Description Logics

Volker Haarslev
Concordia University, Computer Science Department
1455 de Maisonneuve Blvd. W.
Montreal, Quebec H3G 1M8, Canada

Description Logics: Introduction

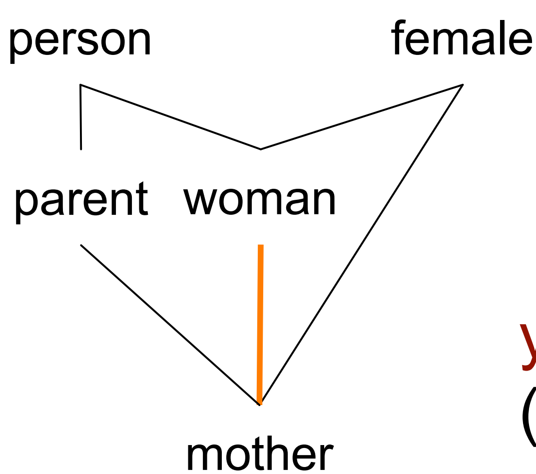
- Important syntactic feature: variable-free notation
  - constructors: $\sqcap$, $\sqcup$, $\neg$, $\exists$, $\forall$
  - standard description logic *ALC*
- Description of concept parent
  - parent $\equiv$ person $\sqcap$ $\exists$has_child.person
- We add two concepts
  - woman $\equiv$ female $\sqcap$ person
  - mother $\equiv$ female $\sqcap$ parent
- What type of inferences are interesting?
  - satisfiability of (named) concepts
  - subsumption of (named) concepts

Inference Service: Concept Satisfiability

- The concepts woman, mother, parent are satisfiable
- However, the concept $\neg$woman $\sqcap$ mother is unsatisfiable
- Why? We unfold the definition of woman and mother
  - $\neg$woman $\sqcap$ mother $\equiv$
  - $\neg$(female $\sqcap$ person) $\sqcap$ female $\sqcap$ parent $\equiv$
  - ($\neg$female $\sqcup$ $\neg$ person) $\sqcap$ female $\sqcap$ parent $\equiv$
  - ($\neg$female $\sqcup$ $\neg$ person) $\sqcap$ female $\sqcap$ parent $\equiv$
  - $\neg$person $\sqcap$ female $\sqcap$ parent $\equiv$
  - $\neg$person $\sqcap$ female $\sqcap$ person $\sqcap$ $\exists$has_child.person $\equiv$
  - $\neg$person $\sqcap$ female $\sqcap$ person $\sqcap$ $\exists$has_child.person
  - Clash
- The conjunct $\neg$woman $\sqcap$ mother can never be satisfied

Inference Service: Concept Subsumption

- Consider the question "Is a mother always a woman?"
- Does the concept woman subsume the concept mother?
- Description logic reasoners offer the computation of a subsumption hierarchy (taxonomy) of all named concepts



$\text{parent} \equiv \text{person} \sqcap \exists \text{has\_child.person}$
$\text{woman} \equiv \text{person} \sqcap \text{female}$
$\text{mother} \equiv \text{parent} \sqcap \text{female}$

yes, woman subsumes mother
(see also proof on previous slide)

Description Logics: Semantics (1)

- Translation to first-order predicate logic usually possible
- Declarative and compositional semantics preferred
- Standard Tarski-style interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

Syntax	Semantics	
A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, A is a concept name	Concepts
$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$	
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	
$\forall R.C$	$\{ x \in \Delta^{\mathcal{I}} \mid \forall y: (x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}} \}$	
$\exists R.C$	$\{ x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}} \}$	
R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, R is a role name	Roles
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	Axioms
$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$	

Description Logics: Concept Examples

- $\text{woman} \equiv \text{person} \sqcap \text{female}$
- $\text{parent} \equiv \text{person} \sqcap \exists \text{has\_child.person}$
- $\text{mother} \equiv \text{parent} \sqcap \text{female}$
- $\text{person} \sqsubseteq \forall \text{has\_child.person}$
- $\text{mother\_having\_only\_female\_kids} \equiv \text{mother} \sqcap \forall \text{has\_child.female}$
- $\text{mother\_having\_only\_daughters} \equiv \text{woman} \sqcap \text{parent} \sqcap \forall \text{has\_child.woman}$

equivalent

- $\text{grandma} \equiv \text{woman} \sqcap \exists \text{has\_child.parent}$
- $\text{great\_grandma} \equiv \text{woman} \sqcap \exists \text{has\_child.} \exists \text{has\_child.parent}$

Description Logics: Concept Examples

- $\text{woman} \equiv \text{person} \sqcap \text{female}$
- $\text{parent} \equiv \text{person} \sqcap \exists \text{has\_child.person}$
- $\text{mother} \equiv \text{parent} \sqcap \text{female}$
- $\text{person} \sqsubseteq \forall \text{has\_child.person}$
- $\text{mother\_having\_only\_female\_kids} \equiv \text{mother} \sqcap \forall \text{has\_child.female}$
- $\text{mother\_having\_only\_daughters} \equiv \text{woman} \sqcap \text{parent} \sqcap \forall \text{has\_child.woman}$

equivalent

- $\text{grandma} \equiv \text{woman} \sqcap \exists \text{has\_child.parent}$
- $\text{great\_grandma} \equiv \text{woman} \sqcap \exists \text{has\_child}.\exists \text{has\_child.parent}$

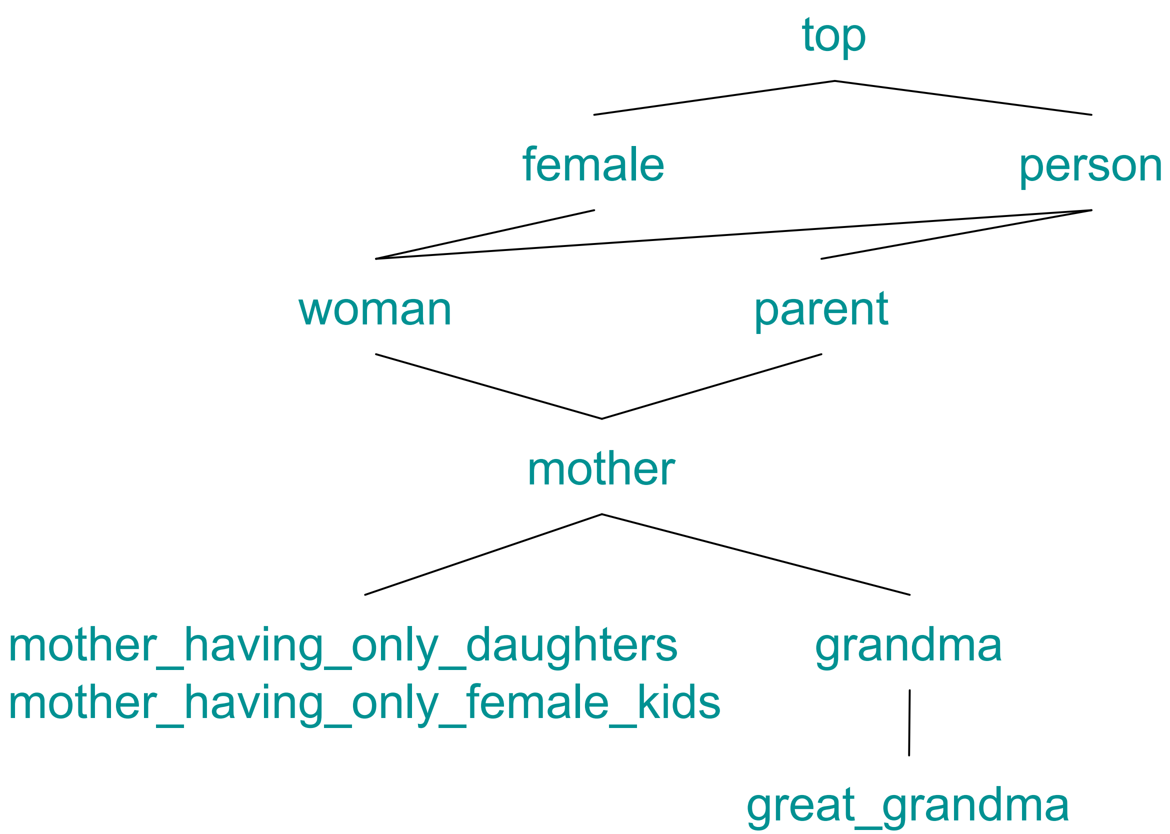
Characteristics of DL Semantics

- Interpretation domain can be chosen arbitrarily
- Distinguishing features of description logics
  - domain can be infinite
  - open world assumption
- A concept C is satisfiable iff there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$
  - $\mathcal{I}$ is called a model of C
- A concept C subsumes a concept D iff for all Interpretations $\mathcal{I}$ it holds that $D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$

Description Logics: TBox

- A collection of concept axioms is called a TBox (Terminological Box)
- Satisfiability of concepts defined w.r.t. a TBox $\mathcal{T}$
- Inference services
  - TBox coherence: List all unsatisfiable concept names in $\mathcal{T}$
  - compute subsumption hierarchy (taxonomy) of concept names in $\mathcal{T}$
- Why emphasize concept names?
  - ontological decisions of users
  - important concepts will be named

Example Taxonomy



Description Logics: Individuals

- How can we assert knowledge about individuals?
- Assertional axioms
  - concept assertion for an individual a
    - a:C satisfied iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$
    - example: elizabeth:mother
  - role assertion for two individuals a and b
    - (a,b):R satisfied iff $(a^{\mathcal{I}},b^{\mathcal{I}}) \in R^{\mathcal{I}}$
    - example: (elizabeth,charles):has_child
- Unique name assumption Not employed per default
  - Different names denote different individuals
  - $a^{\mathcal{I}} \neq b^{\mathcal{I}}$

Description Logics: ABox (1)

- A collection of assertional axioms is called an ABox (Assertional Box)
- Satisfiability of assertions defined w.r.t.
  - ABox $\mathcal{A}$
  - TBox $\mathcal{T}$
- Inference services
  - ABox satisfiability: Is the collection $\mathcal{A}$ of assertions satisfiable?
  - Instance checking: instance?(a,C,$\mathcal{A}$)
    Is a an instance of concept C or subsumes C the individual a?
  - ABox realization: compute for all individuals in $\mathcal{A}$ their most-specific concept names w.r.t. TBox $\mathcal{T}$

Description Logics: ABox (2)

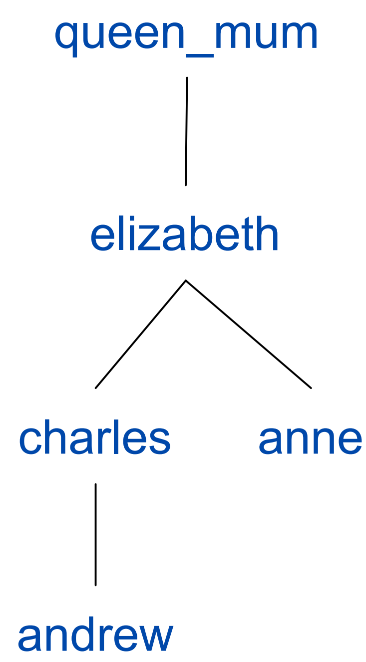
- New basic inference service: ABox satisfiability
  - asat($\mathcal{A}$)
- All other inference services can be reduced to asat
  - instance checking:
    instance?(a,C,$\mathcal{A}$) ≡ ¬asat($\mathcal{A}$ ∪ {a:¬C})
  - concept satisfiability:
    sat(C) ≡ asat({a:C})
  - concept subsumption:
    subsumes(C,D) ≡ ¬sat(¬C ⊓ D) ≡ ¬asat({a:¬C ⊓ D})
- Open world assumption
  - $\mathcal{A}$ = {andrew:male, (charles,andrew):has_child}
  - Does instance?(charles,∀has_child.male, $\mathcal{A}$) hold?

No.
Why?

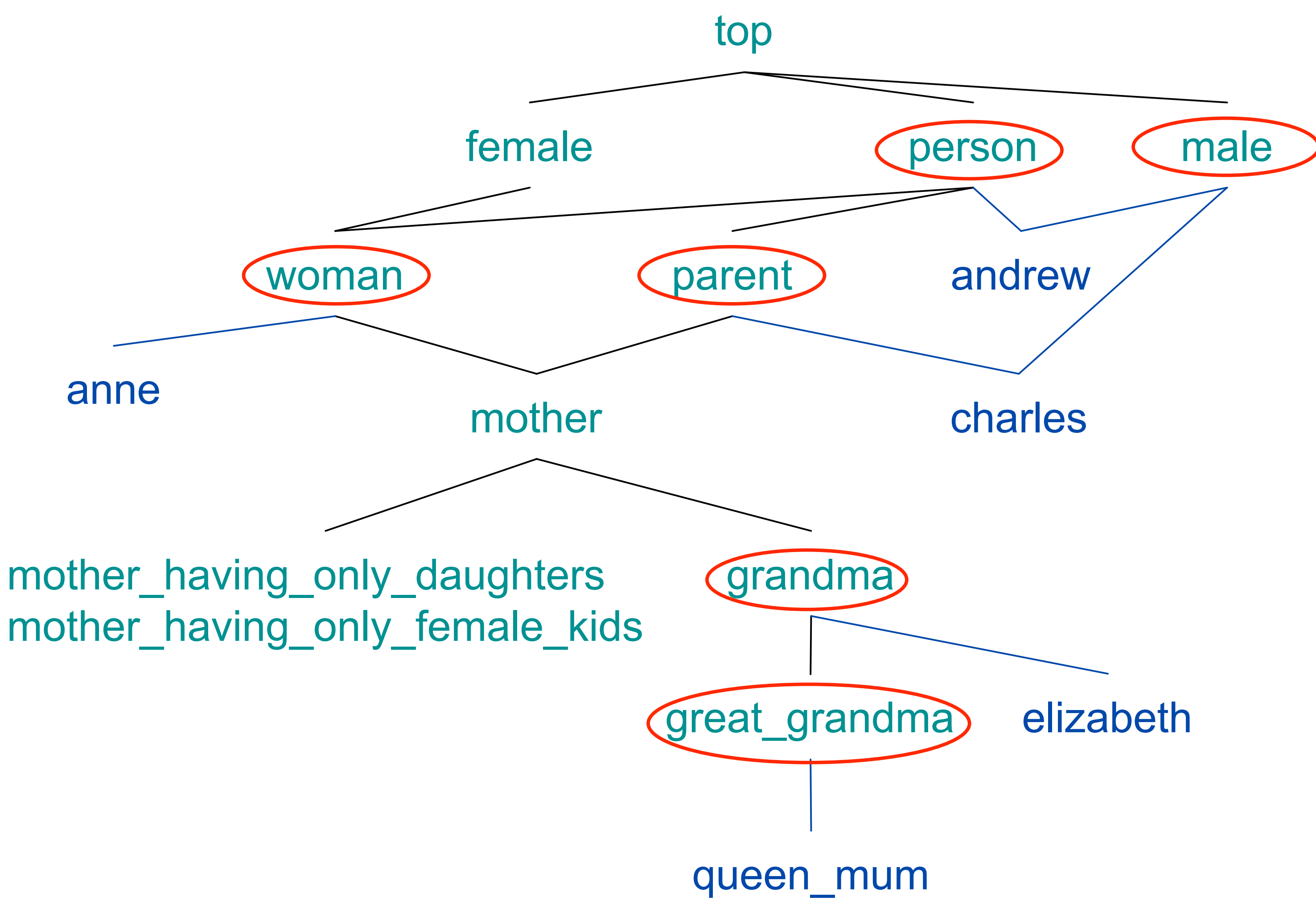
Description Logics: ABox Example

- (male ⊑ ¬female) additional axiom ensuring disjointness

- queen_mum : woman
- (queen_mum,elizabeth) : has_child
- elizabeth : woman
- (elizabeth,charles) : has_child
- (elizabeth,anne) : has_child
- charles : parent ⊓ male
- anne : woman
- (charles,andrew) : has_child
- andrew : person ⊓ male



TBox Taxonomy plus Individuals



Open World Assumption

- Can we prove that instance?(charles,∀has_child.male,$\mathcal{A}$) holds?
- No. Although the ABox contains only knowledge about one male child, it is unknown whether additional information about a female child might be added later.
- In order to prevent this, we could add
  - charles : ∀has_child.male or
  - assert that information about a second child will not be addded in the future, i.e., close a role for an individual
  - Not possible in the logic *ALC* since we need so-called number restrictions

More Description Logics Constructors

- Number restrictions on roles ($\mathcal{N}$ resp. $\mathcal{Q}$)
  - simple: $\exists_{\geq 3}$has_child or $\exists_{\leq 5}$has_child
  - qualified: $\exists_{\geq 2}$has_child.male or $\exists_{\leq 1}$has_child.female
- Role hierarchies ($\mathcal{H}$)
  - has_son $\sqsubseteq$ has_child, has_daughter $\sqsubseteq$ has_child
  - $\exists_{\geq 2}$has_son $\sqcap$ $\exists_{\geq 2}$has_daughter $\sqcap$ $\exists_{\leq 4}$has_child
- Transitive roles (R_+)
  - R declared as transitive:
    - transitive(R) $R^{\mathcal{I}} = (R^{\mathcal{I}})^+$
  - transitive(has_ancestors)
    $\forall$has_ancestors.human applies to all successors of has_ancestors
  - has_parent $\sqsubseteq$ has_ancestors demonstrates use of transitive roles in role hierarchies

More Terminological Axioms

- Inverse roles ($\mathcal{I}$):
  - $R \equiv S^-$ $\qquad (x,y) \in S^{\mathcal{I}} \Rightarrow (y,x) \in R^{\mathcal{I}}$
  - has_parent ≡ has_child$^-$
- Terminological cycles
  - human ⊑ $\exists_{\geq 2}$has_parent.human
  - binary_tree ≡ tree ⊓ $\exists_{\leq 2}$has_branch ⊓ ∀has_branch.binary_tree
- General (global) axioms
  - axioms that have
    - not a concept name on the left-hand side or
    - concept name T (thing, top) as left-hand side
  - sufficient condition for concept grandma
    woman ⊓ ∃has_child.∃has_child.person ⊑ grandma
  - domain for roles: ∃has_child.T ⊑ parent
  - range for roles: T ⊑ ∀has_child.person