On Frames and their components

Wiebke Petersen

Heinrich-Heine-Universität Düsseldorf
Collaborative Research Centre 991:
The Structure of Representations in Language, Cognition, and Science

www.phil-fak.uni-duesseldorf.de/~petersen/

17. April 2012, Tilburg
1. Frame hypothesis

2. Frames as generalized typed feature structures

3. Attributes in frames are types (1st perspective)

4. Types are definable by attributes (2nd perspective)

5. Outlook
1 Frame hypothesis

2 Frames as generalized typed feature structures

3 Attributes in frames are types (1st perspective)

4 Types are definable by attributes (2nd perspective)

5 Outlook
1. Frame hypothesis

2. Frames as generalized typed feature structures

3. Attributes in frames are types (1st perspective)

4. Types are definable by attributes (2nd perspective)

5. Outlook
1. Frame hypothesis
2. Frames as generalized typed feature structures
3. Attributes in frames are types (1st perspective)
4. Types are definable by attributes (2nd perspective)
5. Outlook
1. Frame hypothesis

2. Frames as generalized typed feature structures

3. Attributes in frames are types (1st perspective)

4. Types are definable by attributes (2nd perspective)

5. Outlook
1. **Frame hypothesis**

2. Frames as generalized typed feature structures

3. Attributes in frames are types (1st perspective)

4. Types are definable by attributes (2nd perspective)

5. Outlook
Frame hypothesis (Löbner 2012)

**H1** The human cognitive system operates with one general format of representations.

**H2** If the human cognitive system operates with one general format of representations, this format is essentially Barsalou frame.

**A frame model is needed, that**
- is sufficiently expressive to capture the diversity of representations
- sufficiently precise and restrictive in order to be testable

**Aim:** theory of concepts based on frames as concept representations
### Düsseldorf frame group

- Semantics
- Syntax
- Computational Linguistics
- Psycholinguistics
- Neurolinguistics
- Neuroscience
- Cognitive Science
- Psychology
- Philosophy
- History of Science
- Philologies (German, Romanistic)
The task

Formalizing Barsalou’s cognitive frame theory

- bridging the gap between cognitive linguistics and compositional semantics

Hypothesis: Frames can be defined as generalized typed feature structures (in the sense of Carpenter 1992)
# Outline

1. Frame hypothesis
2. Frames as generalized typed feature structures
3. Attributes in frames are types (1st perspective)
4. Types are definable by attributes (2nd perspective)
5. Outlook

- Frames provide the fundamental representation of knowledge in human cognition.
- At their core, frames contain **attribute-value sets**.
- Frames further contain a variety of relations.
  - **Constraints**
Example: vacation frame with constraints (Barsalou 1992)
Unlimited recursion in frames

Self-similarity in Barsalou’s frames (attributes are frames):

Recursion in classical feature structure theories:
feature structures

typed feature structure

\[
\begin{align*}
\text{phrase} & \quad \text{noun} \\
\text{HEAD:} & \quad \text{agr} \\
\text{AGR:} & \quad \text{PERS: 3} \\
& \quad \text{NUM: pl}
\end{align*}
\]

untyped feature structure

\[
\begin{align*}
\text{CAT: phrase} & \quad \text{noun} \\
\text{HEAD:} & \quad \text{agr} \\
\text{AGR:} & \quad \text{PERS: 3} \\
& \quad \text{NUM: pl}
\end{align*}
\]
feature structures

**typed feature structure**

```
phrase

HEAD:  [noun

AGR:  [agr

PERS: 3

NUM: pl]
```

**untyped feature structure**

```
CAT: phrase

HEAD:  [CAT: noun

AGR:  [PERS: 3

NUM: pl]
```
frames as generalized feature structures

feature structures (Carpenter 1992)

feature structures are connected directed graphs with
- one central node
- nodes labeled with types
- arcs labeled with attributes
- no node with two outgoing arcs with the same label
- and such that each node can be reached from the central node via directed arcs.

\[
\begin{align*}
\text{tree} & \quad \text{CROWN} : \text{crown} \\
\text{trunk} & \quad \text{BARK} : \text{bark} \\
& \quad \text{GIRTH} : \text{girth}
\end{align*}
\]
frames as generalized feature structures

Frames (Petersen 2007)

Frames are connected directed graphs with
- one central node
- nodes labeled with types
- arcs labeled with attributes
- no node with two outgoing arcs with the same label

Open argument nodes are marked as rectangular nodes.

Frames are unrooted feature structures.
Formal Definitions

Definition (Frames)

Given a set \( \text{TYPE} \) of types and a finite set \( \text{ATTR} \) of attributes. A *frame* is a tuple \( F = (Q, \bar{q}, \delta, \theta) \) where:

- \( Q \) is a finite set of nodes,
- \( \bar{q} \in Q \) is the central node,
- \( \delta : \text{ATTR} \times Q \rightarrow Q \) is the partial *transition function*,
- \( \theta : Q \rightarrow \text{TYPE} \) is the total *node typing function*,

such that the underlying graph \((Q, E)\) with edge set

\[
E = \{\{q_1, q_2\} | \exists a \in \text{ATTR} : \delta(a, q_1) = q\}
\]

is connected.
Definition (Subsumption)

A frame $F_1 = \langle Q_1, \bar{q}_1, \delta_1, \theta_1 \rangle$ subsumes a frame $F_2 = \langle Q_2, \bar{q}_2, \delta_2, \theta_2 \rangle$ ($F_1 \sqsubseteq F_2$) iff there is a total function $h : Q_1 \rightarrow Q_2$ with

- $h(\bar{q}_1) = \bar{q}_2$,
- $\forall q \in Q_1 : \theta_1(q) \sqsubseteq \theta_2(h(q))$,
- if $\delta_1(f, q)$ is defined, then $h(\delta_1(f, q)) = \delta_2(f, h(q))$.

(Carpenter 1992)

Definition (Equivalence)

Two frames $F_1$ and $F_2$ are equivalent ($F_1 \sim F_2$), if $F_1 \sqsubseteq F_2$ and $F_2 \sqsubseteq F_1$. 
### Definition (Subsumption)

A frame $F_1 = \langle Q_1, \bar{q}_1, \delta_1, \theta_1 \rangle$ subsumes a frame $F_2 = \langle Q_2, \bar{q}_2, \delta_2, \theta_2 \rangle$ ($F_1 \sqsubseteq F_2$) iff there is a total injective function $h : Q_1 \rightarrow Q_2$ with:

- $h(\bar{q}_1) = \bar{q}_2$,
- $\forall q \in Q_1 : \theta_1(q) \sqsubseteq \theta_2(h(q))$,
- if $\delta_1(f, q)$ is defined, then $h(\delta_1(f, q)) = \delta_2(f, h(q))$.

### Definition (Equivalence)

Two frames $F_1$ and $F_2$ are equivalent ($F_1 \sim F_2$), if $F_1 \sqsubseteq F_2$ and $F_2 \sqsubseteq F_1$. 

Wiebke Petersen

Frames
Adaption of subsumption relation
Adaption of subsumption relation

[Framed diagram showing relationships between 'child', 'person', and 'FATHER' roles.]

Wiebke Petersen
Adaption of subsumption relation
why typed frames and type signatures?

modeling convention (Carpenter 1992:34)

- The nodes of a feature structure are taken to represent objects, and we assume that every node is labeled with a type symbol which represents the most specific conceptual class to which the object is known to belong.

- An arc between two nodes indicates that the object represented by the source node has a feature, represented by a feature symbol, which has a value represented by the target node.

- We think of our types as organizing feature structures into natural classes.
why typed frames and type signatures?

Type signature

A type signature consists of a type hierarchy, \( \langle T, \sqsubseteq \rangle \), a finite set of attributes \( \text{ATTR} \) and an **appropriateness specification**, i.e. a partial function, \( \text{Approp} : \text{ATTR} \times T \rightarrow T \) that respects:

- attribute introduction (each attribute is introduced at a unique most general type)
- upward closure / right monotonicity (inheritance of appropriateness conditions)
why typed frames and type signatures?

- Type signatures
  - capture hierarchical relations
  - capture generalizations
  - express constraints
  - enable underspecified frames
why typed frames and type signatures?

```plaintext
noun
AGR: agr

agr
PERS: pers
NUM: num

atomic

pers
num

<table>
<thead>
<tr>
<th>pers</th>
<th>sing</th>
<th>plur</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

[\begin{array}{c}
\text{noun} \\
\text{AGR: agr} \\
\text{PERS: pers} \\
\text{NUM: num}
\end{array}]

However,

- redundancy in attribute and type labels
- status of types is not clear (Carpenter 1992: types represent conceptual classes)
- frames and types: two means of expressing concepts?
possible solutions

1st perspective:
The distinction between the attribute set and the type set is artificial. The attribute set should be taken as a subset of the type set.

\[ \text{ATTR} \subseteq \text{TYPE} \]

2nd perspective:
The distinction between the attribute set and the type set is artificial. Types are definable on the basis of attribute domain and ranges.

\[ \text{TYPE} \rightsquigarrow \text{ATTR} \]
1. Frame hypothesis

2. Frames as generalized typed feature structures

3. Attributes in frames are types (1st perspective)

4. Types are definable by attributes (2nd perspective)

5. Outlook
attributes in frames

Barsalou, 1992
“I define an attribute as a concept that describes an aspect of at least some category member.”
“Values are subordinate concepts of an attribute.”

Guarino, 1992: *Concepts, attributes and arbitrary relations*
“We define attributes as concepts having an associate relational interpretation, allowing them to act as conceptual components as well as concepts on their own.”
excurus: interpretation of functional concepts

**denotational interpretation**

A functional concept denotes a set of entities:

\[ \delta : \mathcal{R} \rightarrow 2^\mathcal{U} \]

\[ \delta(\text{mother}) = \{ m \mid m \text{ is the mother of someone} \} \]

**relational interpretation**

A functional concept has also a relational interpretation:

\[ \varrho : \mathcal{R} \rightarrow 2^{\mathcal{U} \times \mathcal{U}} \]

\[ \varrho(\text{mother}) = \{ (p, m) \mid m \text{ is the mother of } p \} \]

**consistency postulate (Guarino, 1992)**

Any value of an relationally interpreted functional concept is also an instance of the denotation of that concept.

If \((p, m) \in \varrho(\text{mother})\), then \(m \in \delta(\text{mother})\).
**excursus: interpretation of functional concepts**

<table>
<thead>
<tr>
<th>denotational interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A functional concept denotes a set of entities:</td>
</tr>
<tr>
<td>[ \delta : R \rightarrow 2^U ]</td>
</tr>
<tr>
<td>[ \delta(\text{mother}) = { m \mid m \text{ is the mother of someone} } ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>relational interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A functional concept has also a relational interpretation:</td>
</tr>
<tr>
<td>[ \varrho : R \rightarrow 2^{U \times U} ]</td>
</tr>
<tr>
<td>[ \varrho(\text{mother}) = { (p, m) \mid m \text{ is the mother of } p } ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>consistency postulate (Guarino, 1992)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any value of an relationally interpreted functional concept is also an instance of the denotation of that concept.</td>
</tr>
<tr>
<td>If ((p, m) \in \varrho(\text{mother})), then (m \in \delta(\text{mother})).</td>
</tr>
</tbody>
</table>
excursus: interpretation of functional concepts

**denotational interpretation**
A functional concept denotes a set of entities:

\[ \delta : \mathcal{R} \rightarrow 2^\mathcal{U} \]

\[ \delta(\text{mother}) = \{ m \mid m \text{ is the mother of someone} \} \]

**relational interpretation**
A functional concept has also a relational interpretation:

\[ \rho : \mathcal{R} \rightarrow 2^{\mathcal{U} \times \mathcal{U}} \]

\[ \rho(\text{mother}) = \{ (p, m) \mid m \text{ is the mother of } p \} \]

**consistency postulate (Guarino, 1992)**
Any value of an relationally interpreted functional concept is also an instance of the denotation of that concept.

If \((p, m) \in \rho(\text{mother})\), then \(m \in \delta(\text{mother})\).
attributes in frames

**Thesis:** Attributes in frames are relationally interpreted functional concepts!

- attributes are not frames themselves
- attributes are unstructured
- the possible values of an attribute are subconcepts of the denotationally interpreted functional concept

Diagram:

```
T
 o----o
 |     |
 o----o
object
 COLOR: color
 SHAPE: shape

apple
 SHAPE: round

color
 red green blue round long
```

```
apple
 color
 round
```

Wiebke Petersen
denotational and relational interpretation

**Denotational**

- **Type**: mother
- **MOTHER**: $\delta(\text{person}) \rightarrow \delta(\text{mother})$

**Relational**

- **Attribute**: MOTHER

![Diagram showing denotational and relational interpretations of MOTHER]
attributes in frames (1st perspective)

**thesis:**
Attributes in frames are relationally interpreted functional concepts!

**consequence (1):**
Frames decompose concepts into relationally interpreted functional concepts!

**consequence (2):**
The distinction between the attribute set and the type set is artificial. The attribute set should be taken as a subset of the type set: $\text{ATTR} \subseteq \text{TYPE}$.
attributes in frames (1st perspective)

**thesis:**
Attributes in frames are relationally interpreted functional concepts!

**consequence (1):**
Frames decompose concepts into relationally interpreted functional concepts!

**consequence (2):**
The distinction between the attribute set and the type set is artificial. The attribute set should be taken as a subset of the type set: \( ATTR \subseteq TYPE \).
attributes in frames (1st perspective)

thesis:
Attributes in frames are relationally interpreted functional concepts!

consequence (1):
Frames decompose concepts into relationally interpreted functional concepts!

consequence (2):
The distinction between the attribute set and the type set is artificial. The attribute set should be taken as a subset of the type set: \( \text{ATTR} \subseteq \text{TYPE} \).
Barsalou, 1992: *Frames, Concepts, and Conceptual Fields*

“I define an attribute as a **concept** that describes an aspect of at least some category member.”

“Values are subordinate concepts of an attribute.”
**Definition**

A **minimal upper attribute** of a type is a minimal element of the set of upper attributes of the type. Where an upper attribute of a type is an attribute which is a supertype of the type.
red apple
ATTR ⊆ TYPE

object
TASTE
TEMPERATURE:
COLOR
SHAPE

apple
SHAPE: round

pepper

taste
temperature
color
shape

sour
sweet
hot
cold
red
green
blue
round
long

red apple
\[ \text{ATTR} \subseteq \text{TYPE} \]

- **Object**
  - Taste
  - Temperature
  - Color
  - Shape

- **Taste**: sour, sweet
- **Temperature**: hot, cold
- **Color**: red, green, blue
- **Shape**: round, long

**Frames**

- **Apple**
  - Color: red
  - Shape: round

**Outlook**

Wiebke Petersen
\[ \text{ATTR} \subseteq \text{TYPE} \]

object

TASTE

TEMPERATURE:

COLOR

SHAPE

apple

SHAPE: round

pepper

taste temperature color shape

sour sweet hot cold red green blue round long

red apple
ATR \subseteq TYPE

object
  TASTE
  TEMPERATURE:
    COLOR
    SHAPE

apple
  SHAPE: round

pepper

TASTE
  sour
  sweet
  hot
  cold

COLOR
  red
  green
  blue

SHAPE
  round
  long

red apple

apple
  COLOR
    red
  SHAPE
    round

Wiebke Petersen
Frame hypothesis

Frames as generalized typed feature structures

Attributes in frames are types (1st perspective)

Types are definable by attributes (2nd perspective)

Outlook
radically attribute-oriented perspective
radically attribute-oriented perspective
radically attribute-oriented perspective
radically attribute-oriented perspective
**Definition**

An attribute space is a tuple \((\mathcal{U}, \mathcal{A})\) consisting of a universe set \(\mathcal{U}\) and a finite set of attributes \(\mathcal{A} \subseteq 2^{\mathcal{U} \times \mathcal{U}}\) which are partial functions (i.e., if \((x, y), (x, z) \in \mathcal{A}\) then \(y = z\)).

Attribute composition: the set of paths \(\Pi\) in \((\mathcal{U}, \mathcal{A})\) is the set of all finite attribute sequences. If \(\pi = a_1 a_2 \ldots a_n\) we write \(\pi(x)\) for \(a_n(\ldots(a_2(a_1(x)))\ldots)\).
Definition

Given an attribute space \((\mathcal{U}, \mathcal{A})\) and a set \(S\) of relevant subsets of \(\mathcal{U}\). The set of types \(T\) is \(T = 2^S / \sim\) with:

- \(\forall \varphi, \psi \subseteq S : \varphi \sim \psi\) iff \(\bigcap \varphi = \bigcap \psi\).

\[
\{\square, \square\} \sim \{\square, \square, \square\}
\]
The type hierarchy \((\mathcal{T}, \sqsubseteq)\) is defined by

- \([\varphi] \sqsubseteq [\psi] \text{ iff } \bigcup [\varphi] \subseteq \bigcup [\psi]\)
- or equivalently (extensionally): \([\varphi] \sqsubseteq [\psi] \text{ iff } \bigcap \varphi \supseteq \bigcap \psi\)

\[
\{\square, \square\} \subseteq \{\square, \square\}
\]

The type hierarchy forms a lattice (top element \([\emptyset]\), bottom element \([S]\)).
relevant subsets (example)

\[ S = A_d \cup \Pi_r \text{ with:} \]

- \( A_d \) is the set of attribute domains:
  \[ A_d = \{ a_d | a \in A \} \text{ where } a_d = \{ x \in U | \exists u \in U : a(x) = u \} \]
- \( \Pi_r \) is the set of path ranges:
  \[ \Pi_r = \{ \pi_r | \pi \in \Pi \} \text{ where } \pi_r = \{ x \in U | \exists u \in U : \pi(u) = x \} \]

**Definition**

The type signature on \((U, A)\) with relevant subset set \(A_d \cup \Pi_r\) is \((T, \sqsubseteq, \text{Approp})\) where Approp : \(A \times T \rightarrow T\) is the appropriateness condition defined by:

- \( \text{Approp}(a, [\varphi]) = [\{(\pi a)_r | \pi_r \in U[\varphi]\}] \)
adjusting the granularity of the type hierarchy

The granularity of the type hierarchy can be easily adjusted by adapting the set of relevant subsets $S$

examples of attribute-defined relevant subsets
**FCA\textsc{Type}**

- **FCA\textsc{Type}** is a system for the automatic induction of type signatures from sets of untyped feature structures (i.e., sortal frames).
- It uses methods of Formal Concept Analysis (Ganter & Wille 1998).
- key idea: decomposition of feature structures into paths, path equations and path-value-pairs (note: attribute-based components).
- It can be straightforwardly adapted to general frames.
example input frames

Uther =
\[
\begin{array}{c}
\text{CAT: } \text{np} \\
\text{HEAD: } \begin{array}{c}
\text{AGR: } \\
\text{PERS: third} \\
\text{NUM: sing}
\end{array}
\end{array}
\]

sleeps =
\[
\begin{array}{c}
\text{CAT: } \text{vp} \\
\text{HEAD: } \begin{array}{c}
\text{FORM: } \text{finite} \\
\text{SUBJ: } \begin{array}{c}
\text{CAT: } \text{np} \\
\text{HEAD: } \begin{array}{c}
\text{AGR: } \\
\text{PERS: third} \\
\text{NUM: sing}
\end{array}
\end{array}
\end{array}
\end{array}
\]

knights =
\[
\begin{array}{c}
\text{CAT: } \text{np} \\
\text{HEAD: } \begin{array}{c}
\text{AGR: } \\
\text{PERS: third} \\
\text{NUM: plur}
\end{array}
\end{array}
\]

sleep =
\[
\begin{array}{c}
\text{CAT: } \text{vp} \\
\text{HEAD: } \begin{array}{c}
\text{FORM: } \text{finite} \\
\text{SUBJ: } \begin{array}{c}
\text{CAT: } \text{np} \\
\text{HEAD: } \begin{array}{c}
\text{AGR: } \\
\text{PERS: third} \\
\text{NUM: plur}
\end{array}
\end{array}
\end{array}
\end{array}
\]

(taken from Shieber 1986)
unfolded type signature

Wiebke Petersen
folded type signature

```
          T
         / \
      /     \ 

  t_2 HEAD : head
     CAT : cat
         /  \
      /    \ 
    t_9   t_15
      AGR : t_12 SUBJ : t_2 FORM : finite

  t_3 HEAD : t_9
     CAT : np
  t_6 HEAD : t_15
     CAT : vp

  t_12 PERS : third
     NUM : num

  atomic
     /  \
  pers   form
       /  \ 
  third  finite
       /   \
  sing  plur
  num   cat
        /  \
      vp   np
```
folded type signature

Carpenter 1992: “We think of our types as organizing feature structures into natural classes.”
The granularity of the type hierarchy can be adjusted to capture constraints:

- inverse images of (some) values (e.g. COLOR$^{-1}(\text{red})$; “if a tomato is ripe, it is red)

- inverse images of value ranges (e.g. AGE$^{-1}(\leq 18)$; “a human under 18 is a child”)

- attribute domains (e.g., SHAPE, SIZE; “if something has a shape, it has a size”)

- path ranges (e.g., HAIR COLOR; “hair colors are restricted”)

- path equations

- monotonic constraints (“the older a stamp is, the more expensive it is”)

...
1. Frame hypothesis

2. Frames as generalized typed feature structures

3. Attributes in frames are types (1st perspective)

4. Types are definable by attributes (2nd perspective)

5. Outlook
Düsseldorf frame model

We aim at a frame model that is

- powerful because of unlimited recursiveness (compare e.g. Fillmore frames)
- expressive because of type specifications
- precise because of restriction to functional attributes (compare e.g. semantic nets)
- formalized (mathematical definition and model-theoretic interpretation)
- empirical founded (evidence from cognitive science and psycholinguistics)
Why not traditional PL1 with truth-valued model theory?

**Advantages of frame approach**

- **transparent and preserving composition in frames**: the information of the parts is preserved, accumulated, and configurated
  (in truth-functional logic the meaning of the parts is not recoverable from the meaning of the whole)

- **variable freeness**: cognitive more adequate, information elements are related by attributes not by shared variables

- **no fixed arity of predicates**

- **no fixed argument order**
My project: Formal modeling of frames

**subjects**

1. frames in isolation
   - ontological status of frame elements
   - dynamic attributes
   - focus: space of attributes

2. frames in interaction
   - operations on frames (composition)
   - relations between frames (type shifts)
   - focus: space of frames

3. frame models of dynamic concepts
   - changes of attribute values in time
   - focus: linking object frames with the temporal domain

**Aim**

Frame-based cognitive semantics explaining both decompositional and compositional phenomena in a unified way


outline

6 concept classification and frame graphs

7 Type shifts

8 Concept composition
concept classification

person, pope, house, verb, sun, Mary, wood, brother, mother, meaning, distance, spouse, argument, entrance
**concept classification: relationality**

<table>
<thead>
<tr>
<th>non-relational</th>
<th>person, pope, house, verb, sun, Mary, wood</th>
</tr>
</thead>
<tbody>
<tr>
<td>relational</td>
<td>brother, mother, meaning, distance, spouse, argument, entrance</td>
</tr>
</tbody>
</table>

Löbner
### Concept Classification: Uniqueness of Reference

<table>
<thead>
<tr>
<th>Type</th>
<th>Non-Unique Reference</th>
<th>Unique Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Non-Relational</strong></td>
<td>person, house, verb, wood</td>
<td>Mary, pope, sun</td>
</tr>
<tr>
<td><strong>Relational</strong></td>
<td>brother, argument, entrance</td>
<td>mother, meaning, distance, spouse</td>
</tr>
</tbody>
</table>
## concept classification

<table>
<thead>
<tr>
<th></th>
<th>non-unique reference</th>
<th>unique reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>non-relational</td>
<td>sortal concept</td>
<td>individual concept</td>
</tr>
<tr>
<td>relational</td>
<td>proper relational concept</td>
<td>functional concept</td>
</tr>
</tbody>
</table>

Löbner
## Concept Classification

<table>
<thead>
<tr>
<th>Type</th>
<th>Non-unique Reference</th>
<th>Unique Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-relational</td>
<td>Sortal Concept</td>
<td>Individual Concept</td>
</tr>
<tr>
<td>Relational</td>
<td>Proper Relational Concept</td>
<td>Functional Concept</td>
</tr>
</tbody>
</table>
Definition

A node is a root of a frame if all other nodes can be reached from it by a path of directed arcs.
**Definition**
A node is a **root** of a frame if all other nodes can be reached from it by a path of directed arcs.

**Definition**
A node is a **source** if it has no incoming arc.
lolly-frame (sortal concept)
**lolly-frame (sortal concept)**

```
central node = root = source
```

- **lolly**: Frame graph representing a lolly with a red body and a green stick.
- **Red body**: Attributes: COLOR, SHAPE.
- **Green stick**: Attributes: COLOR, SHAPE.
- **Factory**: PRODUCER.
stick-frame (functional concept)
 Stick-frame (functional concept)

central node $\neq$ root $=$ source
sister-frame (proper relational concept)
sister-frame (proper relational concept)

no root & central node = source
classification of acyclic frame graphs

C: central node, R: root, S: source

<table>
<thead>
<tr>
<th>$C = R$</th>
<th>$C = S$</th>
<th>$\exists R$</th>
<th>$\exists S$</th>
<th>typical graph</th>
<th>frame class</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td><img src="image" alt="sortal graph" /></td>
<td>sortal</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td><img src="image" alt="functional graph" /></td>
<td>functional</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td><img src="image" alt="proper relational graph" /></td>
<td>proper relational</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td><img src="image" alt="questionable graph" /></td>
<td>???</td>
</tr>
</tbody>
</table>
4th frame class: not lexicalized?

relational concept: father of a niece
4th frame class: not lexicalized?

brother in law

male

SEX

person

SPOUSE

person

person

FATHER

MOTHER

FATHER

MOTHER

\neq\n
Wiebke Petersen

Frames
4th frame class: not lexicalized?

brother in law

male

SEX

person

SPOUSE

person

FATHER

MOTHER

=≠

person

FATHER

MOTHER

≠

person

SEX

male

SEX

person

SPOUSE

person

FATHER

MOTHER

=≠

person

FATHER

MOTHER

≠

person
4th frame class: not lexicalized?

Brother in law

“male person who is the spouse of someone who has a sibling”

“male person whose spouse has a sibling”
concept classification and frame graphs

relationality
The arguments of relational concepts are modeled in frames as sources that are not identical to the central node.

functionality
The functionality of functional concepts is modeled by an incoming arc at the central node.

conclusion
The concept classification is reflected by the properties of the frame graphs.
concept classification and frame graphs

relationality

The arguments of relational concepts are modeled in frames as sources that are not identical to the central node.

functionality

The functionality of functional concepts is modeled by an incoming arc at the central node.

conclusion

The concept classification is reflected by the properties of the frame graphs.
concept classification and frame graphs

**relationality**

The arguments of relational concepts are modeled in frames as sources that are not identical to the central node.

**functionality**

The functionality of functional concepts is modeled by an incoming arc at the central node.

**conclusion**

The concept classification is reflected by the properties of the frame graphs.
outline

6 concept classification and frame graphs

7 Type shifts

8 Concept composition
### Type shifts: non-relational $\rightarrow$ relational

<table>
<thead>
<tr>
<th>sortal</th>
<th>individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>proper relational</td>
<td>functional</td>
</tr>
</tbody>
</table>

sortal concept *flat*: "Many flats are offered in the newspaper."

```
HOUSING

TENANT

OWNER

Flat
```
type shifts: non-relational → relational

<table>
<thead>
<tr>
<th>sortal</th>
<th>individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>proper relational</td>
<td>functional</td>
</tr>
</tbody>
</table>

Proper relational concept *flat*: “This flat is a flat of John, he owns more than five.”
type shifts: non-relational $\rightarrow$ relational

<table>
<thead>
<tr>
<th>sortal</th>
<th>individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>proper relational</td>
<td>functional</td>
</tr>
</tbody>
</table>

functional concept *flat*: "The flat of Mary is huge and the rent is reasonable."

Wiebke Petersen
Frames 59
type shifts: relational $\rightarrow$ non-relational

<table>
<thead>
<tr>
<th>sortal</th>
<th>individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>proper relational</td>
<td>functional</td>
</tr>
</tbody>
</table>

functional concept *trunk*: “She sat with her back against the trunk of an oak.”
type shifts: relational → non-relational

<table>
<thead>
<tr>
<th>sortal</th>
<th>individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>proper relational</td>
<td>functional</td>
</tr>
</tbody>
</table>

sortal concept *trunk*: “They rested and sat on a trunk.”
outline

6 concept classification and frame graphs

7 Type shifts

8 Concept composition
hypothesis: composition works uniformly with respect to concept types

<table>
<thead>
<tr>
<th>RC</th>
<th>OF SC</th>
<th>SC</th>
<th>finger OF woman</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>OF IC</td>
<td>SC</td>
<td>finger OF Mary</td>
</tr>
<tr>
<td>RC</td>
<td>OF RC</td>
<td>RC</td>
<td>finger OF friend</td>
</tr>
<tr>
<td>RC</td>
<td>OF FC</td>
<td>RC</td>
<td>finger OF spouse</td>
</tr>
<tr>
<td>FC</td>
<td>OF SC</td>
<td>SC</td>
<td>head OF woman</td>
</tr>
<tr>
<td>FC</td>
<td>OF IC</td>
<td>IC</td>
<td>head OF Mary</td>
</tr>
<tr>
<td>FC</td>
<td>OF RC</td>
<td>RC</td>
<td>head OF friend</td>
</tr>
<tr>
<td>FC</td>
<td>OF FC</td>
<td>FC</td>
<td>head OF spouse</td>
</tr>
</tbody>
</table>

**proper relational concepts:**
- type of composed concept = relational type of possessor concept

**functional concepts:**
- type of composed concept = referential + relational type of possessor concept
\[ \text{FC} \overset{\text{OF}}{\square} \text{RC} \iff \text{RC}: \text{name OF sibling} \]

\[ \lambda y \, \lambda x. \, x = \text{NAME}(y) \quad \lambda y \, \lambda x. \, \text{MOTHER}(x) = \text{MOTHER}(y) \]
Frame graphs

$\text{FC} \sqcup \text{RC} \mapsto \text{RC}: \text{name OF sibling}$

\[
\lambda y'. \lambda x'. \ x' = f(y') \sqcup \lambda y'. \lambda x'. \ S(x', y') \mapsto \lambda y'. \lambda x'. \ x = f(\varepsilon u. \ S(u, y'))
\]

$\text{FC} \circ (\varepsilon \circ \text{RC})$

\[
\langle e, \langle e, t \rangle \rangle \circ (\langle \langle e, t \rangle, e \rangle \circ \langle e, \langle e, t \rangle \rangle) \mapsto \langle e, \langle e, t \rangle \rangle \circ \langle e, e \rangle \mapsto \langle e, \langle e, t \rangle \rangle
\]
Frame graphs

Type shifts

Composition

FC ⊔ RC \leftrightarrow RC: name OF sibling

\[ \langle e, \langle e, t \rangle \rangle \]

\[ \langle e, e \rangle \]

\[ \langle e, \langle e, t \rangle \rangle \]

\[ \lambda y' \lambda x'. x' = f(y') \quad \square \lambda y' \lambda x'. S(x', y') \leftrightarrow \lambda y' \lambda x. x = f(\varepsilon u. S(u, y')) \]

FC \circ (\varepsilon \circ RC)

\[ \langle e, \langle e, t \rangle \rangle \circ (\langle \langle e, t \rangle, e \rangle \circ \langle e, \langle e, t \rangle \rangle) \leftrightarrow \langle e, \langle e, t \rangle \rangle \circ \langle e, e \rangle \leftrightarrow \langle e, \langle e, t \rangle \rangle \]

1. \varepsilon \circ RC: \lambda y'(\lambda Q. \varepsilon u. Q(u)(\lambda x'. S(x', y')))) \rightarrow_\beta \lambda y'(\varepsilon u. \lambda x'. S(x', y')(u)) \rightarrow_\beta \lambda y'. \varepsilon u. S(u, y')

2. FC \circ (\varepsilon \circ RC): (\lambda y \lambda x. x = f(y)) \circ (\lambda y'. \varepsilon u. S(u, y')) \rightarrow \lambda y'(\lambda y \lambda x. x = f(y)(\varepsilon u. S(u, y'))) \rightarrow_\beta \lambda y' \lambda x. x = f(\varepsilon u. S(u, y'))
FC ⊔ RC ↪→ RC: name OF sibling

\[ \lambda y' \lambda x'. x' = f(y') \]  
\[ \lambda y' \lambda x'. S(x', y') \]  
\[ \lambda y' \lambda x. x = f(\varepsilon u. S(u, y')) \]

FC \circ (\varepsilon \circ RC)

1. \[ \varepsilon \circ RC: \lambda y' (\lambda Q. \varepsilon u. Q(u) (\lambda x'. S(x', y')))) \rightarrow_\beta \lambda y' (\varepsilon u. \lambda x'. S(x', y')(u)) \rightarrow_\beta \lambda y'. \varepsilon u. S(u, y') \]

2. FC \circ (\varepsilon \circ RC): (\lambda y \lambda x. x = f(y)) \circ (\lambda y'. \varepsilon u. S(u, y')) \rightarrow \lambda y' (\lambda y \lambda x. x = f(y) (\varepsilon u. S(u, y'))) \rightarrow_\beta \lambda y' \lambda x. x = f(\varepsilon u. S(u, y')) \]