

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

Idea of the Semantic Web



■ World Wide Web

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



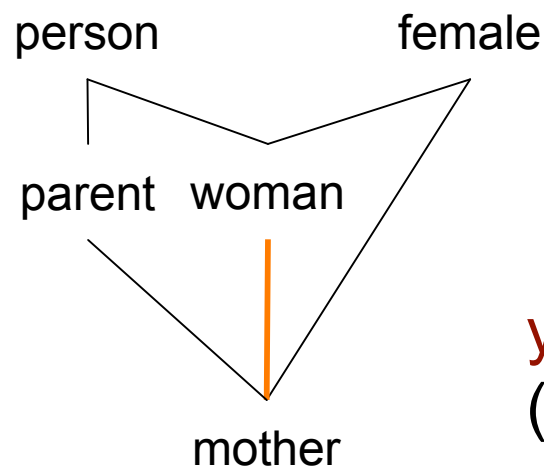
- Important syntactic feature: variable-free notation
 - constructors: \sqcap , \sqcup , \neg , \exists , \forall
 - standard description logic *ALC*
- Description of concept **parent**
 - $\text{parent} \equiv \text{person} \sqcap \exists \text{has_child}.\text{person}$
- We add two concepts
 - $\text{woman} \equiv \text{female} \sqcap \text{person}$
 - $\text{mother} \equiv \text{female} \sqcap \text{parent}$
- 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**
 - 7
- 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



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

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

A

$\neg C$

$C \sqcap D$

$C \sqcup D$

$\forall R.C$

$\exists R.C$

R

$C \sqsubseteq D$

$C \equiv D$

Semantics

$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, A is a concept name

$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$

$C^{\mathcal{I}} \cap D^{\mathcal{I}}$

$C^{\mathcal{I}} \cup D^{\mathcal{I}}$

$\{x \in \Delta^{\mathcal{I}} \mid \forall y: (x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$

$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$

$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, R is a role name

$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

$C^{\mathcal{I}} = D^{\mathcal{I}}$

Concepts

Roles

Axioms



Description Logics: Concept Examples

- $\text{woman} \equiv \text{person} \sqcap \text{female}$
 - $\text{parent} \equiv \text{person} \sqcap \exists \text{has_child.person}$
 - $\text{mother} \equiv \text{parent} \sqcap \text{female}$
 - $\text{mother_having_only_female_kids} \equiv \text{mother} \sqcap \forall \text{has_child.female}$
 - $\text{mother_having_only_daughters} \equiv \text{woman} \sqcap \text{parent} \sqcap \forall \text{has_child.woman}$
- } 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

- $\text{woman} \equiv \text{person} \sqcap \text{female}$
 - $\text{parent} \equiv \text{person} \sqcap \exists \text{has_child.person}$
 - $\text{mother} \equiv \text{parent} \sqcap \text{female}$
 - $\text{mother_having_only_female_kids} \equiv \text{mother} \sqcap \forall \text{has_child.female}$
 - $\text{mother_having_only_daughters} \equiv \text{woman} \sqcap \text{parent} \sqcap \forall \text{has_child.woman}$
- } 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: Semantics (2)



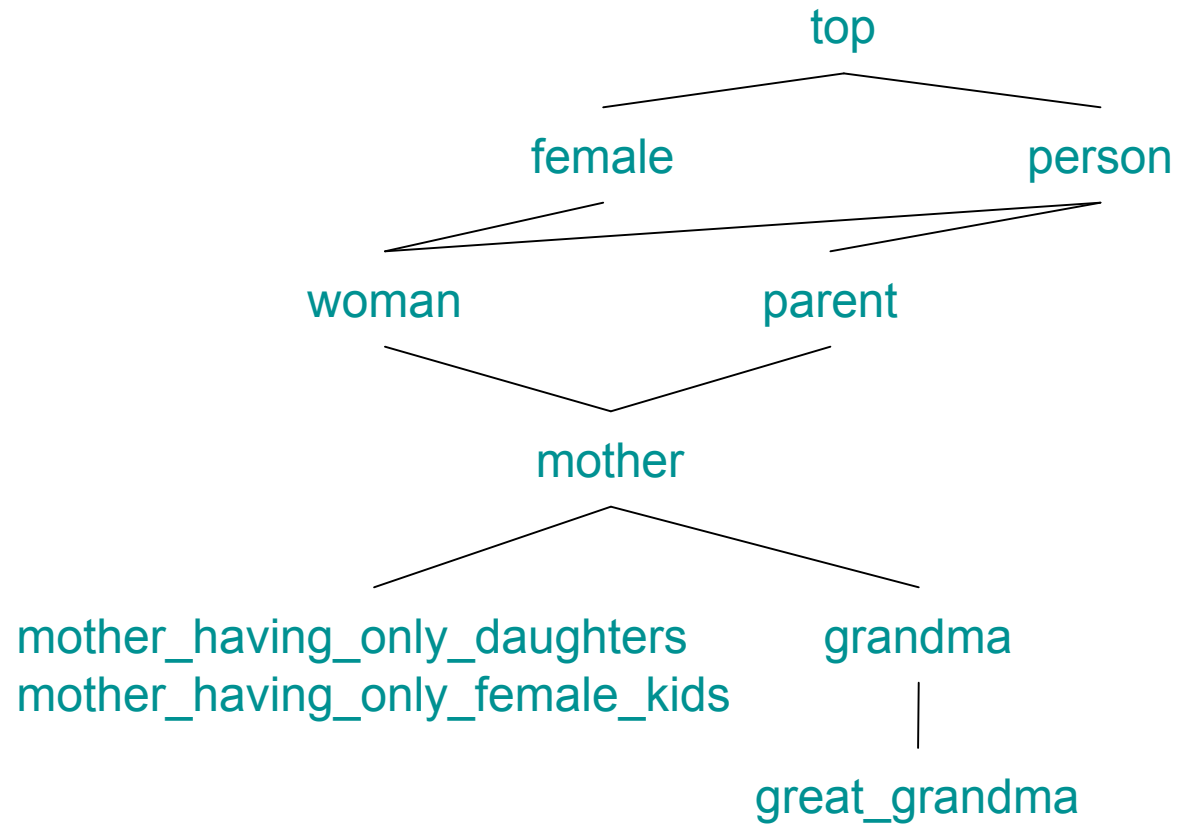
- Interpretation domain can be chosen arbitrarily
- Distinguishing features of description logics
 - domain can be **infinite**
 - **open world assumption**
- A concept **C** is satisfiable iff there exists an interpretation \mathcal{I} such that $C^{\mathcal{I}} \neq \emptyset$
 - \mathcal{I} is called a **model** of **C**
- Subsumption can be reduced to satisfiability
 - $\text{subsumes}(C,D) \Leftrightarrow \neg \text{sat}(\neg C \sqcap D)$
 - denoted as $C \sqsupseteq D$ or $D \sqsubseteq C$

Description Logics: TBox



- A collection of concept axioms is called a **TBox** (**T**erminological **B**ox)
- 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 $a^I \in C^I$
 - example: `elizabeth:mother`
 - role assertion for two individuals a and b
 - $(a,b):R$ satisfied iff $(a^I, b^I) \in R^I$
 - example: `(elizabeth,charles):has_child`
- Unique name assumption
 - Different names denote different individuals
 - $a^I \neq b^I$

Description Logics: ABox (1)

- A collection of assertional axioms is called an **ABox** (**A**ssertional **B**ox)
- 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**: $\text{instance?}(a, C, \mathcal{A})$
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
 - $\text{asat}(\mathcal{A})$
- All other inference services can be reduced to asat
 - instance checking:
 $\text{instance?}(a, C, \mathcal{A}) \equiv \neg \text{asat}(\mathcal{A} \cup \{a: \neg C\})$
 - concept satisfiability:
 $\text{sat}(C) \equiv \text{asat}(\{a: C\})$
 - concept subsumption:
 $\text{subsumes}(C, D) \equiv \neg \text{sat}(\neg C \sqcap D) \equiv \neg \text{asat}(\{a: \neg C \sqcap D\})$
- Open world assumption
 - $\mathcal{A} = \{\text{andrew: male}, (\text{charles}, \text{andrew}): \text{has_child}\}$
 - Does $\text{instance?}(\text{charles}, \forall \text{has_child. male}, \mathcal{A})$ hold?

No.
Why?

Description Logics: ABox Example

■ $(\text{male} \sqsubseteq \neg \text{female})$ additional axiom ensuring disjointness

■ $\text{queen_mum} : \text{woman}$

■ $(\text{queen_mum}, \text{elizabeth}) : \text{has_child}$

■ $\text{elizabeth} : \text{woman}$

■ $(\text{elizabeth}, \text{charles}) : \text{has_child}$

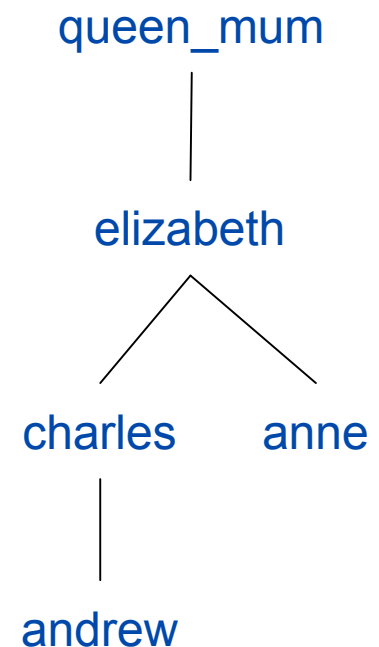
■ $(\text{elizabeth}, \text{anne}) : \text{has_child}$

■ $\text{charles} : \text{parent} \sqcap \text{male}$

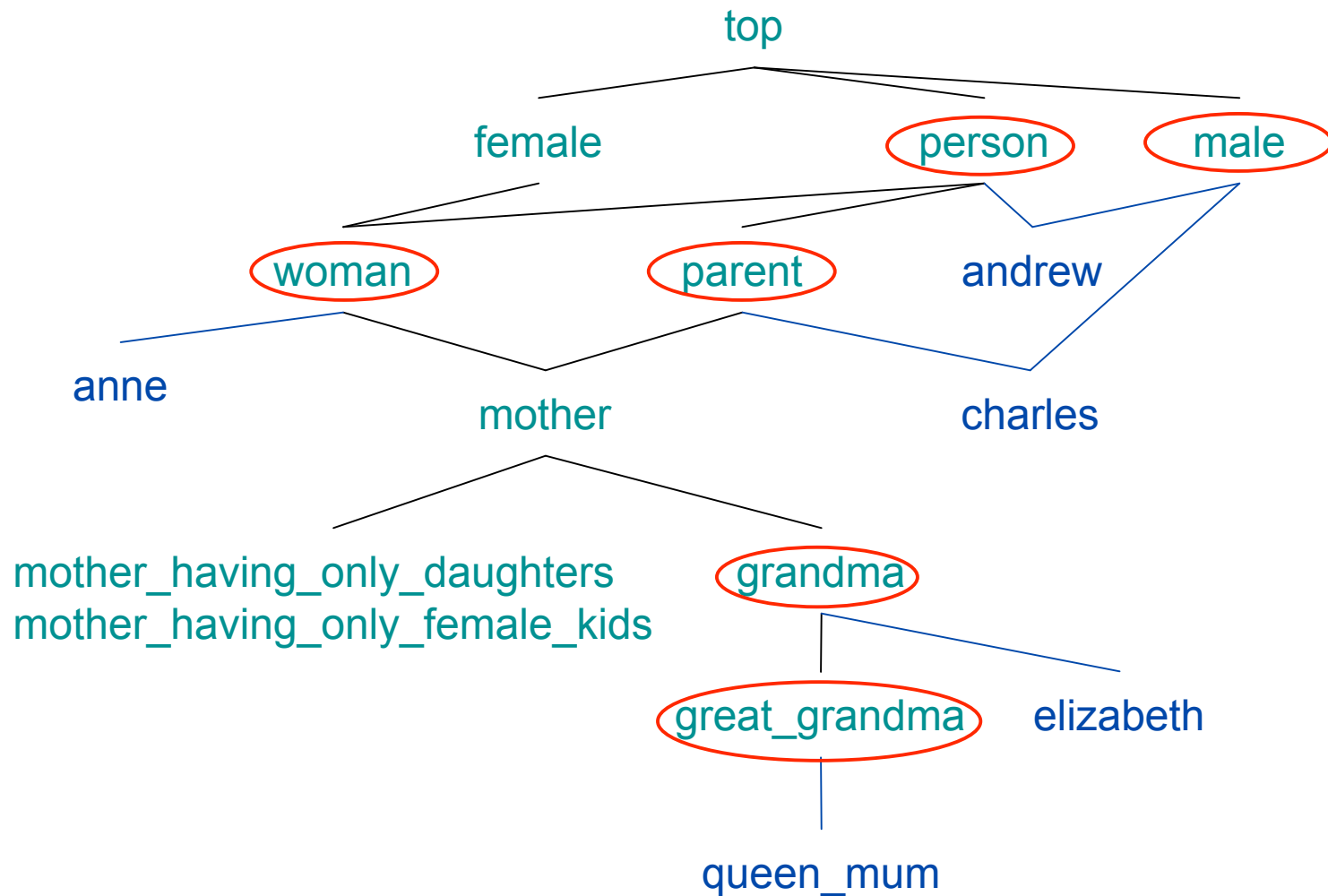
■ $\text{anne} : \text{woman}$

■ $(\text{charles}, \text{andrew}) : \text{has_child}$

■ $\text{andrew} : \text{person} \sqcap \text{male}$



TBox Taxonomy plus Individuals



Open World Assumption



- Can we prove that `instance?(charles, ∀has_child.male, 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 added 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})
 - $\text{has_son} \sqsubseteq \text{has_child}$, $\text{has_daughter} \sqsubseteq \text{has_child}$
 - $\exists_{\geq 2}\text{has_son} \sqcap \exists_{\geq 2}\text{has_daughter} \sqcap \exists_{\leq 4}\text{has_child}$
- Transitive roles (\mathcal{R}_+)
 - R declared as transitive:
 - $\text{transitive}(R)$ $R^{\dagger} = (R^{\dagger})^+$
 - $\text{transitive}(\text{has_ancestors})$
 $\forall \text{has_ancestors.human}$ applies to all successors of has_ancestors
 - $\text{has_parent} \sqsubseteq \text{has_ancestors}$ demonstrates use of transitive roles in role hierarchies



More Terminological Axioms

- Inverse roles (\mathcal{I}):
 - $R \equiv S^-$ $(x,y) \in S^{\mathcal{I}} \Rightarrow (y,x) \in R^{\mathcal{I}}$
 - $\text{has_parent} \equiv \text{has_child}^-$
- Terminological cycles
 - $\text{human} \sqsubseteq \exists_{\geq 2} \text{has_parent.human}$
 - $\text{binary_tree} \equiv \text{tree} \sqcap \exists_{\leq 2} \text{has_branch} \sqcap \forall \text{has_branch.binary_tree}$
- 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**
 $\text{woman} \sqcap \exists \text{has_child}.\exists \text{has_child.person} \sqsubseteq \text{grandma}$
 - **domain** for roles: $\exists \text{has_child.T} \sqsubseteq \text{parent}$
 - **range** for roles: $T \sqsubseteq \forall \text{has_child.person}$