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

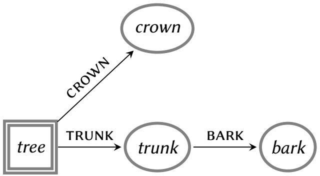
HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

SFB 991

Part 5

Nominal frames and Shifts

frames as attribute-value structures



tree-frame

$$\begin{bmatrix} \text{tree} & \\ \text{CROWN}: \text{crown} & \\ \text{TRUNK}: & \begin{bmatrix} \text{trunk} \\ \text{BARK}: \text{bark} \end{bmatrix} \end{bmatrix}$$

Given a set of types T and a set of attributes A. A basic frame over (T, A) is a tuple $\langle Q, \bar{q}, arg, \theta, \delta \rangle$ with

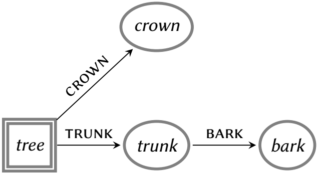
- Q is a finite set of nodes.
- $\bar{q} \in Q$ is the central node
- $arg \subseteq Q$ argument nodes.
- $ind \subseteq Q$ individual nodes.
- $\theta : Q \longrightarrow T$ is the typing function
- $\delta : Q^n \times A \longrightarrow Q$ is the partial transition function.

such that the underlying graph (Q, E) with edge set $E = \{\{q_i, q\} | \exists a \in A, q_i \in \{q_1, \ldots, q_n\} : \delta(q_1, \ldots q_n, a) = q\}$ is connected.

🕮 Carpenter 1992

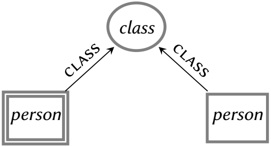
🕮 Petersen 2007

frames as attribute-value structures



tree-frame

trunk (of)-frame



classmate (of)-frame

Given a set of types T and a set of attributes A. A basic frame over (T, A) is a tuple $\langle Q, \bar{q}, arg, \theta, \delta\rangle$ with

- Q is a finite set of nodes.
- $\bar{q} \in Q$ is the central node
- $arg \subseteq Q$ argument nodes.
- $ind \subseteq Q$ individual nodes.
- $\theta : Q \longrightarrow T$ is the typing function
- $\delta : Q^n \times A \longrightarrow Q$ is the partial transition function.

such that the underlying graph (Q, E) with edge set $E = \{\{q_i, q\} | \exists a \in A, q_i \in \{q_1, \ldots, q_n\} : \delta(q_1, \ldots q_n, a) = q\}$ is connected.

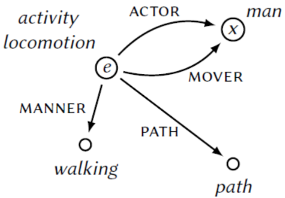
🕮 Carpenter 1992

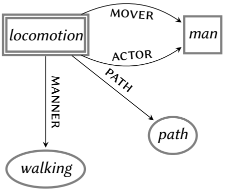
🕮 Petersen 2007

Remarks

The definition and the notation differs slightly from the one we used the last days

- frames are **connected**, directed graphs with labeled nodes (by types) and arcs (by attributes)
- argument nodes are represented as rectangular nodes
- individual nodes are marked by small pointing arrows
- the central node is marked by a double line
- the restriction that every node can be reached via directed arcs from an argument or the central node is dropped
- the types are written inside the nodes





Definition (Subsumption)

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

Definition (Subsumption)

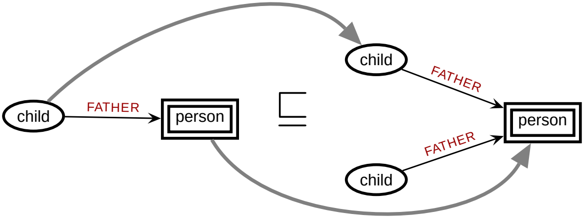
A frame $F_1 = \langle Q_1, \bar{q}_1, \theta_1, \delta_1 \rangle$ subsumes a frame $F_2 = \langle Q_2, \bar{q}_2, \theta_2, \delta_2 \rangle$ ($F \sqsubseteq F'$) iff there is a total injective function $h : Q_1 \longrightarrow Q_2$ with

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

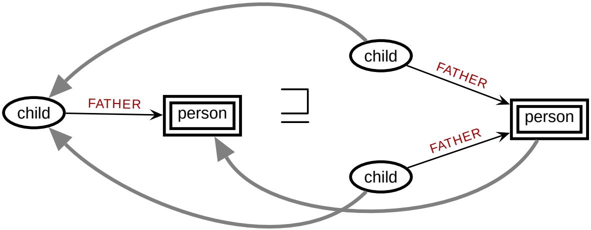
Definition (Equivalence)

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

Adaption of subsumption relation



Adaption of subsumption relation



concept classification

person, pope, house, verb, sun, Mary, wood, brother, mother, meaning, distance, spouse, argument, entrance

concept classification: relationality

non-relational	person, pope, house, verb, sun, Mary, wood
relational	brother, mother, meaning, distance, spouse, argument, entrance

concept classification: uniqueness of reference

	non-unique reference	unique reference
non-relational	person, house, verb, wood	Mary, pope, sun
relational	brother, argument, entrance	mother, meaning, distance, spouse

concept classification

	non-unique reference	unique reference
non-relational	**sortal concept** $\lambda x.\ P(x)$	**individual concept** $\lambda x.\ x = \iota u.\ P(u)$
relational	**proper relational concept** $\lambda y \lambda x.\ R(x, y)$	**functional concept** $\lambda y \lambda x.\ x = f(y)$

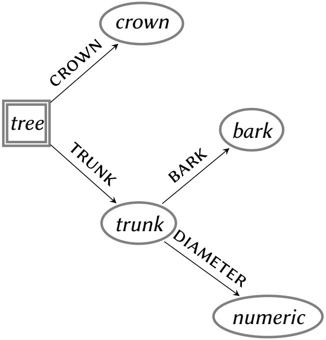
Löbner (2011): 'Concept Types and Determination' Journal of Semantics 28: 279-333

concept classification

	non-unique reference	unique reference
non-relational	**sortal concept** $\lambda x.\ P(x)$	**individual concept** $\lambda x.\ x = \iota u.\ P(u)$ $\iota u.\ P(u)$
relational	**proper relational concept** $\lambda y \lambda x.\ R(x, y)$	**functional concept** $\lambda y \lambda x.\ x = f(y)$ $\lambda y.\ f(y)$

Löbner (2011): 'Concept Types and Determination' Journal of Semantics 28: 279-333

frames and functional concepts



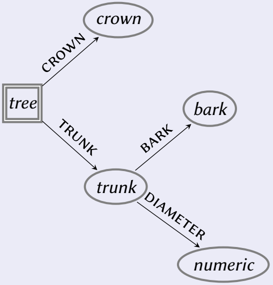
- attributes describe functional relations, i.e., they represent functions
- attributes correspond to functional concepts

⟹ frames decompose concepts into functional concepts

⟹ functional concepts embody the concept type on which categorization is based

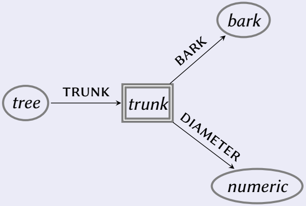
sortal concepts

tree-Frame



$\lambda x.\ \text{crown}(\textsc{crown}(x)) \wedge \text{bark}(\textsc{bark}(\textsc{trunk}(x))) \wedge \text{numeric}(\textsc{diameter}(\textsc{trunk}(x))) \wedge \ldots$

tree trunk-Frame



$\lambda x.\ \textsc{trunk}(\varepsilon u.\ x = \textsc{trunk}(u)) \wedge \text{bark}(\textsc{bark}(x)) \wedge \text{numeric}(\textsc{diameter}(x)) \wedge \ldots$

$\lambda x \exists u.\ x = \textsc{trunk}(u) \wedge \text{bark}(\textsc{bark}(x)) \wedge \text{numeric}(\textsc{diameter}(x)) \wedge \ldots$

individual concepts

Mary-frame

predicate constant ‘Mary’:

$\lambda x.\ x = \iota y.\ (y = \text{Mary})$

pope-frame

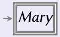
predicate constant ‘pope’:

$\lambda x.\ x = \textsc{head}(\iota y.\ y = \text{RCC})$

individual concepts

Mary-frame

predicate constant 'Mary':



$\lambda x.\ x = \iota y.\ (y = \text{Mary})$

individual constant 'Mary':

$\iota x.\ x = \text{Mary}$

pope-frame

predicate constant 'pope':

$\lambda x.\ x = \textsc{head}(\iota y.\ y = \text{RCC})$

individual constant 'pope':

$\iota x.\ x = \textsc{head}(\text{RCC})$

Summary: non-relational concepts

sortal concepts

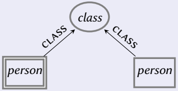
- one open argument

individual concepts

- one open argument
- there is a direct path from a definite node to the central node

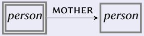
proper relational concepts

classmate-frame



$\lambda y \lambda x.\ \textsc{class}(x) = \textsc{class}(y) \wedge \ldots$

child-frame

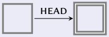


$\lambda y \lambda x.\ y = \textsc{mother}(x) \wedge \text{person}(x) \wedge \text{person}(y)$

functional concepts

head-frame

predicate constant ‘head’:



$\lambda y \lambda x.\ x = \textsc{head}(y)$

haircolor-frame

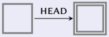
predicate constant ‘haircolor’:

$\lambda y \lambda x.\ x = \textsc{color}(\textsc{hair}(y))$

functional concepts

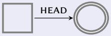
head-frame

predicate constant ‘head’:



$\lambda y \lambda x.\ x = \textsc{head}(y)$

function constant ‘head’:



$\lambda y.\ \textsc{head}(y)$

haircolor-frame

predicate constant ‘haircolor’:

$\lambda y \lambda x.\ x = \textsc{color}(\textsc{hair}(y))$

function constant ‘haircolor’:

$\lambda y.\ \textsc{color}(\textsc{hair}(y))$

Summary: relational concepts

proper relational concepts

- two open arguments
- no direct path from the other open argument to the central node

functional concepts

- two open arguments
- there is a direct path from the other open argument to the central node

Summary: concept classes and frames

sortal concepts

default frame:

$\lambda x.\ P(x)$ $\langle e, t \rangle$

one open node = central node

Examples: stone, teenager, tree

individual concepts

default frame:

$\iota u.\ P(u)$ $\langle e \rangle$

no open node

Examples: pope, Mary

proper relational concepts

default frame:

$\lambda y \lambda x.\ R(x, y)$ $\langle e, \langle e, t \rangle \rangle$

two open nodes, central node is open and not reachable from second open node

Examples: sister, son, finger

functional concepts

default frame:

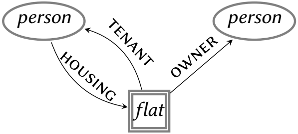
$\lambda y.\ f(y)$ $\langle e, e \rangle$

central node reachable from an open node, central node not open.

Examples: mother, trunk, color

type shifts: non-relational → relational

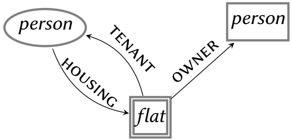
sortal	**individual**
proper relational	**functional**



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

type shifts: non-relational → relational

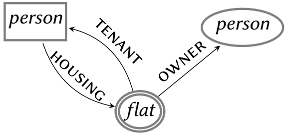
sortal	**individual**
proper relational	**functional**



proper relational concept *flat:*
"This flat is a flat of John, he owns more than five."

type shifts: non-relational → relational

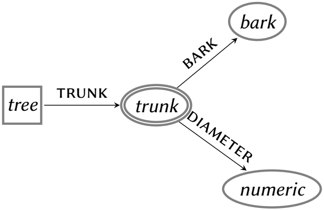
sortal	**individual**
proper relational	**functional**



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

type shifts: relational → non-relational

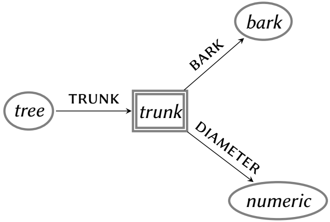
sortal	**individual**
proper relational	**functional**



functional concept *trunk:*
"She sat with her back against the trunk of an oak."

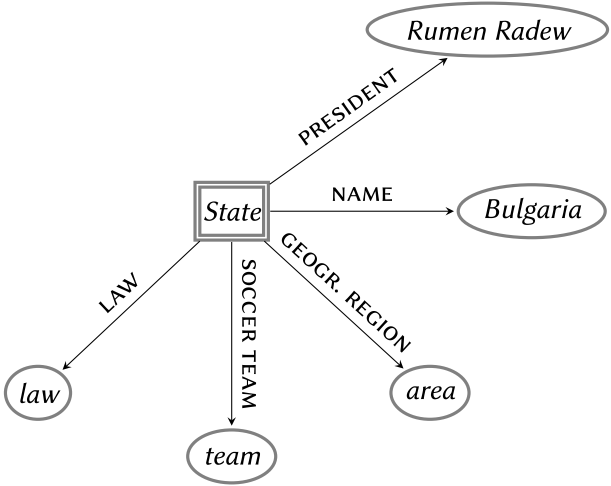
type shifts: relational → non-relational

sortal	**individual**
proper relational	**functional**

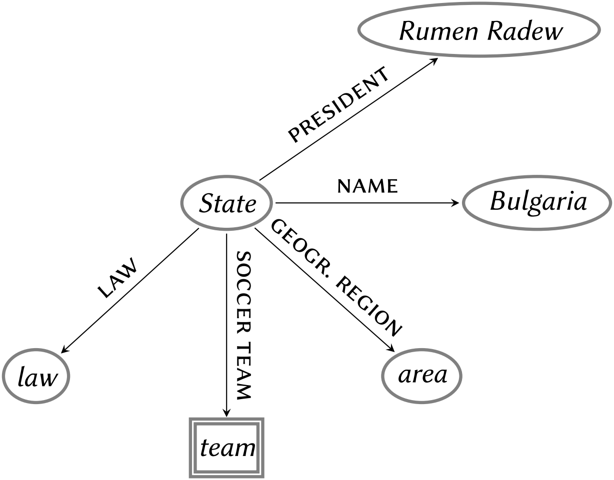


sortal concept *trunk:*
"They rested and sat on a trunk."

Metonymic Shifts

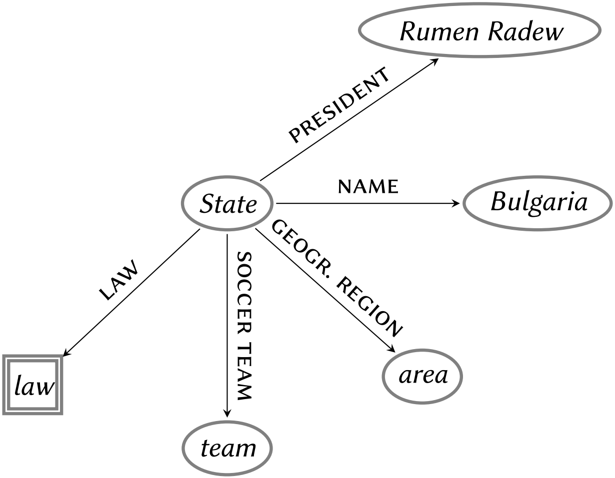


Metonymic Shifts



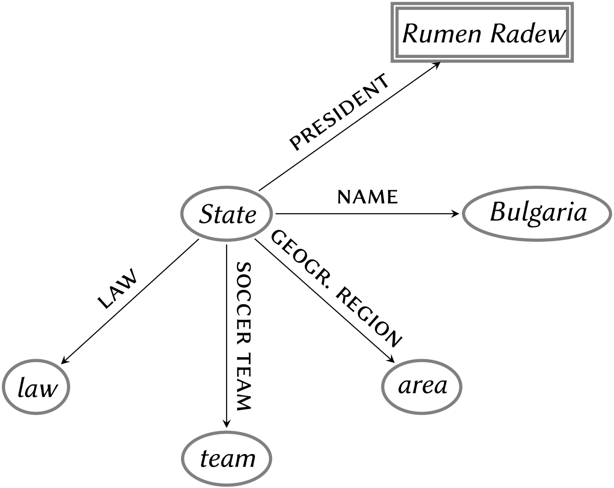
"Bulgaria won 2:3"

Metonymic Shifts



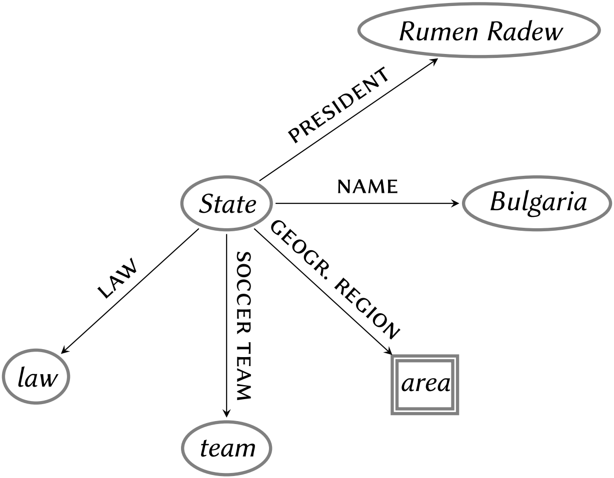
"Bulgaria allows to vote from the age of 18 on"

Metonymic Shifts



“Bulgaria speaks after Latvia”

Metonymic Shifts



“This bus goes to Bulgaria.”