5. Semantics of Web Services

- XML, Google: The syntactic web
- Web Services and WSDL
  - Classical procedure invocation based on:
    - Procedure names
    - Parameter types
    - Routing information
  - Strong coupling of subsystems required
  - In particular in ecommerce scenarios, a loose coupling can be expected
- In ecommerce scenarios, agents use different conceptual data models
- Problems with syntactic approaches to information retrieval and type-checking for calling procedures
- Enable system designers to express the meaning of the names in the conceptual data model

Description Logics: A Logical Foundation of the Semantic Web

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

http://www.cs.concordia.ca/~faculty/haarslev/
haarslev@cs.concordia.ca

Idea of the Semantic Web

Tim Berners-Lee, James Hendler, Ora Lassila: *The Semantic Web*

- World Wide Web
  - medium of
    - documents for people rather than of
    - information that can be manipulated automatically
  - augment web pages with data targeted at computers
  - add documents solely for computers
  - called semantic markup
- ...transforms into the Semantic Web
- Find meaning of semantic data by following
  - hyperlinks to definitions of key terms and
  - rules for reasoning about data logically
- Spur development of automated web services
  - highly functional agents

Typical Information Retrieval Example

- Suppose you are a salesperson, who wishes to find a Ms. Cook you met at a trade conference last year
  - you don't remember her first name but
  - you remember she worked for one of your clients and
  - her daughter is a student of your alma mater
- An intelligent search agent can
  - ignore pages relating to cooks, cookies, Cook Islands, etc.
  - find pages of companies your clients are working for
  - follow links to or find private home pages
  - check whether a daughter is still in school
  - match with students from your alma mater
- If you already have the Semantic Web

Basic Web Technology

- Uniform Resource Identifier (URI)
  - foundation of the Web
  - identify items on the Web
  - uniform resource locator (URL): special form of URI
- Extensible Markup Language (XML)
  - send documents across the Web
  - allows anyone to design own document formats (syntax)
  - can include markup to enhance meaning of document's content
  - machine readable
- Resource Description Framework (RDF)
  - make machine-processable statements
  - triple of URIs: subject, predicate, object
  - intended for information from databases

Schemas and Ontologies for the Web

- Usual assumption: data is nearly perfect
  - book rating with scale 1-10 instead of really_good,...,really _bad
  - conversion without meaning difficult
  - information newly tagged with has_author instead of creator_of
- Even worse: URIs have no meaning
- Solution: schemas and ontologies
- RDF Schemas: author is subclass of contributor
- DARPA Agent Markup Language with Web Ontology Language (OWL)
  - add semantics: has_author is the inverse relation of creator_of
  - now we understand the meaning of has_author
  - has_author(book,author) ≡ creator_of(author,book)

A Logical Foundation for the Semantic Web

- Systems can understand basic concepts such as
  - subclass
  - inverse relation, etc.
- Even better
  - state (any) logical principle
  - permit computers to reason (by inference) using these principles
  - *an employee sells more than 100 items per day* ⇒ *bonus*
  - follow semantic links to construct a proof for your conclusions
  - exchange proofs between agents (and human users)

- OWL is a syntactic variant of a well-known and very expressive description logic

Description Logics: Introduction

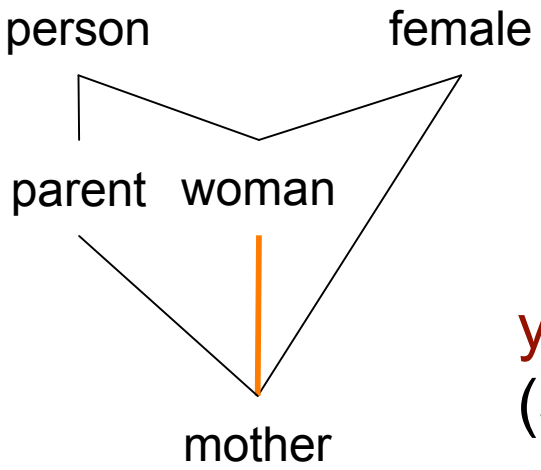
- Important syntactic feature: variable-free notation
  - constructors: $\sqcap$, $\sqcup$, $\neg$, $\exists$, $\forall$
  - standard description logic *ALC*
- Description of concept parent
  - parent $\equiv$ person $\sqcap$ $\exists$has_child.person
- We add two concepts
  - woman $\equiv$ female $\sqcap$ 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 ¬woman ⊓ mother is unsatisfiable
- Why? We unfold the definition of woman and mother
  - ¬woman ⊓ mother ≡
  - ¬(female ⊓ person) ⊓ female ⊓ parent ≡
  - (¬female ⊔ ¬ person) ⊓ female ⊓ parent ≡
  - (¬female ⊔ ¬ person) ⊓ female ⊓ parent ≡
  - ¬person ⊓ female ⊓ parent ≡
  - ¬person ⊓ female ⊓ person ⊓ ∃has_child.person ≡
  - ¬person ⊓ female ⊓ person ⊓ ∃has_child.person
  - 7
- The conjunct ¬woman ⊓ 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



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

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

- woman ≡ person ⊓ female
- parent ≡ person ⊓
  ∃has_child.person
- mother ≡ parent ⊓ female
- mother_having_only_female_kids ≡ mother ⊓
  ∀has_child.female
- mother_having_only_daughters ≡ woman ⊓
  parent ⊓
  ∀has_child.woman

equivalent

- grandma ≡ woman ⊓ ∃has_child.parent
- great_grandma ≡ woman ⊓
  ∃has_child.∃has_child.parent

Description Logics: Concept Examples

- woman ≡ person ⊓ female
- parent ≡ person ⊓
  ∃has_child.person
- mother ≡ parent ⊓ female
- mother_having_only_female_kids ≡ mother ⊓
  ∀has_child.female
- mother_having_only_daughters ≡ woman ⊓
  parent ⊓
  ∀has_child.woman

} equivalent

- grandma ≡ woman ⊓ ∃has_child.parent
- great_grandma ≡ woman ⊓
  ∃has_child.∃has_child.parent

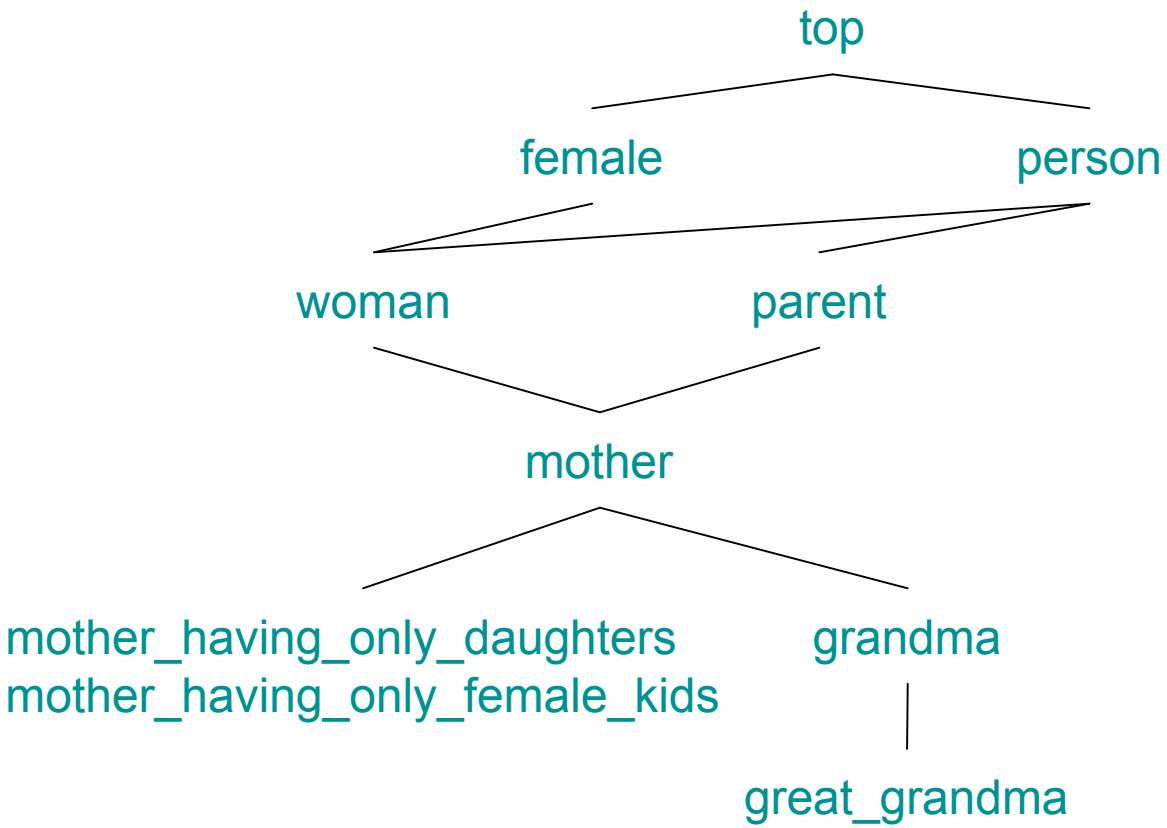
Description Logics: Semantics (2)

- 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
- Subsumption can be reduced to satisfiability
  - subsumes(C,D) $\Leftrightarrow \neg sat(\neg C \sqcap D)$
  - denoted as $C \sqsupseteq D$ or $D \sqsubseteq C$

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
  - 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?

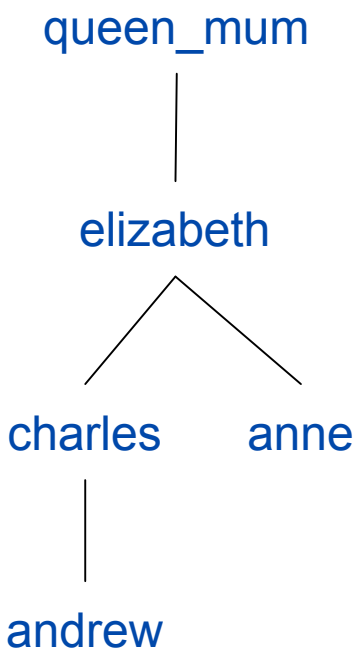
No.
Why?

Description Logics: ABox Example

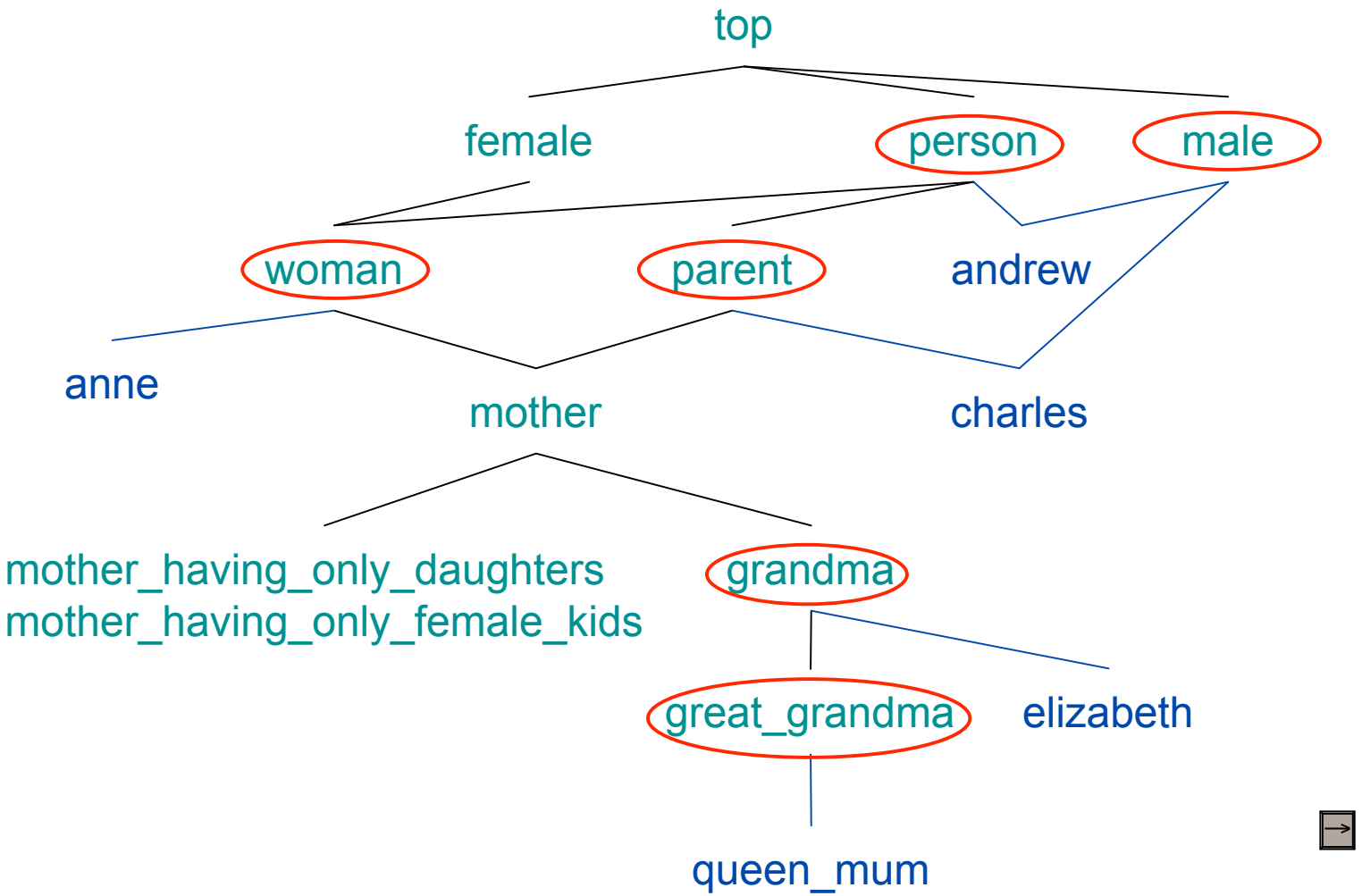
- (male $\sqsubseteq \neg$female)

additional axiom ensuring disjointness

- queen_mum : woman
- (queen_mum,elizabeth) : has_child
- elizabeth : woman
- (elizabeth,charles) : has_child
- (elizabeth,anne) : has_child
- charles : parent $\sqcap$ male
- anne : woman
- (charles,andrew) : has_child
- andrew : person $\sqcap$ 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 ($\mathcal{I}$):
  - $R \equiv S^-$ $\qquad (x,y) \in S^{\mathcal{I}} \Rightarrow (y,x) \in R^{\mathcal{I}}$
  - $has\_parent \equiv has\_child^-$
- Terminological cycles
  - $human \sqsubseteq \exists_{\geq 2} has\_parent.human$
  - $binary\_tree \equiv tree \sqcap \exists_{\leq 2} has\_branch \sqcap \forall has\_branch.binary\_tree$
- 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
    $woman \sqcap \exists has\_child.\exists has\_child.person \sqsubseteq grandma$
  - domain for roles: $\exists has\_child.T \sqsubseteq parent$
  - range for roles: $T \sqsubseteq \forall has\_child.person$