\
\textsc{final} & \begin{bmatrix} \textit{stage} & \\ \textsc{entity} & x \\ \textsc{moisture} & \boxed{2} \end{bmatrix}
\end{bmatrix} \\
\textit{lesser}(\boxed{2}, \boxed{1})
\end{array} \right\|
\end{bmatrix}
$$

meg-szárad (‘dry’)

$$
\begin{bmatrix}
\textit{bounded-event} & \\
\textsc{entity} & x \\
\textsc{prog} & \left\| \begin{array}{l}
\begin{bmatrix}
\textit{incremental-change} & \\
\textsc{entity} & x \\
\textsc{init} & \begin{bmatrix} \textit{stage} & \\ \textsc{entity} & x \\ \textsc{moisture} & \boxed{1} \end{bmatrix} \\
\textsc{final} & \begin{bmatrix} \textit{stage} & \\ \textsc{entity} & x \\ \textsc{moisture} & \boxed{2} \end{bmatrix}
\end{bmatrix} \\
\textit{lesser}(\boxed{2}, \boxed{1})
\end{array} \right\| \\
\textsc{final} & \begin{bmatrix} \textit{stage} & \\ \textsc{entity} & x \\ \textsc{moisture} & \textit{zero} \end{bmatrix}
\end{bmatrix}
$$

$$\textsc{final} : \textit{stage} \;\wedge\; \textsc{prog}\ \|\textsc{final}\| \triangleq T \;\Rrightarrow\; \textsc{final}\ \textit{inst}\ T$$