Ontology Languages

Introduction to Description Logics

Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

Objective

Explain the basic idea of description logics

Objective

Explain the syntax of the simple description logic ALC

Credits

These slides are adapted from the slides by Manolis Koubarakis posted at
http://cgi.di.uoa.gr/~pms509/lectures/dl-intro.pdf

What are Description Logics?

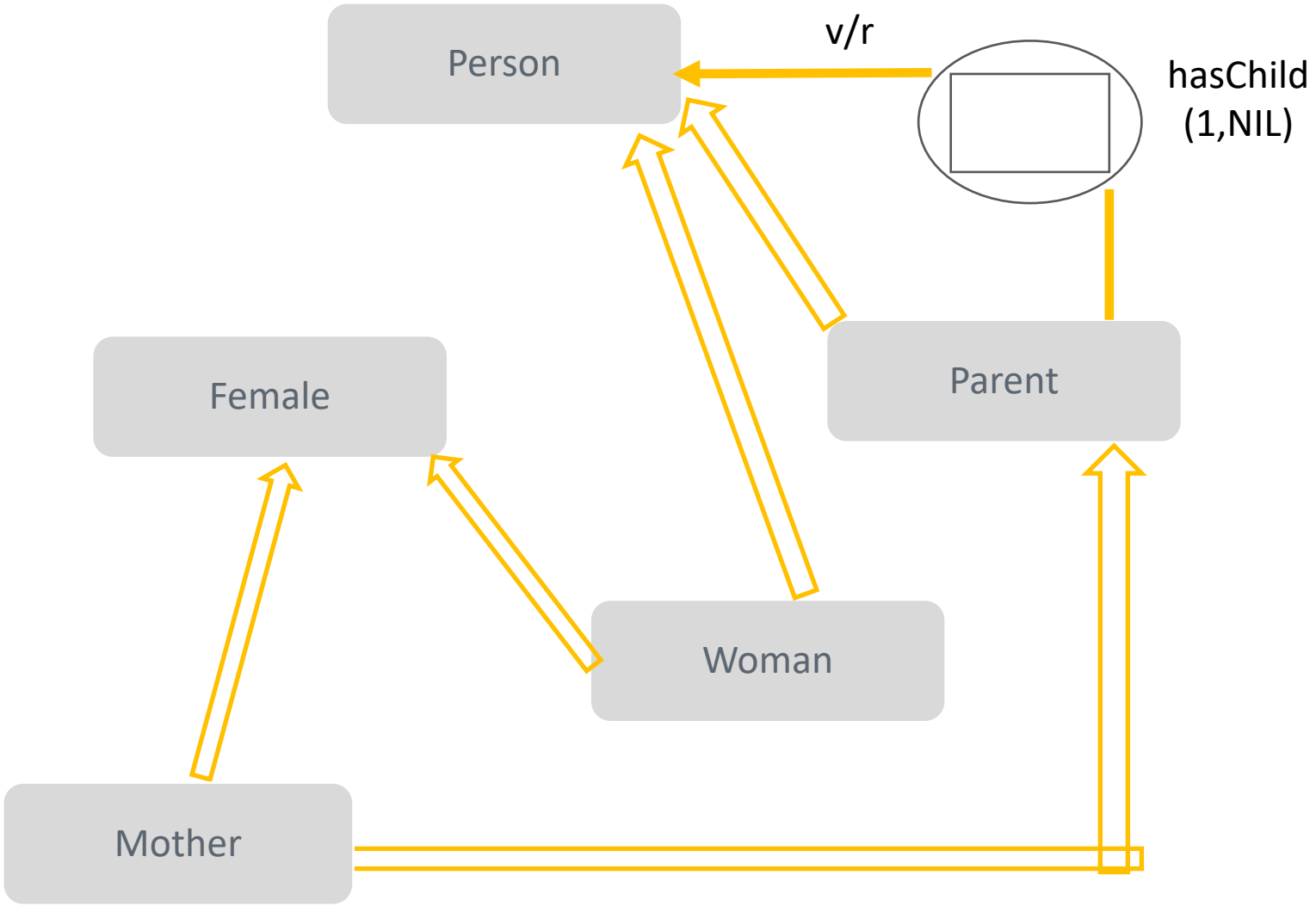
Description logics (DLs) are a family of knowledge representation languages that can be used to represent knowledge of an application domain in a structured and well-understood way

- Descendants of semantic networks and KL-ONE from the 1960-70s
- Describe a domain of interest in terms of
  - Concepts (also called “classes”)
  - Roles (also called relations or properties)
  - Individuals
- Essentially a decidable fragment of FOL

DLs: Some History

- Approach's origins lie in research on semantic networks and frames
- Developed in the 80s and 90s in parallel with pure FOL approaches and other languages for structured objects (e.g., Telos and F-logic)
- Also known as terminological languages or concept languages
- See http://www.dl.kr.org/

An Example KL-One Network



Applications of DLs

- Conceptual Modelling
- Data Integration
- Configuration
- Software Engineering
- Medical Informatics
- Bioinformatics
- Natural Language Processing
- Knowledge Representation and Reasoning in the Semantic Web

ALC: Simple DL

ALC Syntax

To define the syntax of ALC, we start with the following three disjoint alphabets:

- Concept names
- Role names
- Individual names

Concept names and role names are also called atomic concepts and atomic roles

ALC Syntax: Concepts

The set of concept expressions or just concepts is defined inductively as follows:

- Every concept name is a concept.
- $\top$ (top concept) and $\bot$ (bottom concept) are concepts.
- If C and D are concepts and R is a role name then the following are concepts:
  - $\neg C$ (complement of C)
  - $C \sqcap D$ (conjunction of C and D)
  - $C \sqcup D$ (disjunction of C and D)
  - $\forall R.C$ (universal restriction)
  - $\exists R.C$ (existential restriction)

Ex: $\forall \mathbf{hasChild.Male}$

$\exists \mathbf{hasChild.Male}$

ALC Syntax: Terminological Axioms (1 of 2)

Let A be a concept name and C, D be concepts

A terminological axiom is a statement in any of the following forms:

- Concept definitions: $A \equiv D$ which is read “A is defined to be equivalent to D”
- Concept inclusions: $C \sqsubseteq D$ which is read “C is subsumed by D”

ALC Syntax: Terminological Axioms (2 of 2)

$$\mathbf{Woman} \equiv \mathbf{Person} \sqcap \mathbf{Female}$$

$$\mathbf{Mother} \equiv \mathbf{Woman} \sqcap \exists \mathbf{hasChild}.\,\mathbf{Person}$$

$$\mathbf{Student} \sqsubseteq \mathbf{Person}$$

$$\mathbf{Student} \sqsubseteq \exists \mathbf{enrolled}.\,\mathbf{Course}$$

Intuitive Meaning of Concept Definitions

- **Concept definitions are used to introduce new symbolic names for complex concept descriptions**

- **Distinguish between name symbols that occur in the left-hand side of a definition and base symbols that occur only on the right-hand side of some axioms**
  - Defined concepts: name symbols appearing in concept definitions
  - Primitive concepts: name symbols appearing in base symbols

Primitive vs. Defined Concepts

Defined Concepts

- **Have necessary and sufficient conditions for concept membership**
- **Examples: woman, mother, driver, white wine**

Primitive Concepts

- **Cannot be defined or need not be defined**
  - However, we might know some necessary (but not sufficient) conditions for membership
- **Examples: dog** (or any other natural kind), **wine** (in a food and wine recommendation application)

Necessary Conditions

A concept inclusion of the form $C \sqsubseteq D$ states a necessary condition for membership in the concept C:

- For an individual to be in C, it is necessary that it is also in D (it has the properties expressed by D)

Ex: $\text{Student} \sqsubseteq \exists \text{enrolled.Course}$

Concept inclusions express "if" statements

Necessary and Sufficient Conditions

- A concept equivalence (definition) of the form C ≡ D states a necessary and sufficient condition for membership in the concept C:

- **Example:** $\text{Mother} \equiv \text{Woman} \sqcap \exists \text{hasChild}.\text{Person}$

- Concept equivalences express “if and only if” statements

Example: Family Relationships

$$\mathbf{Woman} \equiv \mathbf{Person} \sqcap \mathbf{Female}$$

$$\mathbf{Man} \equiv \mathbf{Person} \sqcap \neg\mathbf{Woman}$$

$$\mathbf{Mother} \equiv \mathbf{Woman} \sqcap \exists \mathbf{hasChild}.\, \mathbf{Person}$$

$$\mathbf{Father} \equiv \mathbf{Man} \sqcap \exists \mathbf{hasChild}.\, \mathbf{Person}$$

$$\mathbf{Parent} \equiv \mathbf{Mother} \sqcup \mathbf{Father}$$

$$\mathbf{Grandmother} \equiv \mathbf{Mother} \sqcap \exists \mathbf{hasChild}.\, \mathbf{Parent}$$

$$\mathbf{MotherWithoutDaughter} \equiv \mathbf{Mother} \sqcap \forall \mathbf{hasChild}.\, \neg\mathbf{Woman}$$

$$\mathbf{Wife} \equiv \mathbf{Woman} \sqcap \exists \mathbf{hasHusband}.\, \mathbf{Man}$$

Examples of Concept Inclusions

- **Disjointness of concepts:** $\text{Male} \sqsubseteq \neg\text{Female}$ Female ⊑ ¬Male
- **Coverings:** $\top \sqsubseteq \text{Male} \sqcup \text{Female}$
- **Domain restrictions:** $\exists \text{hasChild}.\top \sqsubseteq \text{Parent}$
- **Range restrictions:** $\top \sqsubseteq \forall \text{hasChild}.\text{Person}$

ALC Syntax: Assertions about Individuals

In ALC, one can also describe a specific state of affairs of an application domain in terms of individuals, concepts and roles. This is done by:

- Concept assertions: Statements of the form C(a) where C is a concept and a is an individual.
- Role assertions: Statements of the form R(a, b) where R is a role and a, b are individuals.

$$\mathbf{Student}\,(\mathbf{John})$$

$$\mathbf{enrolled}\,(\mathbf{John}, \mathbf{CS415})$$

$$(\mathbf{Student} \sqcup \mathbf{Professor})(\mathbf{PAUL})$$

TBoxes, ABoxes and Knowledge Bases

- A TBox is a set of terminological axioms.
- An Abox is a set of concept and role assertions.
- A knowledge base K is a pair (T , A) where T is a TBox and A is an Abox.

Wrap-Up