Frames – Structures, Types, and Constraints

Laura Kallmeyer

SFB 991, Heinrich-Heine-Universität Düsseldorf

SFB-Kolloquium, 29.10.2015

SFB 991

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Introduction

- Frames allow for a fine-grained representation of meaning, not only at the lexical level.
- Frames allow to capture generalizations via types.
- Integrating logical operators into frames allows us to combine these advantages with standard approaches to logical form.
- We can use a logic in order to formulate constraints on frames.
- We can use standard underspecification techniques on this logic in order to abstract over several readings.

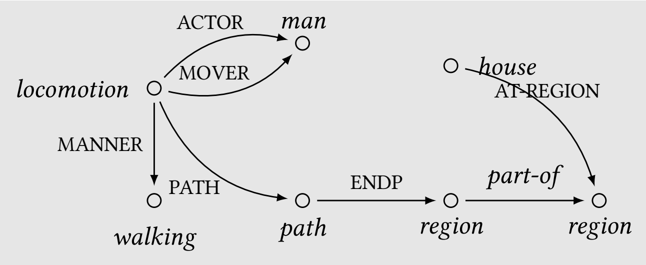
The claim this talk wants to make is the following:

> The combination of a) frame semantics with its system of types, b) a constraint language for frames that includes quantification, and c) underspecification of the constraints allows us to capture semantic generalizations in an elegant way.

Table of contents

Düsseldorf Frames

- Frames are a representation format of conceptual and lexical knowledge.
- They are commonly presented as semantic graphs with labelled nodes and edges where nodes correspond to entities (individuals, events, …) and edges to (functional or non-functional) relations between these entities.

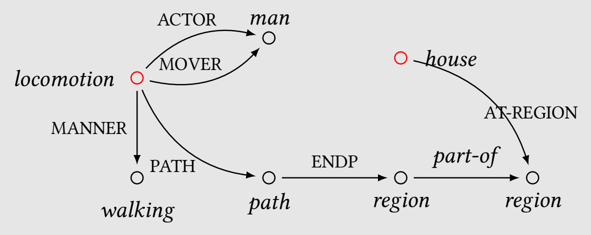


Düsseldorf Frames

There are different ways in which the CRC projects focus on frames. However, there is a consensus of what type of structures and representations we mean when using the term *frame*.

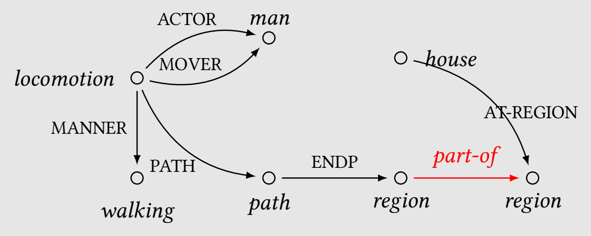
Düsseldorf frames can be formalized as an extended form of typed features structures (Petersen, 2007; Kallmeyer & Osswald, 2013; Petersen & Osswald, 2014). They differ from standard feature structures in some aspects:

a) There need not be a unique root node, i. e., a single node from which all other nodes are reachable via attribute-value paths.



Düsseldorf Frames

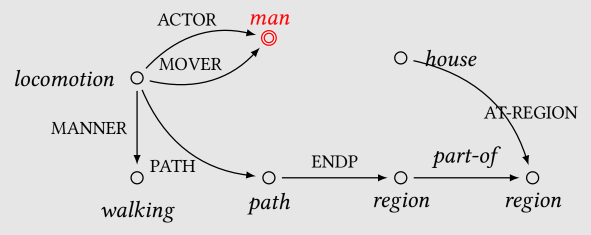
b) We can have relations between nodes, not only functional attributes.



(see also the *comparators* in Löbner, 2015)

Düsseldorf Frames

c) We can focus on internal nodes of a frame, i. e., the frame is not only accessible via its source nodes.[1]



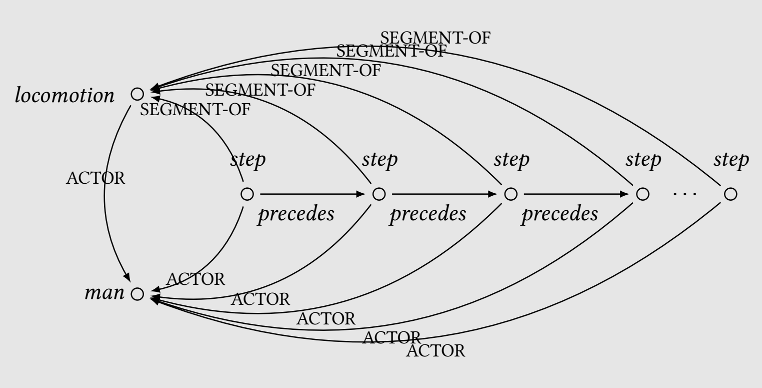
"the/a man who walked to the house"

[1]Source node = node without an incoming edge.

Düsseldorf Frames

Beyond this, the projects differ in the way they use frames.

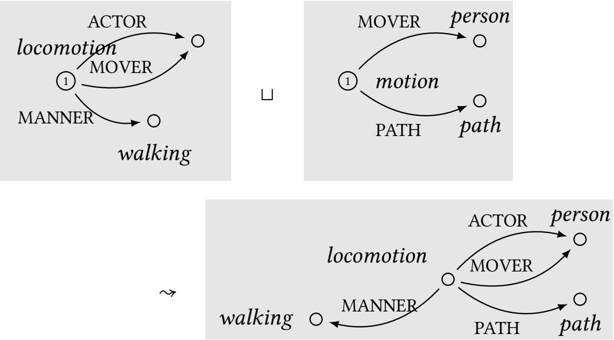
1. **Form of the frames**: What is a node in a frame and what is an attribute? What is implicitely given by a certain type and what do I spell out in the frame structure?– Different possibilities depending on the perspective one has on something.



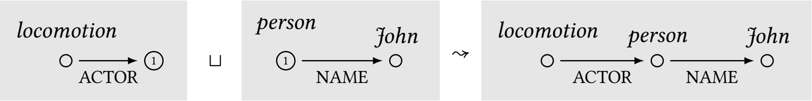
Düsseldorf Frames

2. **Operations on frames**: *Unification* is an important means for us to combine frames.

a) identifications of root nodes:



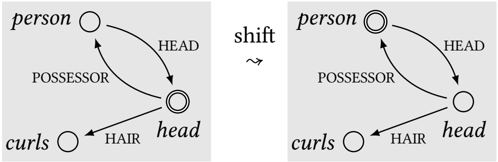
b) identifications of other nodes:



Düsseldorf Frames

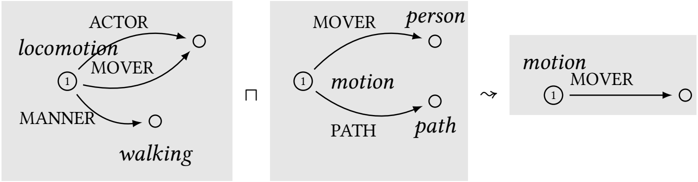
Other operations that play a role in the CRC:

Shifts:



"Lockenkopf"

Overlap/generalization:



and *zooming*.

Logical operators and frames

Question: How can we integrate quantification and negation into frames?

This is important not only for standard quantifier examples such as

(1) every man loves a woman

but also for cases where quantification and event semantics interact in interesting ways. Here, we can combine the strengths of frame semantics with an approach to logical form.

Rest of the talk:

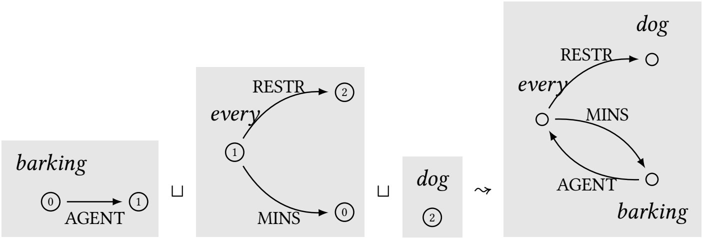
1. Overview: Different strategies for integrating logical operators into frames.
2. One proposal: Combining hybrid logic and frame semantics.
3. Case study: progression and iteration in event semantics; the case of *for*-adverbials.

Logical operators and frames

One possibility is to **include the logical operators into the frame structure**.

(2) every dog barked

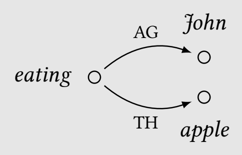
Frames according to Kallmeyer & Richter (2014):



- Requires an additional mechanism to read off truth conditions from frames.
- Mixes aspects of meaning with aspects of logical form syntax.

Logical operators and frames

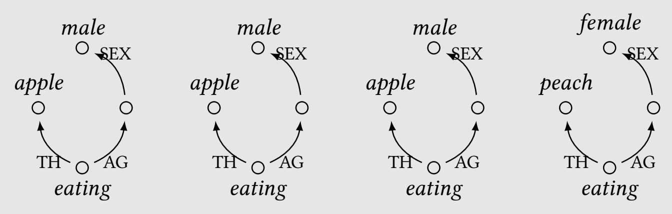
Other possibility: **Separate logical operators from frames by using a logic to talk about frames**. This logic allows for negation and quantification.
Approach pursued by Muskens (2013).



(3) a. John ate an apple

b. *eating e* ∘ AG *e x* ∘ *John x* ∘ TH *e y* ∘ *apple y*

Logical operators and frames



(4) a. every man ate an apple

b. $\forall x[\exists z[\mathit{person}\ x \circ \mathrm{SEX}\ x\ z\ \circ \mathit{male}\ z]w_0 \rightarrow$
$\exists ye[\mathit{apple}\ y \circ \mathit{eating}\ e \circ \mathrm{AG}\ e\ x \circ \mathrm{TH}\ e\ y]w_0]$

(wide scope of universal)

Hybrid logic for frames

Kallmeyer et al. (2015a,b) follows Muskens (2013) by keeping frames (= our models) and logical operators (= part of the logic) separate.

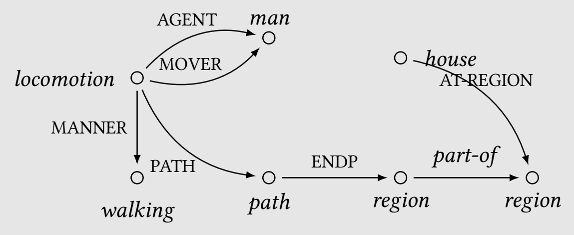
Question: Which logic to choose in order to talk about frames?

Our proposal: extended version of *modal logic* (Blackburn et al., 2007)

- Modal logic has been proposed as a logic for feature structures (Blackburn, 1993).
- It supports the local perspective on graphs that we adopt when talking about frames: Formulas are evaluated in a specific node.
- Extensions of modal logic allow to incorporate the logical operators we need. This leads to *hybrid logic* (HL, Areces & ten Cate, 2007)

Hybrid logic for frames

Model $\mathcal{M}_1$:



- *region* is true in the two nodes on the right at the bottom.
- $\langle$AGENT$\rangle$*man* is true at the *locomotion* node.
- *locomotion* $\wedge$ $\langle$MANNER$\rangle$*walking* $\wedge$ $\langle$PATH$\rangle\langle$ENDP$\rangle\top$ is also true at the *locomotion* node.

HL extends this with

- the possibility to name nodes in order to go back to them without following a specific path;
- quantification over nodes.

Hybrid logic for frames

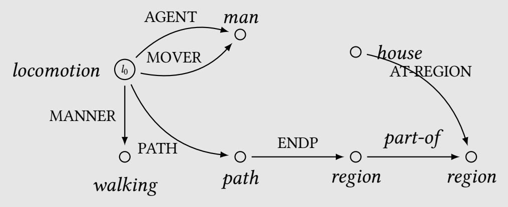
Given:

- Rel = Func ∪ PropRel (functional/non-functionsl relational symbols),
- Type (type symbols = propositional variables),
- Nom (nominals = node names), Nvar (node variables), Node := Nom ∪ Nvar.

$$\text{Forms} ::= \top \mid p \mid n \mid \langle R \rangle \phi \mid \mathbf{\exists} \phi \mid @_n \phi \mid \downarrow x.\phi \mid \exists x.\phi \mid \neg \phi \mid \phi_1 \wedge \phi_2$$

with $p \in \text{Type}$, $n \in \text{Node}$, $R \in \text{Rel}$, $\phi, \phi_1, \phi_2 \in \text{Forms}$.

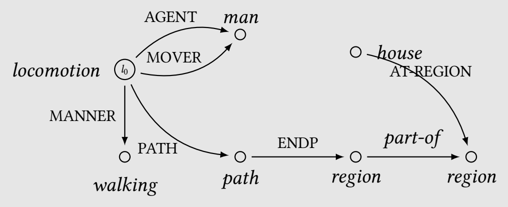
Hybrid logic for frames



The truth of a formula is given with respect to a specific node w of a model $\mathcal{M}$ and some assignment mapping Node to the nodes in $\mathcal{M}$. (For Nvar, this is g.)

- $\exists\phi$ is true in w if there exists a w' in $\mathcal{M}$ that makes ϕ true. (I.e., we move into some node in our frame and there ϕ is true.)
  $\exists$*house* is true in any node in $\mathcal{M}_1$.
  As usual: $\forall\phi \equiv \neg\exists(\neg\phi)$
  $\forall(path \rightarrow \langle \text{ENDP} \rangle \top)$ is true in any node in $\mathcal{M}_1$.
- $@_n\phi$ is true in w if ϕ is true in the node assigned to n. (We move into the (unique) node named n and there, ϕ holds.)
  $@_{l_0}$ *locomotion* is true in any node in $\mathcal{M}_1$.

Hybrid logic for frames



- $\downarrow x.\phi$ is true in w if ϕ is true in w under the assignment g^x_w.
  (We call the node we are located at x, and then ϕ is true in that node.)
  $\langle \text{PATH} \rangle \langle \text{ENDP} \rangle \langle \textit{part-of} \rangle \downarrow x.(\textit{region} \wedge \exists(\textit{house} \wedge \langle \text{AT-REGION} \rangle x))$
  is true in the *locomotion* node in $\mathcal{M}_1$.
- $\exists x.\phi$ is true in w if there is a w' such that ϕ is true in w under an assignment $g^x_{w'}$.
  (There is a node that we name x but for the evaluation of ϕ, we do not move to that node.)
  $\exists x.\langle \text{PATH} \rangle \langle \text{ENDP} \rangle \langle \textit{part-of} \rangle (x \wedge \textit{region}) \wedge \exists(\textit{house} \wedge \langle \text{AT-REGION} \rangle x)$
  is true in the *locomotion* node in $\mathcal{M}_1$.

HL and the syntax-semantics interface

Idea:

- Pair each syntactic building block with a set of underspecified HL formulas, which can contain holes and which can be labeled.
- Composition is then triggered by unifications of interface features that arise from syntactic composition.

In the following, $\{0, 1, 2, \ldots\}$ are HL nominals while $\{\boxed{0}, \boxed{1}, \boxed{2}, \ldots\}$ are variables at the interface.

HL and the syntax-semantics interface

Example from C08: *psych* verbs and *-ment* nominalization:

Syntactic rule (*ment* construction):

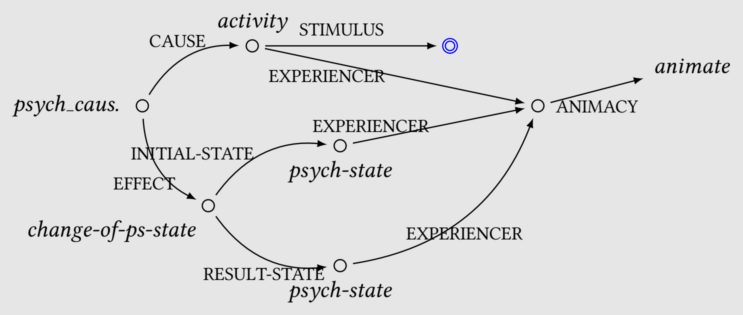
$[X_{[\textsc{cat}=V,\ \textsc{sem}=\boxed{0}]}\ \text{- ment}]_{[\textsc{cat}=N,\ \textsc{sem}=1]}$

Semantic constraints paired with this:

$$
\begin{aligned}
&\exists x.@_{\boxed{0}}[\mathit{psych\_causation}\\
&\quad \wedge\langle \textsc{cause}\rangle[\mathit{activity}\\
&\qquad \wedge\langle \textsc{stimulus}\rangle\top\\
&\qquad \wedge\langle \textsc{experiencer}\rangle[x \wedge \langle \textsc{animacy}\rangle \mathit{animate}]]\\
&\quad \wedge\langle \textsc{effect}\rangle[\mathit{change\text{-}of\text{-}psych\text{-}state}\\
&\qquad \wedge\langle \textsc{initial\text{-}state}\rangle[\mathit{psych\text{-}state}\\
&\qquad\qquad \wedge\langle \textsc{experiencer}\rangle x]\\
&\qquad \wedge\langle \textsc{result\text{-}state}\rangle[\mathit{psych\text{-}state}\\
&\qquad\qquad \wedge\langle \textsc{experiencer}\rangle x]]\\
&\quad \wedge[\langle \textsc{cause}\rangle 1 \vee \langle \textsc{cause}\rangle\langle \textsc{stimulus}\rangle 1\\
&\qquad \vee\langle \textsc{effect}\rangle 1 \vee \langle \textsc{effect}\rangle\langle \textsc{result\text{-}state}\rangle 1]]
\end{aligned}
$$

HL and the syntax-semantics interface

There are several minimal models (= models whose nodes are all described in the formula) for the resulting semantics of a *ment*-construction.



etc.

Such a computation of a minimal model can be done by XMG (Lichte & Petitjean, 2015).

HL and the syntax-semantics interface

We can separate constraints on the form of psych verbs from the *ment*-construction:

General constraints:

(5) a. $\forall \textit{psych\_causation} \rightarrow \textit{causation}$
b. $\forall \textit{causation} \rightarrow \langle \textsc{cause} \rangle \top \wedge \langle \textsc{effect} \rangle \top$
c. $\forall \textit{psych\_causation} \rightarrow [\langle \textsc{cause} \rangle \langle \textsc{stimulus} \rangle \top \wedge \langle \textsc{cause} \rangle \langle \textsc{experiencer} \rangle \top]$
d. $\forall \textit{psych\_causation} \rightarrow \exists x.[\langle \textsc{cause} \rangle \langle \textsc{experiencer} \rangle x \wedge \langle \textsc{effect} \rangle \langle \textsc{experiencer} \rangle x]$
e. …

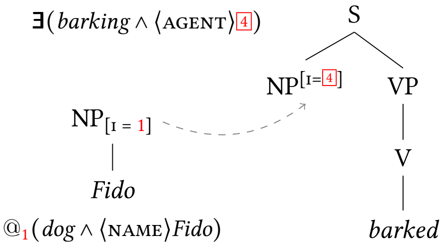
ment-construction:

(6) a. $[\mathrm{X}_{[\textsc{cat}=\mathrm{V},\ \textsc{sem}=\boxed{0}]}\text{ - ment}]_{[\textsc{cat}=\mathrm{N},\ \textsc{sem}=1]}$
b. $@_{\boxed{0}}[\textit{psych\_causation} \wedge [\langle \textsc{cause} \rangle 1 \vee \langle \textsc{cause} \rangle \langle \textsc{stimulus} \rangle 1 \vee \langle \textsc{effect} \rangle 1 \vee \langle \textsc{effect} \rangle \langle \textsc{result-state} \rangle 1]]$

HL and the syntax-semantics interface

A simple example from A02:

(7) Fido barked.



Putting things together (=conjoining) yields

(8) $\exists(\mathit{barking} \wedge \langle \textsc{agent} \rangle 1) \wedge @_1(\mathit{dog} \wedge \langle \textsc{name} \rangle \mathit{Fido})$

HL and the syntax-semantics interface

Adding underspecification: We can apply *hole semantics* (Bos, 1995) to our logic:

- We remove subformulas and replace them with *holes* (we "unplug" them).
- We introduce *dominance constraints* that tell us that a specific formula must be somewhere under a hole, i.e., must be a subformula of the formula that this hole stands for.
- We give names (= *labels*) to formulas in order to describe the way they are involved in dominance constraints.

(9) a. every dog barked

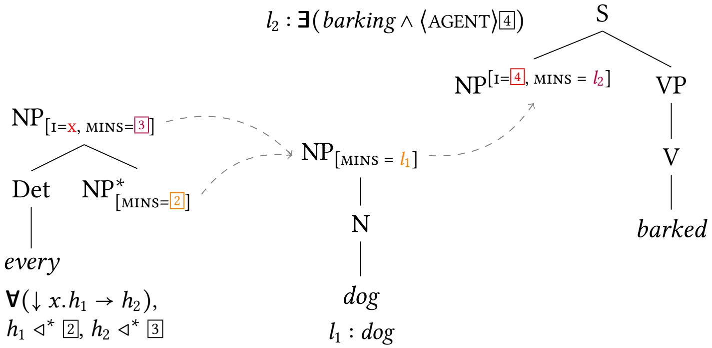
b. $\forall(\downarrow x.dog \rightarrow \exists(barking \wedge \langle \text{AGENT} \rangle x))$

c. $\forall(\downarrow x.h_1 \rightarrow h_2)$, $l_1 : dog$, $l_2 : \exists(barking \wedge \langle \text{AGENT} \rangle x)$,
$h_1 \triangleleft^* l_1$, $h_2 \triangleleft^* l_2$

This is interesting because it allows us to capture several readings in one underspecified representation.

HL and the syntax-semantics interface

TAG derivation for (9-a):



(10) $\forall(\downarrow x.h_1 \rightarrow h_2),\ l_1 : dog,\ l_2 : \exists(barking \wedge \langle \textsc{agent} \rangle x),$
$h_1 \triangleleft^* l_1,\ h_2 \triangleleft^* l_2$

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

Atelicity and telicity and *for*-adverbials

(11) Peter swam for one hour

(12) Peter knocked at the door for ten minutes

- In (11), the verb denotes an activity and is thus immediately compatible with the *for*-adverbial.
- In (12), the verb denotes a punctual event, and, hence, calls for additional adjustments in order to be compatible with *for*-adverbials.

⇒ (12) is interpreted as describing a sequence or iteration of knockings.

Atelicity and telicity and *for*-adverbials

Semantics of *for*-adverbials following Champollion (2013):

(13) $\lambda P \lambda I[AT(P,I) \wedge \mathit{hours}(I) = 1 \wedge \forall \mathcal{J}[\mathcal{J} \in \mathcal{R}_I^{\mathit{short}(I)} \rightarrow AT(P,\mathcal{J})]]$

In other words, a *for*-adverbial can only apply to an event P if we can fix a partition of the entire time interval I such that in each of the smaller intervals $\mathcal{J}$, P holds as well.

- *swim* can be directly used as P.
- In the case of *knock*, one has to apply an iteration operator first (**knock*), and the result can then become the argument of (13).

Goal of Kallmeyer et al. (2015b): provide an analysis with a similar semantics for *for*-adverbials that avoids the assumption of an iteration operator.

Atelicity and telicity and *for*-adverbials

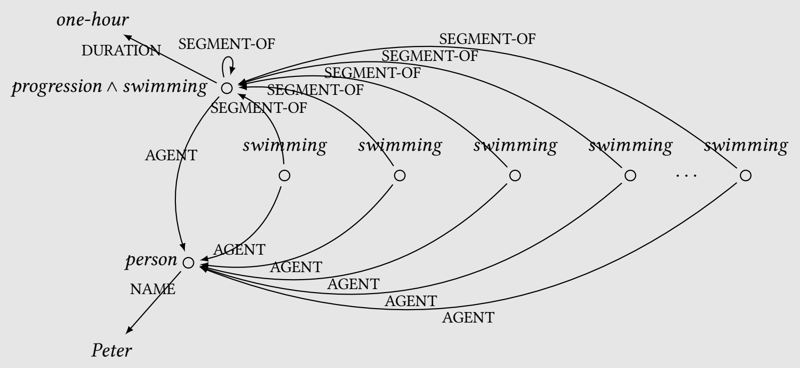
Why frames? And why HL and underspecification?

- Frames and HL allow for a fine-grained event decomposition and for quantification over subevents.
- Underspecification in the logical formulas (= dominance constraints) allows to distinguish between embedded subevents and the entire event and to “leave some room” for adverbials to apply.
- Underspecification in the frame types, in combination with appropriate HL constraints on frames, allows us to underspecify the type of the event resulting from applying a *for*-adverbial while making this type dependent on the type of the embedded event.

Atelic events

(11) Peter swam for one hour

Representation in the frame:



Atelic events

(14) Peter + swam

$@_i(\textit{person} \wedge \langle \textsc{name} \rangle \textit{Peter}) \wedge \exists \textit{swimming} \wedge \langle \textsc{agent} \rangle i$

Idea:

- We want the $\exists$ to scope over the entire progression.
- Such a progression can be conceived as a sequence of smaller events of the same type.
- The $\textit{swimming} \wedge \langle \textsc{agent} \rangle i$ describes these smaller events.
- Since the event is a progression, a general constraint will take care of lifting the *swimming* type to the entire event.

We therefore unplug the *swimming*… formula. Semantics of *swam*:

(15) $\exists h_1, l_1 : \textit{swimming} \wedge \langle \textsc{agent} \rangle i,\ h_1 \triangleleft^* l_1$

After combination with *Peter* and disambiguation, we obtain (14).

Atelic events

(11) Peter swam for one hour

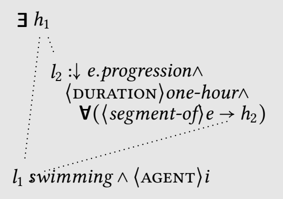
For *Peter swam*, we have

(16) $@_i(\mathit{person} \wedge \langle\textsc{name}\rangle \mathit{Peter}) \wedge$
$\exists h_1, l_1 : \mathit{swimming} \wedge \langle\textsc{agent}\rangle i,\ h_1 \lhd^* l_1$

This combines (conjoins) with the semantics of *for one hour*:

(17) a. for one hour

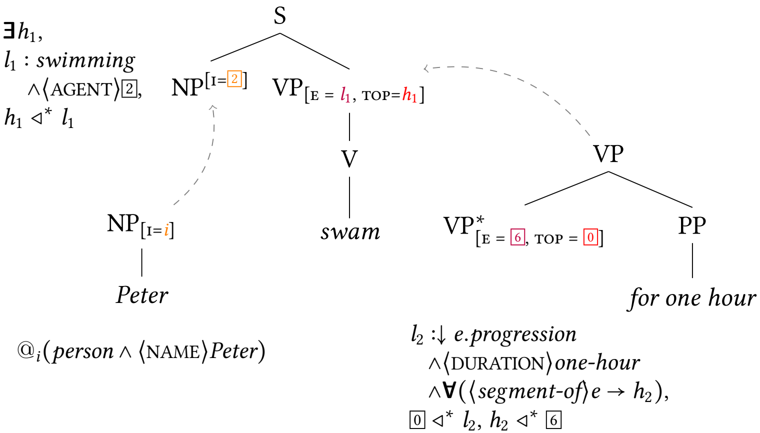
b. $l_2 :\downarrow e.\mathit{progression} \wedge \langle\textsc{duration}\rangle \mathit{one\text{-}hour}$
$\wedge\forall(\langle \mathit{segment\text{-}of}\rangle e \rightarrow h_2)$,
$h_1 \lhd^* l_2,\ h_2 \lhd^* l_1$



Atelic events

Syntax-semantics interface with TAG:

(11) Peter swam for one hour



Atelic events

This yields the underspecified representation:

(18) $@_i(\mathit{person} \wedge \langle \textsc{name} \rangle \mathit{Peter})$,
$\exists h_1, l_2 :\downarrow e.\mathit{progression} \wedge \langle \textsc{duration} \rangle \mathit{one\text{-}hour} \wedge \forall(\langle \mathit{segment\text{-}of} \rangle e \rightarrow h_2)$,
$l_1 : \mathit{swimming} \wedge \langle \textsc{agent} \rangle i$,
$h_1 \triangleleft^* l_1,\ h_1 \triangleleft^* l_2,\ h_2 \triangleleft^* l_1$

After disambiguation, one obtains:

(19) $@_i(\mathit{person} \wedge \langle \textsc{name} \rangle \mathit{Peter})$
$\wedge \exists \downarrow e.(\mathit{progression} \wedge \langle \textsc{duration} \rangle \mathit{one\text{-}hour} \wedge$
$\forall(\langle \mathit{segment\text{-}of} \rangle e \rightarrow \mathit{swimming} \wedge \langle \textsc{agent} \rangle i))$

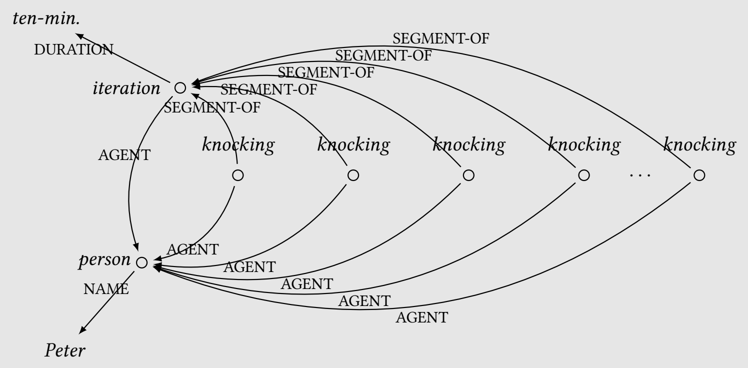
Additional constraint lifting P to the entire event:

(20) $\forall(\downarrow e.\mathit{progression} \rightarrow \langle \mathit{segment\text{-}of} \rangle e)$

Punctual events

(12) Peter knocked at the door for ten minutes

Representation in the frame:



Punctual events

(12) Peter knocked at the door for ten minutes

We adopt a more general type *nq-event* which is a supertype of *progression* and *iteration* and which is intended to capture *non-quantized* event types in the sense of Krifka (1998).

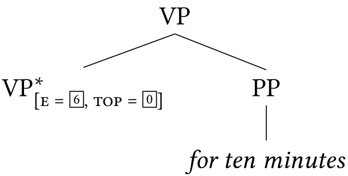
(21) $\forall(\textit{nq-event} \leftrightarrow \textit{iteration} \vee \textit{progression})$
$\forall(\textit{iteration} \rightarrow \neg\textit{progression})$

Additional constraints on iterations and progressions concerning the possible types of their segments:

(22) $\forall(\langle\textit{segment-of}\rangle\textit{iteration} \rightarrow \textit{bounded})$
$\forall(\textit{punctual} \rightarrow \textit{bounded})$
$\forall(\langle\textit{segment-of}\rangle\textit{progression} \rightarrow \neg\textit{bounded})$

Punctual events

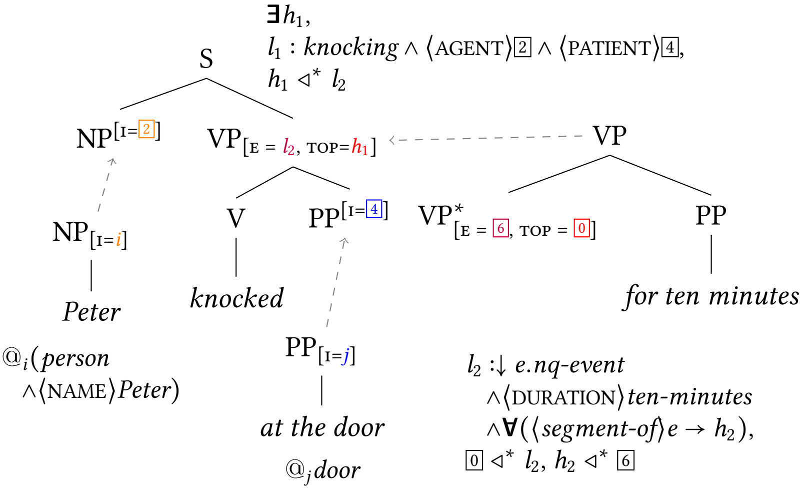
We generalize the lexical entry for *for*-adverbials:



$$\begin{aligned} & l_2 :\downarrow e.\textit{nq-event} \\ & \quad \wedge \langle \textsc{duration} \rangle \textit{ten-minutes} \\ & \quad \wedge \forall (\langle \textit{segment-of} \rangle e \rightarrow h_2), \\ & \boxed{0} \lhd^* l_2,\ h_2 \lhd^* \boxed{6} \end{aligned}$$

Depending on the type of the event segments, the non-quantized event is either a progression or an iteration.

Punctual events



Punctual events

Result:

(23) $\exists h_1,$
$l_1 : knocking \wedge \langle \text{AGENT} \rangle i \wedge \langle \text{PATIENT} \rangle j,$
$l_2 :\downarrow e.nq\text{-}event \wedge \langle \text{DURATION} \rangle ten\text{-}minutes \wedge \forall(\langle segment\text{-}of \rangle e \rightarrow h_2),$
$@_i(person \wedge \langle \text{NAME} \rangle Peter),$
$@_j door,$
$h_1 \lhd^* l_1,\ h_1 \lhd^* l_2,\ h_2 \lhd^* l_1$

After disambiguation:

(24) $\exists(\downarrow e.nq\text{-}event \wedge \langle \text{DURATION} \rangle ten\text{-}minutes$
$\wedge\forall(\langle segment\text{-}of \rangle e \rightarrow knocking \wedge \langle \text{AGENT} \rangle i \wedge \langle \text{PATIENT} \rangle j))$
$\wedge @_i(person \wedge \langle \text{NAME} \rangle Peter) \wedge @_j door$

With our constraints and with $\forall(knocking \rightarrow punctual)$, e in (24) is necessarily of type *iteration*.

Conclusion

- Frames allow for a fine-grained representation of meaning where generalizations can be captured via types and constraints on types.
- We can use a logic in order to formulate constraints on frames and in order to integrate logical operators.
- We can use standard underspecification techniques on this logic in order to abstract over several readings.

We have seen that this combination allows us to capture semantic generalizations in an elegant way.

Areces, Carlos & Balder ten Cate. 2007. Hybrid logics. In Patrick Blackburn, Johan Van Benthem & Frank Wolter (eds.), *Handbook of modal logic*, chap. 14, 821–868. Elsevier.

Blackburn, Patrick. 1993. Modal logic and attribute value structures. In Maarten de Rijke (ed.), *Diamonds and defaults*, 19—-65. Kluwer, Dordrecht.

Blackburn, Patrick, Johan Van Benthem & Frank Wolter (eds.). 2007. *Handbook of modal logic*, vol. 3 Studies in Logic and Practical Reasoning. Elsevier.

Bos, Johan. 1995. Predicate logic unplugged. In Paul Dekker & Martin Stokhof (eds.), *Proceedings of the 10th amsterdam colloquium*, 133–142.

Champollion, Lucas. 2013. The scope and processing of *for*-adverbials: A reply to Deo and Piñango. In *Proceedings of SALT*, 432–452.

Kallmeyer, Laura, Timm Lichte, Rainer Osswald, Sylvain Pogodalla & Christian Wurm. 2015a. Quantification in frame semantics with hybrid logic. Tech. Rep. ⟨hal-01151641v3⟩ HAL-Inria. Abstract presented at *TYTLES: TYpe Theory and LExical Semantics* workshop, ESSLLI, Barcelona.

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. 2015b. Progression and iteration in event semantics – an ltag analysis using hybrid logic and frame semantics. Tech. Rep. ⟨hal-01184872⟩ HAL-Inria. Abstract presented at *Colloque de Syntaxe et Sémantique à Paris (CSSP 2015)*.

Kallmeyer, Laura & Frank Richter. 2014. Quantifiers in frame semantics. In Glyn Morrill, Reinhard Muskens, Rainer Osswald & Frank Richter (eds.), *Formal grammar*, vol. 8612 Lecture Notes in Computer Science, 69–85. Springer Berlin Heidelberg.

Krifka, Manfred. 1998. The origins of telicity. In Susan Rothstein (ed.), *Events and grammar*, 197–235. Dordrecht: Kluwer.

Lichte, Timm & Simon Petitjean. 2015. Implementing semantic frames as typed feature structures with xmg. *Journal of Language Modelling* 3(1). 185–228.

Löbner, Sebastian. 2015. Frame theory with first-order comparators: Modeling the lexical meaning of verbs of change with frames. Presentation at the 11th TbiLLC, Tbilisi, Georgia.

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* $\diamond\phi$, chap. 24, 175–183. Pumbo.nl.

Petersen, Wiebke. 2007. Representation of concepts as frames. In *The baltic international yearbook of cognition, logic and communication*, vol. 2, 151–170.

Petersen, Wiebke & Tanja Osswald. 2014. Concept composition in frames: Focusing on genitive constructions. In Thomas Gamerschlag, Doris Gerland, Rainer Osswald & Wiebke Petersen (eds.), *Frames and Concept Types. Applications in Language and Philosophy*, vol. 94 Studies in Linguistics and Philosophy, 243–266. Dordrecht: Springer.