Introduction to Description Logics

Outline

- **History of DLs**
- Informal introduction to DLs
- A Simple DL: $\mathcal{ALC}$
- Semantics of $\mathcal{ALC}$
- Reasoning problems in $\mathcal{ALC}$
- The DLs $\mathcal{ALCN}$, $\mathcal{ALCQ}$ and $\mathcal{ALCQO}$
- Translating DLs to FOL

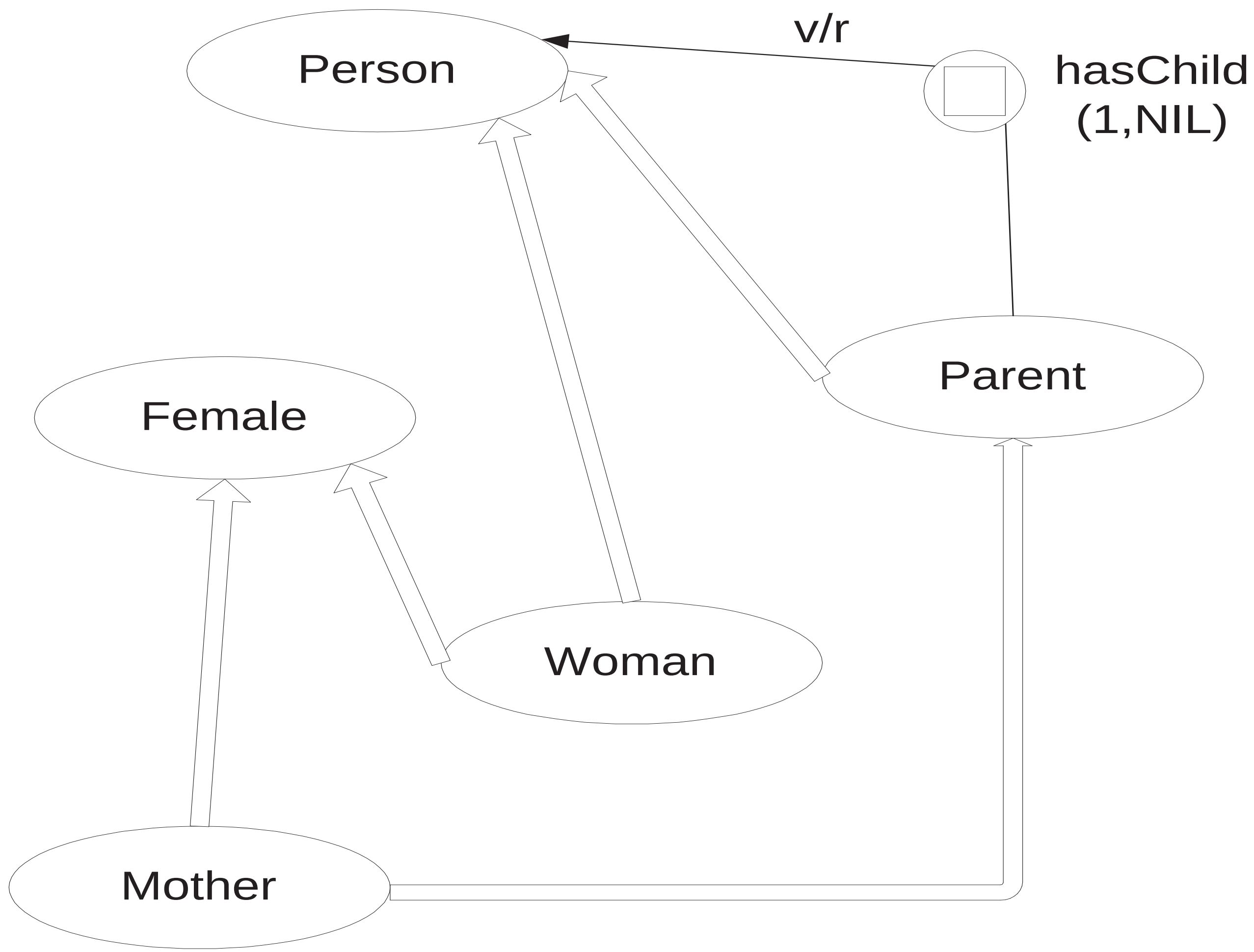
DLs: Some History

- The origins of DLs lie in research on **semantic networks** and **frames**. DLs are languages for describing the **nature and structure of objects.**
- The DL approach to KR was developed in the 80's and 90's in parallel with pure FOL approaches and other languages for structured objects like Telos and F-logic. Recently, DLs have been used to provide the **foundations for ontology languages** for the Web e.g., OWL.
- DLs are logics based on **descriptions** of **concepts** or **terms.** They are also known as **terminological languages** or **concept languages.**

DLs: Some History (cont'd)

- It all started with work on KL-ONE by Ron Brachman and colleagues. KL-ONE is the root of the family of DLs.
- There is currently a great body of theoretical work in DLs and many implemented DL systems (see the site `http://www.dl.kr.org`).

An Example of a Kl-One Network



Outline

- History of DLs
- **Informal introduction to DLs**
- A Simple DL: $\mathcal{ALC}$
- Semantics of $\mathcal{ALC}$
- Reasoning problems in $\mathcal{ALC}$
- The DLs $\mathcal{ALCN}$, $\mathcal{ALCQ}$ and $\mathcal{ALCQO}$
- Translating DLs to FOL

DLs are Logics for Knowledge Representation

For each DL of interest, we will define:

- Syntax
- Semantics
- Reasoning (inference, proof-theory)

We will use DLs to represent knowledge about a domain of interest.

Syntax

- Three disjoint alphabets of symbols:
  - **Concept names**
  - **Role names**
  - **Individual names**

Concept names are the equivalent of **class names** in other languages.

Role names are the equivalent of **property** or **relationship names** in other languages.

Individual names are the equivalent of **object names** in other languages.

Syntax (cont'd)

- **Complex concepts** and **roles** are built from symbols of the above 3 alphabets using **constructors**:
  - **conjunction, disjunction** and **negation of concepts**
  - **value restrictions**
  - **number restrictions**
  - ...
- **Sentences** that express knowledge about a domain are made by relating concepts to each other, asserting that an individual belongs to a concept and relating two individuals via a role.

Examples - Syntax

- Concept names: `Person`, `Male`, `Female`, `Doctor`, `GreekUniversity`
- Role names: `hasChild`, `isAlumniOf`
- Individual names: `ANNA`, `JOHN`

Examples - Syntax (cont'd)

Complex concepts:

- $\texttt{Person} \sqcap \neg\texttt{Female}$
- $\texttt{Female} \sqcup \texttt{Male}$
- $\forall\texttt{hasChild.Male}$
- $\exists\texttt{hasChild.Male}$
- $\exists\texttt{hasChild.Male} \sqcap \forall\texttt{hasChild.Person}$
- $(\geqslant 3\ \texttt{hasChild})$
- $(\geqslant 3\ \texttt{hasChild.Male})$
- $(\geqslant 3\ \texttt{hasChild}) \sqcap \texttt{Male}$

Note: The above constructors can be nicely read as: not, and, or, all, some, at least etc.

Examples - Syntax (cont'd)

- $\exists$hasChild.($\exists$hasChild.Person)
- $\forall$hasChild.($\exists$isAlumniOf.GreekUniversity)
- $\forall$hasChild.(Doctor $\sqcap$ $\exists$isAlumniOf.GreekUniversity)
- ($\geqslant$3 hasChild.($\forall$isAlumniOf.GreekUniversity))
- ($\geqslant$3 hasChild.(Doctor $\sqcap$ $\forall$isAlumniOf.GreekUniversity))
- Female$\sqcap$($\geqslant$3 hasChild.(Doctor$\sqcap$$\forall$isAlumniOf.GreekUniversity))

Syntactic Conventions

- Individual names are written in uppercase.
- Concept names start with an uppercase letter followed by a lowercase letter.
- Role names start with a lowercase letter.

Knowledge Representation with DLs

In DLs we make a clear distinction between **intensional knowledge** and **extensional knowledge**.

A **knowledge base (KB)** consists of two components: a **TBox** and an **ABox**.

- **TBox: intensional knowledge** in the form of sentences relating concepts (terms) to other concepts. In other frameworks, this is called **schema knowledge**.
- **ABox: extensional (assertional) knowledge**. In other frameworks, this is called **instance knowledge**.

TBox

In the TBox one defines concepts of the application domain, their properties and their relations to each other:

Examples:

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

$$\texttt{GreekUniversityAlumni} \equiv \texttt{Person} \sqcap \exists \texttt{isAlumniOf.GreekUniversity}$$

$$\texttt{Man} \sqsubseteq \texttt{Person}$$

$$\texttt{Man} \sqsubseteq \neg\texttt{Woman}$$

$$\texttt{Male} \sqsubseteq \neg\texttt{Female}$$

ABox

In the ABox one makes assertions about the individuals in the application domain: **membership in classes** and **role filling.**

Examples:

$$\texttt{Female}(\texttt{ANNA}),\quad \texttt{hasChild}(\texttt{ANNA}, \texttt{JOHN}),$$

$$(\texttt{Person} \sqcap \neg\texttt{Male})(\texttt{ANNA}),\quad ((\geqslant 3\ \texttt{hasChild}) \sqcap \texttt{Male})(\texttt{JOHN}),$$

$$\texttt{GreekUniversityAlumni}(\texttt{JOHN}),$$

$$(\geqslant 3\ \texttt{hasChild}.(\texttt{Person} \sqcap \forall \texttt{isAlumniOf}.\texttt{GreekUniversity}))(\texttt{ANNA})$$

Semantics

DL expressions and sentences can be used to represent knowledge about some domain of interest.

The semantics of such DL expressions and sentences are given by introducing the notion of **interpretation**:

- An interpretation has a **domain**.
- Individual names are interpreted as **elements of the domain.**
- Concept names are interpreted as **subsets** of the domain.
- Role names are interpreted as **binary relations** over the domain.

Semantics (cont'd)

- The semantics of complex DL concepts is defined by appropriate **set expressions** which refer to sets that give the semantics of the parts of these expressions (e.g., the semantics of conjunction is defined by set intersection).
- The semantics of TBox or ABox sentences is defined by **set theory** (set membership, inclusion, equality, disjointness etc.).

Examples - Informal Semantics

- `Male`

  The set of male individuals.

- `hasChild`

  The set of pairs of individuals (x, y) such that y is a child of x.

- $\texttt{Person} \sqcap \neg\texttt{Female}$

  The set of individuals that are persons but not female.

- $\exists\texttt{hasChild}.\texttt{Male} \sqcap \forall\texttt{hasChild}.\texttt{Male}$

  The set of individuals that have at least one child who is male, and additionally, all of their children are male.

Examples - Informal Semantics (cont'd)

- $(\geqslant 3\ \texttt{hasChild}) \sqcap \texttt{Male}$

  The set of individuals that have at least 3 children, and additionally, they are male.

- $(\geqslant 3\ \texttt{hasChild.Male})$

  The set of individuals that have at least 3 children that are male.

- $\exists \texttt{hasChild}.(\exists \texttt{hasChild.Person})$

  The set of individuals that have at least one child who has at least one child that is a person (i.e., the set of individuals that are grandparents).

Examples - Informal Semantics (cont'd)

- $\forall$`hasChild`.($\exists$`isAlumniOf`.`GreekUniversity`)

  The set of individuals such that all their children have graduated from at least one Greek University.

- $\forall$`hasChild`.(`Doctor` $\sqcap$ $\exists$`isAlumniOf`.`GreekUniversity`)

  The set of individuals such that all their children are doctors that have graduated from at least one Greek University.

- ($\geqslant$3 `hasChild`.($\forall$`isAlumniOf`.`GreekUniversity`))

  The set of individuals that have at least three children such that all their degrees are from Greek Universities.

Examples - Informal Semantics (cont'd)

- $(\geqslant 3\ \texttt{hasChild}.(\texttt{Doctor} \sqcap \forall \texttt{isAlumniOf}.\texttt{GreekUniversity}))$

  The set of individuals that have at least three children that are doctors and have degrees only from Greek Universities.

- $\texttt{Female} \sqcap (\geqslant 3\ \texttt{hasChild}.(\texttt{Doctor} \sqcap \forall \texttt{isAlumniOf}.\texttt{GreekUniversity}))$

  The set of female individuals that have at least three children that are doctors and all their degrees are from Greek Universities.

Examples - Informal Semantics (cont'd)

- $\texttt{Male} \sqsubseteq \neg\texttt{Female}$

  The set of male individuals and the set of female individuals are disjoint.

- $\texttt{Woman} \equiv \texttt{Person} \sqcap \texttt{Female}$

  An individual is a woman if and only if she is a female person.

- $\texttt{Female}(\texttt{ANNA})$

  The individual denoted by `ANNA` is female.

- $\texttt{hasChild}(\texttt{ANNA}, \texttt{JOHN})$

  The individual denoted by `JOHN` is a child of the individual denoted by `ANNA`.

Examples - Informal Semantics (cont'd)

- $((\geqslant 3\ \texttt{hasChild}) \sqcap \texttt{Male})(\texttt{JOHN})$

  The individual denoted by JOHN is male and has a least 3 children.

- $\texttt{Female} \sqcap (\geqslant 3\ \texttt{hasChild}.(\texttt{Doctor} \sqcap \forall \texttt{isAlumniOf}.\texttt{GreekUniversity}))(\texttt{ANNA})$

  The individual denoted by ANNA is female and has at least three children that are doctors and all their degrees are from Greek Universities.

Outline

- History of DLs
- Informal Introduction to DLs
- **A Simple DL:** $\mathcal{ALC}$
- Semantics of $\mathcal{ALC}$
- Reasoning problems in $\mathcal{ALC}$
- The DLs $\mathcal{ALCN}$, $\mathcal{ALCQ}$ and $\mathcal{ALCQO}$
- Translating DLs to FOL

$\mathcal{ALC}$: The Smallest Propositionally Closed DL

Syntax	Semantics	Terminology
A	$A^{\mathcal{I}} \subseteq \Delta$	atomic concept
R	$R^{\mathcal{I}} \subseteq \Delta \times \Delta$	atomic role
$\top$	Δ	top (universal) concept
$\bot$	$\emptyset$	bottom concept
$\neg C$	$\Delta \setminus C^{\mathcal{I}}$	concept complement
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	concept conjunction
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	concept disjunction
$\forall R.C$	$\{x \mid (\forall y)((x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}})\}$	universal restriction
$\exists R.C$	$\{x \mid (\exists y)((x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$	existential restriction

$\mathcal{ALC}$ Syntax

To define the syntax of $\mathcal{ALC}$, 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**.

$\mathcal{ALC}$ Syntax: Concepts

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

1. $\top$ (**top concept**) and $\bot$ (**bottom concept**) are concepts.
2. Every **concept name** is a concept.
3. 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**)
4. Nothing else is a concept.

$\mathcal{ALC}$ Syntax: Terminological Axioms

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".

Note: In the literature, concept definitions are frequently written as $C \doteq D$.

Examples

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

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

$$\texttt{Student} \sqsubseteq \texttt{Person}$$

$$\texttt{Student} \sqsubseteq \exists \texttt{enrolled.Course}$$

Intuitive Meaning of Concept Definitions

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

In a set of concept definitions, we 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.

Name symbols appearing in concept definitions are usually called **defined concepts** and base symbols **primitive concepts**.

Primitive vs. Defined Concepts

An important feature of DLs is their ability to **distinguish** primitive from defined concepts:

- **Defined concepts** have necessary and sufficient conditions for concept membership.

  Examples: woman, mother, driver, white wine etc.

- **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).

Example: $\texttt{Student} \sqsubseteq \exists\texttt{enrolled}.\texttt{Course}$

Concept inclusions express "if" statements.

Necessary and Sufficient Conditions

A **concept equivalence (definition)** of the form $C \equiv D$ states a **necessary and sufficient 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). If an individual is in D, this is a sufficient condition for concluding that it is also in C.

Example:

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

Concept equivalences express "if and only if" statements.

Example: Family Relationships

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

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

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

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

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

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

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

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

Examples of Concept Inclusions

Concept inclusions are useful for expressing properties of concepts and roles. For example:

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

$\mathcal{ALC}$ Syntax: Assertions about Individuals

In $\mathcal{ALC}$, 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.

Examples of Assertions

$$\texttt{Student}(\texttt{JOHN})$$

$$\texttt{enrolled}(\texttt{JOHN}, \texttt{CS415})$$

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

$$(\texttt{Female} \sqcap (\geqslant 3\ \texttt{hasChild}.(\texttt{Doctor} \sqcap \forall \texttt{isAlumniOf}.\texttt{GreekUniversity})))(\texttt{ANNA})$$

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** $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{A})$ where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an Abox.

Outline

- History of DLs
- Informal Introduction to DLs
- A Simple DL: $\mathcal{ALC}$
- **Semantics of $\mathcal{ALC}$**
- Reasoning problems in $\mathcal{ALC}$
- The DLs $\mathcal{ALCN}$, $\mathcal{ALCQ}$ and $\mathcal{ALCQO}$
- Translating DLs to FOL

$\mathcal{ALC}$ Semantics

Definition. An **interpretation** $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ which consists of:

- a nonempty set $\Delta^{\mathcal{I}}$ (the **domain** of the interpretation)
- a function $\cdot^{\mathcal{I}}$ (the **interpretation function**) which maps
  - every individual name a to an element $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$
  - every concept name C to a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$
  - every role name R to a subset $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

Unique Names Assumption (UNA): We will assume that if a and b are distinct individuals then $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

Note: The UNA might not be assumed in related ontology languages e.g., OWL.

$\mathcal{ALC}$ Semantics (cont'd)

Then $\mathcal{I}$ is extended to arbitrary concepts as follows:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$$

$$\bot^{\mathcal{I}} = \emptyset$$

$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$

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

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

$\mathcal{ALC}$ Semantics (cont'd)

Notice that $\mathcal{ALC}$ is a **propositionally closed** language:

- $\neg\top \equiv \bot$
- $\neg\bot \equiv \top$
- $\neg(C \sqcap D) \equiv \neg C \sqcup \neg D$
- $\neg(C \sqcup D) \equiv \neg C \sqcap \neg D$
- $\neg(\forall R.C) \equiv \exists R.\neg C$
- $\neg(\exists R.C) \equiv \forall R.\neg C$

TBox: Semantics

Satisfaction. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation.

- $\mathcal{I}$ **satisfies** the statement $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- $\mathcal{I}$ **satisfies** the statement $C \equiv D$ if $C^{\mathcal{I}} = D^{\mathcal{I}}$.

Model. An interpretation $\mathcal{I}$ is a **model** for a TBox $\mathcal{T}$ if $\mathcal{I}$ satisfies all the statements in $\mathcal{T}$.

Satisfiability. A TBox $\mathcal{T}$ is **satisfiable** if it has a model.

ABox: Semantics

Satisfaction. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation.

- $\mathcal{I}$ **satisfies** $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$.
- $\mathcal{I}$ **satisfies** $R(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

Model. An interpretation $\mathcal{I}$ is a **model** of an ABox $\mathcal{A}$ if it satisfies every assertion of $\mathcal{A}$.

Satisfiability. An ABox $\mathcal{A}$ is **satisfiable** if it has a model.

Knowledge Bases - Semantics

Satisfaction. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ **satisfies** a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ satisfies both $\mathcal{T}$ and $\mathcal{A}$.

Model. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a **model** of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ is a model of $\mathcal{T}$ and $\mathcal{A}$.

Satisfiability. A knowledge base $\mathcal{K}$ is **satisfiable** if it has a model.

Entailment (Logical Implication)

Definition. Let $\mathcal{K}$ be a knowledge base and φ a terminological axiom or an assertion. We will say that $\mathcal{K}$ **entails** φ (denoted by $\mathcal{K} \models \varphi$) if every model of $\mathcal{K}$ is a model of φ.

Example: Let the TBox $\mathcal{T}$ be

$$\texttt{Female} \sqsubseteq \texttt{Person}$$

and the ABox $\mathcal{A}$ be

$$\texttt{Female}(\texttt{ANNA}).$$

If $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ then $\mathcal{K} \models \texttt{Person}(\texttt{ANNA})$.

Example

Let the TBox $\mathcal{T}$ be

$$\exists \texttt{teaches}.\texttt{Course} \sqsubseteq \neg\texttt{Undergrad} \sqcup \texttt{Professor}$$

and the ABox $\mathcal{A}$ be

$$\texttt{teaches}(\texttt{JOHN}, \texttt{CS415}),\ \texttt{Course}(\texttt{CS415}),\ \texttt{Undergrad}(\texttt{JOHN}).$$

If $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ then $\mathcal{K} \models \texttt{Professor}(\texttt{JOHN})$.

Example (cont'd))

There is nothing wrong with the entailment

$$\mathcal{K} \models \texttt{Professor}(\texttt{JOHN})$$

since the TBox has no axiom that precludes somebody from being and undergrad and also a professor.

Example (Revisited)

Let the TBox $\mathcal{T}$ be

$$\exists \texttt{teaches}.\texttt{Course} \sqsubseteq \texttt{Undergrad} \sqcup \texttt{Professor}$$

and the ABox $\mathcal{A}$ be

$$\texttt{teaches}(\texttt{JOHN}, \texttt{CS415}),\ \texttt{Course}(\texttt{CS415}),\ \texttt{Undergrad}(\texttt{JOHN}).$$

If $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ which one of the following holds?

$$\mathcal{K} \models \texttt{Professor}(\texttt{JOHN}), \quad \mathcal{K} \models \neg\texttt{Professor}(\texttt{JOHN})$$

Example

Let the TBox $\mathcal{T}$ be

$$\exists\texttt{hasChild}.\top \sqsubseteq \texttt{Parent}, \quad \top \sqsubseteq \forall\texttt{hasChild}.\texttt{Person}$$

and the ABox $\mathcal{A}$ be

$$\texttt{hasChild}(\texttt{ANNA}, \texttt{JOHN}).$$

If $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ then

$$\mathcal{K} \models \texttt{Parent}(\texttt{ANNA}) \text{ and } \mathcal{K} \models \texttt{Person}(\texttt{JOHN}).$$

Validity

Definition. Let φ be a terminological axiom or assertion. We will say that φ is **valid** if every interpretation is a model of φ.

Examples:

$$\mathtt{A} \sqcap \mathtt{B} \sqsubseteq \mathtt{A}, \quad \mathtt{A} \sqcap \mathtt{B} \sqcap \mathtt{C} \sqsubseteq \mathtt{A} \sqcap \mathtt{B}, \quad \forall \mathtt{R}.(\mathtt{A} \sqcap \mathtt{B}) \sqsubseteq \forall \mathtt{R}.\mathtt{A}$$

$$\top(\mathtt{ANNA}), \quad \neg\bot(\mathtt{ANNA})$$

Outline

- History of DLs
- Informal Introduction to DLs
- A Simple DL: $\mathcal{ALC}$
- Semantics of $\mathcal{ALC}$
- **Reasoning problems in** $\mathcal{ALC}$
- The DLs $\mathcal{ALCN}$, $\mathcal{ALCQ}$ and $\mathcal{ALCQO}$
- Translating DLs to FOL

Typical Reasoning Problems

- Concept satisfiability
- Subsumption
- Classification
- Knowledge base satisfiability
- Instance checking
- Answering queries
- Realization

Concept satisfiability

This is the problem of checking whether a concept C is **satisfiable with respect to a knowledge base** $\mathcal{K}$, i.e., whether there exists a model $\mathcal{I}$ of $\mathcal{K}$ such that $C^{\mathcal{I}} \neq \emptyset$.

Formally: $\mathcal{K} \not\models C \equiv \bot$

Example

Let us consider the knowledge base $\mathcal{K}$ containing exactly the following sentences:

$$\texttt{ParentOfOnlyMaleChildren} \equiv \texttt{Person} \sqcap \exists \texttt{hasChild.Male} \sqcap \forall \texttt{hasChild.Male}$$

$$\texttt{Male} \sqsubseteq \neg \texttt{Female}$$

Questions:

- Is the concept

$$\texttt{ParentOfOnlyMaleChildren} \sqcap \exists \texttt{hasChild.Male}$$

  satisfiable with respect to $\mathcal{K}$?

- Is the concept

$$\texttt{ParentOfOnlyMaleChildren} \sqcap \exists \texttt{hasChild.Female}$$

  satisfiable with respect to $\mathcal{K}$?

Subsumption

This is the problem of checking whether C **is subsumed by** D **with respect to a knowledge base** $\mathcal{K}$, i.e., whether $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in every model $\mathcal{I}$ of $\mathcal{K}$.

Formally: $\mathcal{K} \models C \sqsubseteq D$

Example

Let us consider the knowledge base $\mathcal{K}$ containing exactly the following sentences:

$$\texttt{Parent} \equiv \texttt{Person} \sqcap \exists \texttt{hasChild.Person} \sqcap \forall \texttt{hasChild.Person}$$

$$\texttt{ParentOfOnlyMaleChildren} \equiv \texttt{Person} \sqcap \exists \texttt{hasChild.Male} \sqcap \forall \texttt{hasChild.Male}$$

$$\texttt{Male} \sqsubseteq \texttt{Person}$$

Example (cont'd)

Questions:

- Does the subsumption relationship

$$\texttt{ParentOfOnlyMaleChildren} \sqsubseteq \texttt{Parent}$$

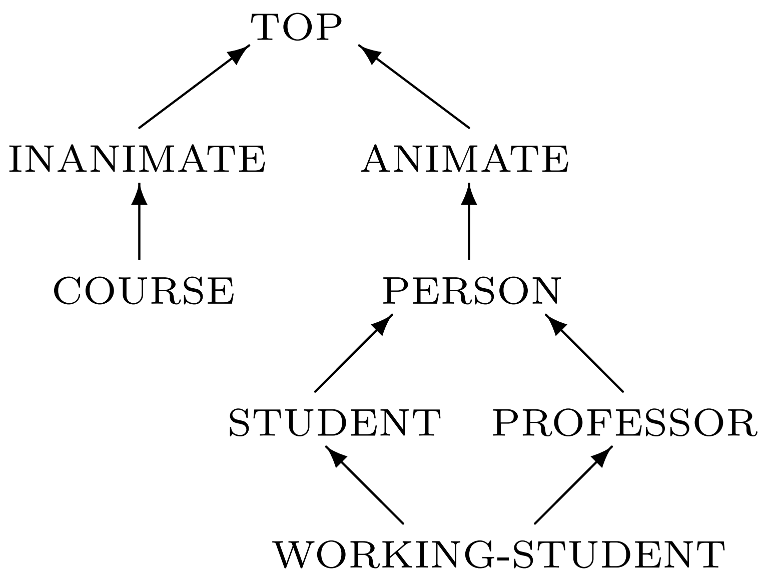
  hold with respect to $\mathcal{K}$?

- Does the subsumption relationship

$$\texttt{ParentOfOnlyMaleChildren} \sqsubseteq \texttt{Male}$$

  hold with respect to $\mathcal{K}$?

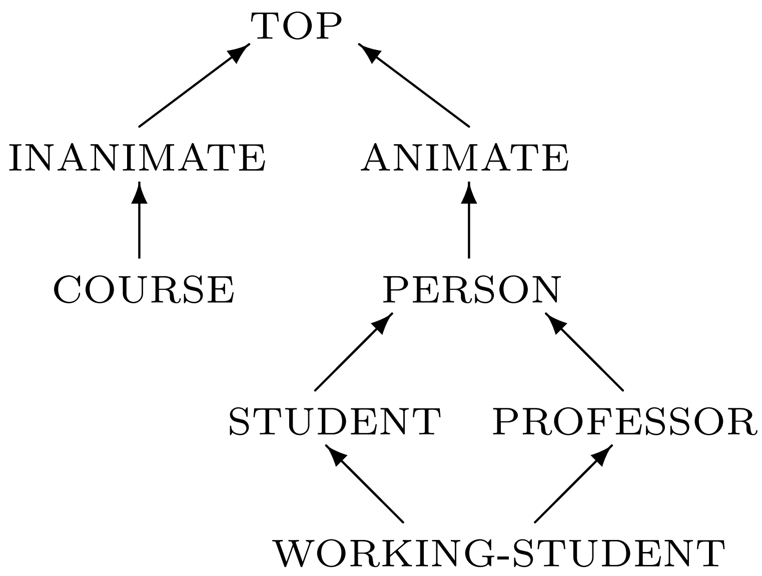
Taxonomies


TOP
INANIMATE
ANIMATE
COURSE
PERSON
STUDENT
PROFESSOR
WORKING-STUDENT

Taxonomies (cont'd)

- The subsumption relationship between concepts defined by $\sqsubseteq$ is a **partial order** (i.e., it is reflexive, antisymmetric and transitive).
- Subsumption induces a **taxonomy** such as the one on the previous slide where only direct subsumptions have been explicitly drawn.

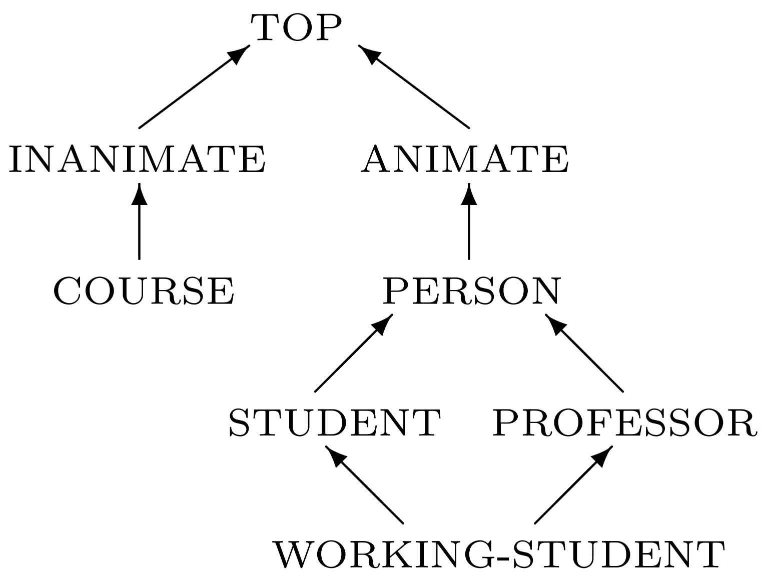
Taxonomies (cont'd)



Question: What is the place of the following concept in the above taxonomy?

$$\mathtt{N} \equiv \mathtt{ANIMATE} \sqcap (\mathtt{STUDENT} \sqcup \mathtt{PROFESSOR})$$

Taxonomies (cont'd)



Answer:

$$\texttt{STUDENT} \sqsubseteq \texttt{N}, \quad \texttt{PROFESSOR} \sqsubseteq \texttt{N}, \quad \texttt{N} \sqsubseteq \texttt{Person}$$

Classification

- The problem of **classification:** Given a concept C and a TBox $\mathcal{T}$, for all concepts D of $\mathcal{T}$ determine whether D subsumes C, or D is subsumed by C.
- Intuitively, this amounts to finding the "right place" for C in the taxonomy implicitly present in $\mathcal{T}$.
- **Classification** is the task of inserting new concepts in a taxonomy. It is **sorting** in partial orders.
- What is the solution to the classification problem posed in the previous slide?

Knowledge base satisfiability

This is the problem of checking whether $\mathcal{K}$ is **satisfiable**, i.e., whether it **has a model**.

Example

Is the knowledge base containing exactly the following sentences satisfiable?

$$\mathtt{ParentOfOnlyMaleChildren} \equiv \mathtt{Person} \sqcap \exists \mathtt{hasChild.Male} \sqcap \forall \mathtt{hasChild.Male}$$

$$\mathtt{Male} \sqsubseteq \mathtt{Person}$$

$$\mathtt{Male} \sqsubseteq \neg \mathtt{Female}$$

$$\mathtt{Male(JOHN)},\ \mathtt{Male(NICK)},\ \mathtt{Female(ANNA)},$$

$$\mathtt{hasChild(JOHN, NICK)},\ \mathtt{hasChild(JOHN, ANNA)},$$

$$\mathtt{ParentOfOnlyMaleChildren(JOHN)}$$

Example

Is the knowledge base containing exactly the following sentences satisfiable?

$$\texttt{ParentOfOnlyMaleChildren} \equiv \texttt{Person} \sqcap \exists \texttt{hasChild.Male} \sqcap \forall \texttt{hasChild.Male}$$

$$\texttt{Male} \sqsubseteq \neg \texttt{Female}$$

$$\texttt{ParentOfOnlyMaleChildren} \sqsubseteq \exists \texttt{hasChild.Female}$$

Instance checking

This is the problem of checking whether the assertion $C(a)$ is satisfied in every model of $\mathcal{K}$.

Formally: $\mathcal{K} \models C(a)$

Answering concept queries

Find all a such that $\{a \mid \mathcal{K} \models C(a)\}$.

Example

Let us consider the knowledge base $\mathcal{K}$ containing exactly the following sentences:

$$\texttt{Parent} \equiv \texttt{Person} \sqcap \exists \texttt{hasChild.Person} \sqcap \forall \texttt{hasChild.Person}$$

$$\texttt{ParentOfOnlyMaleChildren} \equiv \texttt{Person} \sqcap \exists \texttt{hasChild.Male} \sqcap \forall \texttt{hasChild.Male}$$

$$\texttt{Male} \sqsubseteq \texttt{Person}$$

$$\texttt{Female} \sqsubseteq \texttt{Person}$$

$$\texttt{Male}(\texttt{JOHN}),\ \texttt{Male}(\texttt{NICK}),\ \texttt{Female}(\texttt{ANNA}),$$

$$\texttt{hasChild}(\texttt{JOHN}, \texttt{NICK}),\ \texttt{hasChild}(\texttt{JOHN}, \texttt{ANNA})$$

Example (cont'd)

Questions:

- Which ones of the following assertions

  `Person`(`NICK`), `Person`(`ANNA`), `Parent`(`JOHN`), `Parent`(`NICK`),

  `Parent`(`ANNA`), `ParentOfMaleChildren`(`JOHN`),

  are entailed by $\mathcal{K}$?
- Find all individuals that are parents.
- Find all individuals that are parents of only male children.

Realization

Given an individual a, find the **most specific concepts** C such that $\mathcal{K} \models C(a)$. These concepts are the **lowest ones in the taxonomy** induced by the subsumption relationship.

Example

Let us consider the knowledge base $\mathcal{K}$ containing exactly the following sentences:

$$\texttt{Parent} \equiv \texttt{Person} \sqcap \exists \texttt{hasChild.Person} \sqcap \forall \texttt{hasChild.Person}$$

$$\texttt{ParentOfOnlyMaleChildren} \equiv \texttt{Person} \sqcap \exists \texttt{hasChild.Male} \sqcap \forall \texttt{hasChild.Male}$$

$$\texttt{Male} \sqsubseteq \texttt{Person}$$

$$\texttt{Female} \sqsubseteq \texttt{Person}$$

$$\texttt{Male}(\texttt{JOHN}),\ \texttt{Male}(\texttt{NICK}),\ \texttt{Female}(\texttt{ANNA}),$$

$$\texttt{hasChild}(\texttt{JOHN}, \texttt{NICK}),\ \texttt{hasChild}(\texttt{JOHN}, \texttt{ANNA})$$

Example (cont'd)

Questions:

- Draw the concept taxonomy corresponding to the above knowledge base.
- Realize the following individuals:

`NICK, ANNA, JOHN`

Reduction to Satisfiability

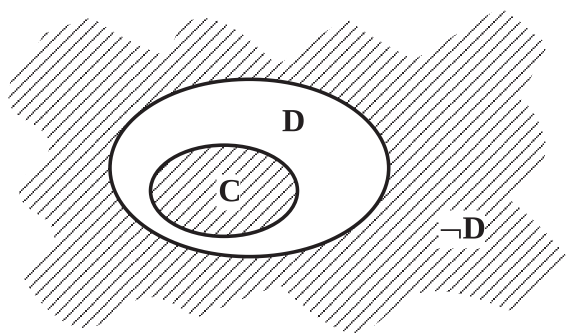
Some of the previous reasoning problems can be solved by reducing them to the problem of knowledge base satisfiability:

- Concept Satisfiability

  $\mathcal{K} \not\models C \equiv \bot$ iff there exists an x such that $\mathcal{K} \cup \{C(x)\}$ is satisfiable

- Subsumption

  $\mathcal{K} \models C \sqsubseteq D$ iff there exists an x such that $\mathcal{K} \cup \{(C \sqcap \neg D)(x)\}$ is not satisfiable



Reduction to Satisfiability (cont'd)

- Instance Checking

$$\mathcal{K} \models C(a) \text{ iff } \mathcal{K} \cup \{\neg C(a)\} \text{ is not satisfiable}$$

Reasoning Algorithms

- Terminating, complete and efficient algorithms for deciding **satisfiability**, and all the other reasoning problems mentioned earlier, are available for $\mathcal{ALC}$.
- These algorithms are based on **tableaux-calculi** techniques.
- Completeness is important for the usability of description logics in real applications.
- Such algorithms have been shown to be **efficient** for real knowledge bases, even if the problem in the corresponding logic is in PSPACE or EXPTIME.
- We will talk about tableaux-calculi for DLs in the next lecture.

Outline

- History of DLs
- Informal introduction to DLs
- A Simple DL: $\mathcal{ALC}$
- Semantics of $\mathcal{ALC}$
- Reasoning problems in $\mathcal{ALC}$
- **The DLs $\mathcal{ALCN}$, $\mathcal{ALCQ}$ and $\mathcal{ALCQO}$**
- Translating DLs to FOL

Number Restrictions

Role quantification **cannot** express that a teacher teaches **at least 3** (or **at most 5**) courses.

Number restrictions can express **arithmetic constraints on the number of fillers of a role**.

Examples:

- $\texttt{BusyTeacher} \equiv \texttt{Teacher} \sqcap (\geqslant 3\ \texttt{teaches})$
- $\texttt{ConsciousTeacher} \equiv \texttt{Teacher} \sqcap (\leqslant 5\ \texttt{teaches})$

Number Restrictions (cont'd)

Semantics:

$$(\geqslant n\ R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid card(\{y \mid (x,y) \in R^{\mathcal{I}}\}) \geq n\}$$

$$(\leqslant n\ R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid card(\{y \mid (x,y) \in R^{\mathcal{I}}\}) \leq n\}$$

Observation: $(\geqslant 1\ R) \equiv \exists R.\top$

The above number restrictions are called **unqualified.**

Notation: The description logic which extends $\mathcal{ALC}$ with number restrictions is denoted by $\mathcal{ALCN}$.

Qualified Number Restrictions

Qualified number restrictions **constrain the number but also the type of fillers** (they give the concepts to which fillers should belong).

Examples:

- $\texttt{BusyTeacher} \equiv \texttt{Teacher} \sqcap (\geqslant 3\ \texttt{teaches.Course})$
- $\texttt{ConsciousTeacher} \equiv \texttt{Teacher} \sqcap (\leqslant 5\ \texttt{teaches.Course})$

Semantics:

$$(\geqslant n\ R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid card(\{y \mid (x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}) \geq n\}$$

$$(\leqslant n\ R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid card(\{y \mid (x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}) \leq n\}$$

Notation: The description logic which extends $\mathcal{ALC}$ with qualified number restrictions is denoted by $\mathcal{ALCQ}$.

Enumerations - Nominals

Sometimes it is useful to define a concept that contains **exactly the individuals** $\mathtt{I}_1, \ldots, \mathtt{I}_\mathtt{m}$. This concept is written as $\{\mathtt{I}_1, \ldots, \mathtt{I}_\mathtt{m}\}$.

Examples:

$$\mathtt{Weekday} \equiv \{\mathtt{MON}, \mathtt{TUE}, \mathtt{WED}, \mathtt{THU}, \mathtt{FRI}, \mathtt{SAT}, \mathtt{SUN}\}$$

$$\mathtt{Citizen} \equiv \mathtt{Person} \sqcap \exists \mathtt{hasCountry}.\mathtt{Country}$$

$$\mathtt{Greek} \equiv \mathtt{Citizen} \sqcap \exists \mathtt{hasCountry}.\{\mathtt{GREECE}\}$$

Nominals

A **nominal** is a concept that contains **exactly one** individual.

If we have the ability to define nominals, then using $\sqcup$, we can define concepts containing more than one individual.

Example: $\texttt{Weekend} \equiv \{\texttt{SAT}, \texttt{SUN}\}$

If we have nominals and we **do not want to make the UNA**, then we can explicity state whether two individuals are the same or different.

Examples:

$$\{\texttt{CRETA}\} \equiv \{\texttt{CRETE}\}, \quad \{\texttt{CRETE}\} \sqcap \{\texttt{CYPRUS}\} \sqsubseteq \bot$$

Notation: If we add nominals to DL $\mathcal{ALCQ}$, we get the DL $\mathcal{ALCQO}$.

A Naming Scheme for DLs

Historically, in the family of languages we presented, the first language was $\mathcal{AL}$ (**attributive concept description language**). Extensions of $\mathcal{AL}$ have been studied and have been identified by strings of the form:

$$\mathcal{AL}[\mathcal{C}][\mathcal{N}][\mathcal{Q}]\cdots$$

The name $\mathcal{ALC}$ originally comes from "**attributive concept description language with complement**".

Because combinations of constructs can **simulate** others there can be different names for languages that are essentially the same, i.e., have the same expressive power.

Example: $\mathcal{ALCQ}$ has same expressivity as $\mathcal{ALCNQ}$. Why?

Example

Let a knowledge base $\mathcal{K}$ be:

$$\texttt{BusyTeacher} \equiv \texttt{Teacher} \sqcap (\geqslant 3\ \texttt{teaches})$$

$$\texttt{ConsciousTeacher} \equiv \texttt{Teacher} \sqcap (\leqslant 5\ \texttt{teaches})$$

$$\texttt{Teacher}(\texttt{MARY}),$$

$$\texttt{teaches}(\texttt{MARY}, \texttt{AI}),\ \texttt{teaches}(\texttt{MARY}, \texttt{KR}),\ \texttt{teaches}(\texttt{MARY}, \texttt{DB})$$

Questions:

- $\mathcal{K} \models \texttt{BusyTeacher}(\texttt{MARY})$?
- $\mathcal{K} \models \texttt{ConsciousTeacher}(\texttt{MARY})$?

Example

Let $\mathcal{K}$ be the following knowledge base:

$$\texttt{Family}(\texttt{F})$$

$$\texttt{Father}(\texttt{F}, \texttt{JOHN}),\ \texttt{Mother}(\texttt{F}, \texttt{SUE})$$

$$\texttt{Son}(\texttt{F}, \texttt{PAUL}),\ \texttt{Son}(\texttt{F}, \texttt{GEORGE}),\ \texttt{Son}(\texttt{F}, \texttt{ALEX})$$

Question: How many children does family `F` have?

OWA vs. CWA

Contrary to databases, DLs make the Open World Assumption. Absence of information is **not** interpreted as presence of negative information but simply as **lack of knowledge.**

Thus in the previous example:

- $\mathcal{K} \models (\geqslant 3\ \texttt{Son})(\texttt{F})$ **Yes**
- $\mathcal{K} \models (\leqslant 1\ \texttt{Son})(\texttt{F})$ **No**
- $\mathcal{K} \models (\geqslant 5\ \texttt{Son})(\texttt{F})$ **Unknown**

Outline

- History of DLs
- Informal introduction to DLs
- A Simple DL: $\mathcal{ALC}$
- Semantics of $\mathcal{ALC}$
- Reasoning problems in $\mathcal{ALC}$
- The DLs $\mathcal{ALCN}$, $\mathcal{ALCQ}$ and $\mathcal{ALCQO}$
- **Translating DLs to FOL**

Translating DLs to FOL

We will now give a translation of $\mathcal{ALCQO}$ statements to FOL statements. This shows that $\mathcal{ALCQO}$ (and DLs in general) are subsets of FOL and provides us with an alternative semantics for DLs.

We will give a function π that translates any axiom of an $\mathcal{ALCQO}$ knowledge base into a FOL statement (with equality and unary predicates $\top$ and $\bot$ with obvious semantics).

The resulting FOL statement will contain a unary predicate for each concept name and a binary predicate for each role name in the $\mathcal{ALCQO}$ axiom.

Translating Concept Inclusions

Let A be a concept name, C, D be arbitrary concept expressions and R a role name. Then:

$$\pi(C \sqsubseteq D) = (\forall x)(\pi_x(C) \Rightarrow \pi_x(D))$$

$$\pi_x(A) = A(x)$$

$$\pi_x(\neg C) = \neg \pi_x(C)$$

$$\pi_x(C \sqcap D) = \pi_x(C) \wedge \pi_x(D)$$

$$\pi_x(C \sqcup D) = \pi_x(C) \vee \pi_x(D)$$

$$\pi_x(\forall R.C) = (\forall x_1)(R(x, x_1) \Rightarrow \pi_{x_1}(C))$$

$$\pi_x(\exists R.C) = (\exists x_1)(R(x, x_1) \wedge \pi_{x_1}(C))$$

Translating Concept Inclusions (cont'd)

$$\pi_x((\geqslant n \;\; R.\ C)) = (\exists x_1) \cdots (\exists x_n)(\bigwedge_{i \neq j} (x_i \neq x_j) \wedge \bigwedge_{i} (R(x, x_i) \wedge \pi_{x_i}(C)))$$

$$\pi_x((\leqslant n \;\; R.\ C)) = \neg(\exists x_1) \cdots (\exists x_{n+1})(\bigwedge_{i \neq j} (x_i \neq x_j) \wedge \bigwedge_{i} (R(x, x_i) \wedge \pi_{x_i}(C)))$$

$$\pi_x(\{a\}) = (x = a)$$

Notes on the Translation

In the previous translation, π_x, π_{x_1} etc. are auxiliary functions where x, x_1 etc. are FOL variables. The variables introduced in the right hand sides of the above translations must be **new** variables that have not been used before in the translation.

Note: We do not give a translation for **concept definition** since they can be rewritten using concept inclusion and conjunction.

Translating ABox Assertions

Let C be concept name, R a role name and a, b are individual names. Then:

$$\pi(C(a)) = C(a)$$

$$\pi(R(a, b)) = R(a, b)$$

Note: We do not give a translation for the case when C is an arbitrary concept expression. Each such concept assertion $C(a)$ can be written into two axioms $D \equiv C,\ D(a)$ where D is a new concept name.

Example

The FOL expression for concept inclusion

$$\texttt{Male} \sqsubseteq \neg\texttt{Female}$$

is

$$(\forall \texttt{x})(\texttt{Male}(\texttt{x}) \Rightarrow \neg\texttt{Female}(\texttt{x})).$$

The FOL expression for concept inclusion

$$\forall\texttt{hasChild}.(\exists\texttt{isAlumniOf}.\texttt{GreekUniversity}) \sqsubseteq \texttt{ProudGreekFather}$$

is

$$(\forall \texttt{x})((\forall \texttt{y})(\texttt{hasChild}(\texttt{x},\texttt{y}) \Rightarrow (\exists \texttt{z})(\texttt{isAlumniOf}(\texttt{y},\texttt{z}) \land \texttt{GreekUniversity}(\texttt{z})))$$

$$\Rightarrow \texttt{ProudGreekFather}(\texttt{x})).$$

Implemented DL Systems

- The beginning of it all: KL-ONE (1977)
- KRYPTON (1983), NIKL (1983), KANDOR (1984), PENNI, KL-TWO (1985)
- Second generation DL systems: LOOM (1987), CLASSIC (1989)
- BACK (1990), FLEX (1995), KRIS (1991), CRACK (1995)
- Optimization techniques take charge: FaCT (1997), DLP (1998), RACER (1999)
- DL reasoners for the ontologies and Semantic Web era: FaCT++, RacerPro, KAON2, Pellet, HermiT

See `http://www.cs.man.ac.uk/~sattler/reasoners.html` for pointers to Web pages of DL reasoners.

Applications of DLs

- Conceptual Modelling
- Data Integration
- Configuration
- Software Engineering
- Medical Informatics
- Bioinformatics
- Natural Language Processing
- Knowledge Representation and Reasoning in the Semantic Web (remaining of this course!)

Readings

- Franz Baader, Ian Horrocks, and Ulrike Sattler. Description Logics. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, Handbook of Knowledge Representation. Elsevier, 2007.

  Available from `http://www.comlab.ox.ac.uk/people/ian.horrocks/Publications/complete.html#2007`

  This is a recent comprehensive survey of the area of DLs.

- Franz Baader. Description Logics. In "Reasoning Web: Semantic Technologies for Information Systems". 5th International Summer School 2009, volume 5689 of Lecture Notes in Computer Science, pages 1-39. Springer-Verlag, 2009.

  Available from `http://lat.inf.tu-dresden.de/research/papers.html`.

  Another excellent recent survey of the area of DLs.

Readings (cont'd)

- R. J. Brachman and D. L. McGuinness and P. Patel-Schneider and L. A. Resnick and A. Borgida. Living with CLASSIC: When and how to use a KL-ONE-like language. In *Principles of Semantic Networks*. John Sowa (editor), Morgan Kaufmann, 1991, pages 401-456.

  Available from:

  `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.31.9028`

  This paper contains a lot of nice examples so it is great for explaining where to use description logics. The syntax used is that of the DL-based language CLASSIC, but this should not be a problem in appreciating the examples and discussion in the paper.

Readings (cont'd)

- Chapter 1 (An Introduction to DLs) and Chapter 2 (Reasoning in DLs) and Chapter 10 (Conceptual Modelling with DLs) of the DL Handbook:

  F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P. F. Patel-Schneider (editors). The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2002.

  Available from:

  `http://www.inf.unibz.it/~franconi/dl/course/dlhb/home.html`

  Chapters 1 and 2 are good introductions to DLs.

  Chapter 10 is useful for ontology development using DLs.

Acknowledgements

These slides have been prepared by modifying slides by Enrico Franconi, University of Bolzano-Bozen, Italy.

See `http://www.inf.unibz.it/~franconi/dl/course/` for Enrico's course on DLs.

Some other courses on DLs are listed on `http://dl.kr.org/courses.html`.