Semantic Web

2. Description logics (1/2)

Gerd Gröner, Matthias Thimm

{groener,thimm}@uni-koblenz.de

Institute for Web Science and Technologies (WeST)
University of Koblenz-Landau

July 8, 2013

Recap

- What is the Semantic Web?
- The Semantic Web Architecture
  - Unicode, URIs, XML
  - RDF, RDF-S, SPARQL
  - OWL, RIF
  - Unifying Logic, Proof, Trust, Crypto
  - User interfaces and applications
- Linked Open Data

Recap

- What is the Semantic Web?
- The Semantic Web Architecture
  - Unicode, URIs, XML
  - RDF, RDF-S, SPARQL
  - OWL, RIF
  - Unifying Logic, Proof, Trust, Crypto
  - User interfaces and applications
- Linked Open Data

Home assignment:
Go to the DBPedia entry of "Paul McCartney" (`http://dbpedia.org/page/Paul_McCartney`) and try out navigating through the relationships. Can you find the semantic relationship between "Paul McCartney" and the band "Spooky Tooth" (`http://dbpedia.org/page/Spooky_Tooth`)?

Recap

- What is the Semantic Web?
- The Semantic Web Architecture
  - Unicode, URIs, XML
  - RDF, RDF-S, SPARQL
  - OWL, RIF
  - Unifying Logic, Proof, Trust, Crypto
  - User interfaces and applications
- Linked Open Data

Home assignment:
Go to the DBPedia entry of "Paul McCartney" (`http://dbpedia.org/page/Paul_McCartney`) and try out navigating through the relationships. Can you find the semantic relationship between "Paul McCartney" and the band "Spooky Tooth" (`http://dbpedia.org/page/Spooky_Tooth`)?

"Paul McCartney" - "Wings" - "Henry McCullough" - "Spooky Tooth"

Overview

- Description logics are the formal basis for most knowledge representation formalisms on the semantic web

Overview

- Description logics are the formal basis for most knowledge representation formalisms on the semantic web
- Description logic knowledge bases are a way to formalize *ontologies*

Overview

- Description logics are the formal basis for most knowledge representation formalisms on the semantic web
- Description logic knowledge bases are a way to formalize *ontologies*
- We will discuss general concepts of ontology formalization and the description logic $\mathcal{ALC}$
  - Syntax
  - Semantics
  - Inference

WeST People and Knowledge Networks

Outline

1. Knowledge and Ontologies
2. The Description Logic $\mathcal{ALC}$
   - Syntax
   - Semantics
   - $\mathcal{ALC}$ and first-order logic
3. Summary and Exercises

Outline

WeST People and Knowledge Networks

Ontologies 1/3

Ontology is the philosophical study of the nature of being, becoming, existence, or reality, as well as the basic categories of being and their relations.

In computer science and information science, an ontology formally represents knowledge as a set of concepts within a domain, and the relationships between pairs of concepts. It can be used to model a domain and support reasoning about concepts.

[Wikipedia]

- In computer science an ontology is an engineering artifact
  - with a specific vocabulary
  - with a set of assumptions on the meaning of the terms in the vocabulary

Ontologies 2/3

- In computer science an ontology is an engineering artifact
  - with a specific vocabulary
  - with a set of assumptions on the meaning of the terms in the vocabulary
- shared understanding of a specific domain

Ontologies 2/3

- In computer science an ontology is an engineering artifact
  - with a specific vocabulary
  - with a set of assumptions on the meaning of the terms in the vocabulary
- shared understanding of a specific domain
- formal

Ontologies 2/3

- In computer science an ontology is an engineering artifact
  - with a specific vocabulary
  - with a set of assumptions on the meaning of the terms in the vocabulary
- shared understanding of a specific domain
- formal
- from the perspective of artificial intelligence
  - ontology = knowledge base (logical theory)

Ontologies 2/3

- In computer science an ontology is an engineering artifact
  - with a specific vocabulary
  - with a set of assumptions on the meaning of the terms in the vocabulary
- shared understanding of a specific domain
- formal
- from the perspective of artificial intelligence
  - ontology = knowledge base (logical theory)
- from the perspective of data bases
  - ontology = conceptual data model

WeST People and Knowledge Networks

Ontologies 3/3

Examples

- Biology: Inheritance hierarchies

Ontologies 3/3

Examples

- Biology: Inheritance hierarchies
- Life sciences:
  - disease classifications,
  - anatomy
  - genetics
  - proteins

Ontologies 3/3

Examples

- Biology: Inheritance hierarchies
- Life sciences:
  - disease classifications,
  - anatomy
  - genetics
  - proteins
- Linguistics: word relations, synonyms, metonyms, etc. (WordNet)

Ontologies 3/3

Examples

- Biology: Inheritance hierarchies
- Life sciences:
  - disease classifications,
  - anatomy
  - genetics
  - proteins
- Linguistics: word relations, synonyms, metonyms, etc. (WordNet)
- Economics: process modeling

Structure of an ontology

Ontologies are comprised of different components:

- Individuals, entities, instances, objects: *Tweety*, *Hammer07*

Structure of an ontology

Ontologies are comprised of different components:

- Individuals, entities, instances, objects: *Tweety*, *Hammer07*
- Classes, concepts: *Bird*, *Tool*

Structure of an ontology

Ontologies are comprised of different components:

- Individuals, entities, instances, objects: *Tweety*, *Hammer07*
- Classes, concepts: *Bird*, *Tool*
- Attributes, properties: *hasColor*

Structure of an ontology

Ontologies are comprised of different components:

- Individuals, entities, instances, objects: *Tweety*, *Hammer07*
- Classes, concepts: *Bird*, *Tool*
- Attributes, properties: *hasColor*
- Relations: *friendOf*, *knows*

Structure of an ontology

Ontologies are comprised of different components:

- Individuals, entities, instances, objects: *Tweety*, *Hammer07*
- Classes, concepts: *Bird*, *Tool*
- Attributes, properties: *hasColor*
- Relations: *friendOf*, *knows*
- Functional relations: *fatherOf*

Structure of an ontology

Ontologies are comprised of different components:

- Individuals, entities, instances, objects: *Tweety*, *Hammer07*
- Classes, concepts: *Bird*, *Tool*
- Attributes, properties: *hasColor*
- Relations: *friendOf*, *knows*
- Functional relations: *fatherOf*
- Axioms: *Bird isA Animal*

Ontology languages

- Graphical languages
  - Inheritance networks

Ontology languages

- Graphical languages
  - Inheritance networks
  - Semantic networks

Ontology languages

- Graphical languages
  - Inheritance networks
  - Semantic networks
- Logic-based languages
  - Description logics ($\mathcal{ALC}$, $\mathcal{SHOIN}$, OIL, OWL)

Ontology languages

- Graphical languages
  - Inheritance networks
  - Semantic networks
- Logic-based languages
  - Description logics ($\mathcal{ALC}$, $\mathcal{SHOIN}$, OIL, OWL)
  - Rule languages (Answer set programming)

Ontology languages

- Graphical languages
  - Inheritance networks
  - Semantic networks
- Logic-based languages
  - Description logics ($\mathcal{ALC}$, $\mathcal{SHOIN}$, OIL, OWL)
  - Rule languages (Answer set programming)
  - First-order logic

Ontology languages

- Graphical languages
  - Inheritance networks
  - Semantic networks
- Logic-based languages
  - Description logics ($\mathcal{ALC}$, $\mathcal{SHOIN}$, OIL, OWL)
  - Rule languages (Answer set programming)
  - First-order logic
  - Non-classical logics

Ontology languages

- Graphical languages
  - Inheritance networks
  - Semantic networks
- Logic-based languages
  - Description logics ($\mathcal{ALC}$, $\mathcal{SHOIN}$, OIL, OWL)
  - Rule languages (Answer set programming)
  - First-order logic
  - Non-classical logics
- Object-oriented languages
  - UML

Ontology languages

- Graphical languages
  - Inheritance networks
  - Semantic networks
- Logic-based languages
  - Description logics ($\mathcal{ALC}$, $\mathcal{SHOIN}$, OIL, OWL)
  - Rule languages (Answer set programming)
  - First-order logic
  - Non-classical logics
- Object-oriented languages
  - UML
  - Z

WeST People and Knowledge Networks

Ontology languages

- Graphical languages
  - Inheritance networks
  - Semantic networks
- Logic-based languages
  - Description logics ($\mathcal{ALC}$, $\mathcal{SHOIN}$, OIL, OWL)
  - Rule languages (Answer set programming)
  - First-order logic
  - Non-classical logics
- Object-oriented languages
  - UML
  - Z
- Web schema languages
  - XML/XMLS (not an ontology language per se)

WeST People and Knowledge Networks

Ontology languages

- Graphical languages
  - Inheritance networks
  - Semantic networks
- Logic-based languages
  - Description logics ($\mathcal{ALC}$, $\mathcal{SHOIN}$, OIL, OWL)
  - Rule languages (Answer set programming)
  - First-order logic
  - Non-classical logics
- Object-oriented languages
  - UML
  - Z
- Web schema languages
  - XML/XMLS (not an ontology language per se)
  - RDF/RDFS

Ontology engineering

- An ontology (or any other form of knowledge base)
  - can be used for reasoning, answering queries

Ontology engineering

- An ontology (or any other form of knowledge base)
  - can be used for reasoning, answering queries
  - should be reusable

Ontology engineering

- An ontology (or any other form of knowledge base)
  - can be used for reasoning, answering queries
  - should be reusable
- Process of building an ontology is called *ontology engineering*

Ontology engineering

- An ontology (or any other form of knowledge base)
  - can be used for reasoning, answering queries
  - should be reusable
- Process of building an ontology is called *ontology engineering*
  1. Define language
     - terms
     - concepts,
     - relations, etc.

Ontology engineering

- An ontology (or any other form of knowledge base)
  - can be used for reasoning, answering queries
  - should be reusable
- Process of building an ontology is called *ontology engineering*
  1. Define language
     - terms
     - concepts,
     - relations, etc.
  2. Define knowledge
     - What are equivalent concepts?
     - Are there subset relations?
     - Which constraints have to be imposed? etc.

Ontology engineering

- An ontology (or any other form of knowledge base)
  - can be used for reasoning, answering queries
  - should be reusable
- Process of building an ontology is called *ontology engineering*
  1. Define language
     - terms
     - concepts,
     - relations, etc.
  2. Define knowledge
     - What are equivalent concepts?
     - Are there subset relations?
     - Which constraints have to be imposed? etc.
  3. Use ontology for reasoning

WeST People and Knowledge Networks

Ontology engineering

- An ontology (or any other form of knowledge base)
  - can be used for reasoning, answering queries
  - should be reusable
- Process of building an ontology is called *ontology engineering*
  1. Define language
     - terms
     - concepts,
     - relations, etc.
  2. Define knowledge
     - What are equivalent concepts?
     - Are there subset relations?
     - Which constraints have to be imposed? etc.
  3. Use ontology for reasoning

→ more on that later

Outline

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Overview 1/2

Ontologies can be particularly well described using *description logics*.

We use $\mathcal{ALC}$ (*Attributive Language with Complements*) as an example (Schmidt-Schauß, Smolka; 1991).

Description logic $\mathcal{ALC}$ - Overview 2/2

Languages building on the logic $\mathcal{ALC}$ use a strict grammar to define their syntax.

Description logic $\mathcal{ALC}$ - Overview 2/2

Languages building on the logic $\mathcal{ALC}$ use a strict grammar to define their syntax.

The basic ingredients for the definition of a logical language are

1. Definition of the *signature*:
   - What are the atomic elements?

Description logic $\mathcal{ALC}$ - Overview 2/2

Languages building on the logic $\mathcal{ALC}$ use a strict grammar to define their syntax.

The basic ingredients for the definition of a logical language are

1. Definition of the *signature*:
   - What are the atomic elements?
2. Definition of the *syntax*:
   - How are complex formulas built from atomic elements?

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Overview 2/2

Languages building on the logic $\mathcal{ALC}$ use a strict grammar to define their syntax.

The basic ingredients for the definition of a logical language are

1. Definition of the *signature*:
   - What are the atomic elements?
2. Definition of the *syntax*:
   - How are complex formulas built from atomic elements?
3. Definition of the *semantics*:
   - What is the meaning of formulas?
   - What can be inferred from formulas?

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Signature

A *signature* $\mathcal{S} = (N_C, N_R, N_O)$ for an $\mathcal{ALC}$ knowledge base contains

Description logic $\mathcal{ALC}$ - Signature

A *signature* $\mathcal{S} = (N_C, N_R, N_O)$ for an $\mathcal{ALC}$ knowledge base contains

- a set N_C of atomic concept names
  - Example: *Human*, *Animal*, *FlyingObject*

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Signature

A *signature* $\mathcal{S} = (N_C, N_R, N_O)$ for an $\mathcal{ALC}$ knowledge base contains

- a set N_C of atomic concept names
  - Example: *Human*, *Animal*, *FlyingObject*
  - We always assume $\top, \bot \in N_C$
  - $\top$: the concept of everything
  - $\bot$: the concept of nothing

Description logic $\mathcal{ALC}$ - Signature

A *signature* $\mathcal{S} = (N_C, N_R, N_O)$ for an $\mathcal{ALC}$ knowledge base contains

- a set N_C of atomic concept names
  - Example: *Human*, *Animal*, *FlyingObject*
  - We always assume $\top, \bot \in N_C$
  - $\top$: the concept of everything
  - $\bot$: the concept of nothing
- a set N_R of atomic relation names
  - Example: *friendOf*, *hasColor*, *fatherOf*

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Signature

A *signature* $\mathcal{S} = (N_C, N_R, N_O)$ for an $\mathcal{ALC}$ knowledge base contains

- a set N_C of atomic concept names
  - Example: *Human*, *Animal*, *FlyingObject*
  - We always assume $\top, \bot \in N_C$
  - $\top$: the concept of everything
  - $\bot$: the concept of nothing
- a set N_R of atomic relation names
  - Example: *friendOf*, *hasColor*, *fatherOf*
  - *Remark*: properties, relations, and functional expressions are indistinguishable in most DLs

Description logic $\mathcal{ALC}$ - Signature

A *signature* $\mathcal{S} = (N_C, N_R, N_O)$ for an $\mathcal{ALC}$ knowledge base contains

- a set N_C of atomic concept names
  - Example: *Human*, *Animal*, *FlyingObject*
  - We always assume $\top, \bot \in N_C$
  - $\top$: the concept of everything
  - $\bot$: the concept of nothing
- a set N_R of atomic relation names
  - Example: *friendOf*, *hasColor*, *fatherOf*
  - *Remark*: properties, relations, and functional expressions are indistinguishable in most DLs
- a set N_O of individual names
  - Example: *Carl*, *Dave*, *Hammer07*

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Syntax

The syntax of a logic describes how complex formulas are build from atomic elements.

Description logic $\mathcal{ALC}$ - Syntax

The syntax of a logic describes how complex formulas are build from atomic elements.

$\mathcal{ALC}$ uses the following *connectives* to combine atomic elements:

$\sqcap$	intersection or conjunction of concepts
$\sqcup$	union or disjunction of concepts
$\neg$	complement of a concept
$\forall$	universal restriction
$\exists$	existential restriction
$\sqsubseteq$	concept inclusion
$:$	concept/role assertion

Description logic $\mathcal{ALC}$ - Concepts 1/2

Let $\mathcal{S} = (N_C, N_R, N_O)$ be an $\mathcal{ALC}$ signature.

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Concepts 1/2

Let $\mathcal{S} = (N_C, N_R, N_O)$ be an $\mathcal{ALC}$ signature.

The set $\mathcal{C}_\mathcal{S}$ of valid *concepts* of $\mathcal{S}$ is the minimal set $\mathcal{C}$ satisfying

1. $N_C \subseteq \mathcal{C}$
   - every atomic concept is a concept

Description logic $\mathcal{ALC}$ - Concepts 1/2

Let $\mathcal{S} = (N_C, N_R, N_O)$ be an $\mathcal{ALC}$ signature.

The set $\mathcal{C}_\mathcal{S}$ of valid *concepts* of $\mathcal{S}$ is the minimal set $\mathcal{C}$ satisfying

1. $N_C \subseteq \mathcal{C}$
   - every atomic concept is a concept
2. for every $C_1, C_2 \in \mathcal{C}$ we have $C_1 \sqcap C_2 \in \mathcal{C}$

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Concepts 1/2

Let $\mathcal{S} = (N_C, N_R, N_O)$ be an $\mathcal{ALC}$ signature.

The set $\mathcal{C}_\mathcal{S}$ of valid *concepts* of $\mathcal{S}$ is the minimal set $\mathcal{C}$ satisfying

1. $N_C \subseteq \mathcal{C}$
   - every atomic concept is a concept
2. for every $C_1, C_2 \in \mathcal{C}$ we have $C_1 \sqcap C_2 \in \mathcal{C}$
   - the intersection of two concepts is a concept as well
   - Example: Female $\sqcap$ Architect (the concept of all female architects)

Description logic $\mathcal{ALC}$ - Concepts 1/2

Let $\mathcal{S} = (N_C, N_R, N_O)$ be an $\mathcal{ALC}$ signature.

The set $\mathcal{C}_\mathcal{S}$ of valid *concepts* of $\mathcal{S}$ is the minimal set $\mathcal{C}$ satisfying

1. $N_C \subseteq \mathcal{C}$
   - every atomic concept is a concept
2. for every $C_1, C_2 \in \mathcal{C}$ we have $C_1 \sqcap C_2 \in \mathcal{C}$
   - the intersection of two concepts is a concept as well
   - Example: Female $\sqcap$ Architect (the concept of all female architects)
3. for every $C_1, C_2 \in \mathcal{C}$ we have $C_1 \sqcup C_2 \in \mathcal{C}$

Description logic $\mathcal{ALC}$ - Concepts 1/2

Let $\mathcal{S} = (N_C, N_R, N_O)$ be an $\mathcal{ALC}$ signature.

The set $\mathcal{C}_\mathcal{S}$ of valid *concepts* of $\mathcal{S}$ is the minimal set $\mathcal{C}$ satisfying

1. $N_C \subseteq \mathcal{C}$
   - every atomic concept is a concept
2. for every $C_1, C_2 \in \mathcal{C}$ we have $C_1 \sqcap C_2 \in \mathcal{C}$
   - the intersection of two concepts is a concept as well
   - Example: Female $\sqcap$ Architect (the concept of all female architects)
3. for every $C_1, C_2 \in \mathcal{C}$ we have $C_1 \sqcup C_2 \in \mathcal{C}$
   - the union of two concepts is a concept as well
   - Example: Student $\sqcup$ Professor (the concept of all people who are students or professors)

Description logic $\mathcal{ALC}$ - Concepts 2/2

4. for every $C \in \mathcal{C}$ we have $\neg C \in \mathcal{C}$

Description logic $\mathcal{ALC}$ - Concepts 2/2

4. for every $C \in \mathcal{C}$ we have $\neg C \in \mathcal{C}$
   - the complement of a concept is a concept as well
   - Example: $\neg$Student (the concept of all people who are not a student)

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Concepts 2/2

4. for every $C \in \mathcal{C}$ we have $\neg C \in \mathcal{C}$
   - the complement of a concept is a concept as well
   - Example: ¬Student (the concept of all people who are not a student)
5. for every $C \in \mathcal{C}, R \in N_R$ we have $\forall R.C \in \mathcal{C}$

Description logic $\mathcal{ALC}$ - Concepts 2/2

4. for every $C \in \mathcal{C}$ we have $\neg C \in \mathcal{C}$
   - the complement of a concept is a concept as well
   - Example: ¬Student (the concept of all people who are not a student)
5. for every $C \in \mathcal{C}, R \in N_R$ we have $\forall R.C \in \mathcal{C}$
   - The set of individuals whose R-successors are in C is a concept
   - Example: ∀hasChild.Female
     (the concept of all people who *only* have daughters)

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Concepts 2/2

4. for every $C \in \mathcal{C}$ we have $\neg C \in \mathcal{C}$
   - the complement of a concept is a concept as well
   - Example: $\neg$Student (the concept of all people who are not a student)
5. for every $C \in \mathcal{C}, R \in N_R$ we have $\forall R.C \in \mathcal{C}$
   - The set of individuals whose R-successors are in C is a concept
   - Example: $\forall$hasChild.Female
     (the concept of all people who *only* have daughters)
6. for every $C \in \mathcal{C}, R \in N_R$ we have $\exists R.C \in \mathcal{C}$

Description logic $\mathcal{ALC}$ - Concepts 2/2

4. for every $C \in \mathcal{C}$ we have $\neg C \in \mathcal{C}$
   - the complement of a concept is a concept as well
   - Example: ¬Student (the concept of all people who are not a student)
5. for every $C \in \mathcal{C}, R \in N_R$ we have $\forall R.C \in \mathcal{C}$
   - The set of individuals whose R-successors are in C is a concept
   - Example: ∀hasChild.Female
     (the concept of all people who *only* have daughters)
6. for every $C \in \mathcal{C}, R \in N_R$ we have $\exists R.C \in \mathcal{C}$
   - The set of individuals where one R-successor is in C is a concept
   - Example: ∃visited.GermanCity
     (the concept of all people who visited *some* German city)

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Axioms 1/2

$\mathcal{ALC}$ distinguishes between two types of axioms (formulas, sentences):

1. Terminological axioms: for general statements about concepts and roles

Description logic $\mathcal{ALC}$ - Axioms 1/2

$\mathcal{ALC}$ distinguishes between two types of axioms (formulas, sentences):

1. Terminological axioms: for general statements about concepts and roles
2. Assertional axioms: for specific statements about individuals

Description logic $\mathcal{ALC}$ - Axioms 1/2

$\mathcal{ALC}$ distinguishes between two types of axioms (formulas, sentences):

1. Terminological axioms: for general statements about concepts and roles
2. Assertional axioms: for specific statements about individuals

The set of *terminological axioms* $\mathfrak{T}_\mathcal{S}$ of $\mathcal{S}$ is the minimal set $\mathfrak{T}$ satisfying

- for every $C_1, C_2 \in \mathcal{C}_\mathcal{S}$ we have $C_1 \sqsubseteq C_2 \in \mathfrak{T}$
  - C_1 is a subconcept of C_2 ("C_2 subsumes C_1")
  - Example: Woman $\sqsubseteq$ Human (every woman is a human)

Description logic $\mathcal{ALC}$ - Axioms 2/2

The set of *assertional axioms* $\mathfrak{A}_{\mathcal{S}}$ of $\mathcal{S}$ is the minimal set $\mathfrak{A}$ satisfying

- for every $C \in \mathcal{C}_{\mathcal{S}}$ and $t \in N_O$ we have $t : C \in \mathfrak{A}$
  - t satisfies concept C ("t is an element of concept C")
  - Example: *Carl* : *Human* ("Carl is a human")

Description logic $\mathcal{ALC}$ - Axioms 2/2

The set of *assertional axioms* $\mathfrak{A}_{\mathcal{S}}$ of $\mathcal{S}$ is the minimal set $\mathfrak{A}$ satisfying

- for every $C \in \mathcal{C}_{\mathcal{S}}$ and $t \in N_O$ we have $t : C \in \mathfrak{A}$
  - t satisfies concept C ("t is an element of concept C")
  - Example: *Carl* : *Human* ("Carl is a human")
- for every $R \in N_R$ and $t, t' \in N_O$ we have $(t, t') : R \in \mathfrak{A}$
  - t and t' are in relation R
  - Example: $(Carl, Dave) : fatherOf$ ("Carl is the father of Dave")

Description logic $\mathcal{ALC}$ - Knowledge base

A *knowledge base* contains the axioms explicitly accepted to be true.

In $\mathcal{ALC}$ a knowledge base is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with

- $\mathcal{T} \subseteq \mathfrak{T}_{\mathcal{S}}$ is a (finite) set of terminological axioms ($\mathcal{T}$ is also called TBox)
- $\mathcal{A} \subseteq \mathfrak{A}_{\mathcal{S}}$ is a (finite) set of assertional axioms ($\mathcal{A}$ is also called ABox)

Example

Let $\mathcal{S}_1 = (N_C, N_R, N_O)$ be the signature defined via

$$N_C = \{\text{Human}, \text{Man}, \text{Woman}, \text{Architect}, \text{Father}\}$$
$$N_R = \{\text{hasChild}, \text{hasJob}\}$$
$$N_O = \{\text{Carl}, \text{Dave}, \text{Anna}\}$$

Example

Let $\mathcal{S}_1 = (N_C, N_R, N_O)$ be the signature defined via

$$N_C = \{\text{Human}, \text{Man}, \text{Woman}, \text{Architect}, \text{Father}\}$$
$$N_R = \{\text{hasChild}, \text{hasJob}\}$$
$$N_O = \{\text{Carl}, \text{Dave}, \text{Anna}\}$$

Consider $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A}_1)$ given by

$$\begin{aligned}
\mathcal{T}_1 = \{ \quad & \text{Architect} \sqsubseteq \text{Human} \\
& \text{Man} \sqcup \text{Woman} \sqsubseteq \text{Human} \\
& \text{Human} \sqsubseteq \forall \text{hasChild}.\text{Human} \\
& \text{Father} \sqsubseteq \text{Man} \sqcap \exists \text{hasChild}.\top \quad \} \\
\mathcal{A}_1 = \{ \quad & \text{Dave} : \text{Architect}, \text{Anna} : \text{Woman}, \\
& \text{Carl} : \text{Father}, (\text{Dave}, \text{Anna}) : \text{hasChild} \quad \}
\end{aligned}$$

Example

Let $\mathcal{S}_1 = (N_C, N_R, N_O)$ be the signature defined via

$$N_C = \{\text{Human}, \text{Man}, \text{Woman}, \text{Architect}, \text{Father}\}$$
$$N_R = \{\text{hasChild}, \text{hasJob}\}$$
$$N_O = \{\text{Carl}, \text{Dave}, \text{Anna}\}$$

Consider $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A}_1)$ given by

$$\begin{aligned}
\mathcal{T}_1 = \{ \quad & \text{Architect} \sqsubseteq \text{Human} \\
& \text{Man} \sqcup \text{Woman} \sqsubseteq \text{Human} \\
& \text{Human} \sqsubseteq \forall \text{hasChild}.\text{Human} \\
& \text{Father} \sqsubseteq \text{Man} \sqcap \exists \text{hasChild}.\top \quad \} \\
\mathcal{A}_1 = \{ \quad & \text{Dave} : \text{Architect}, \text{Anna} : \text{Woman}, \\
& \text{Carl} : \text{Father}, (\text{Dave}, \text{Anna}) : \text{hasChild} \quad \}
\end{aligned}$$

- What is the meaning of the concepts above?
- What can be "inferred" from the knowledge base?

Description logic $\mathcal{ALC}$ - Semantics 1/4

Semantics specify the *meaning* of concepts and axioms with respect to the “real world”.

Description logic $\mathcal{ALC}$ - Semantics 1/4

Semantics specify the *meaning* of concepts and axioms with respect to the "real world".

Let $\mathcal{S} = (N_C, N_R, N_O)$ be a signature. An *interpretation* $\mathcal{I}$ for $\mathcal{S}$ is a tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ with

1. $\Delta^{\mathcal{I}}$ is a non-empty set called the *domain* or the *universe*
2. $\cdot^{\mathcal{I}}$ is the *interpretation function* which maps
   - every $t \in N_O$ to an element $t^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
   - every $C \in N_C$ to a set $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
   - every $R \in N_R$ to a set $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

Description logic $\mathcal{ALC}$ - Semantics 1/4

Semantics specify the *meaning* of concepts and axioms with respect to the "real world".

Let $\mathcal{S} = (N_C, N_R, N_O)$ be a signature. An *interpretation* $\mathcal{I}$ for $\mathcal{S}$ is a tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ with

1. $\Delta^{\mathcal{I}}$ is a non-empty set called the *domain* or the *universe*
2. $\cdot^{\mathcal{I}}$ is the *interpretation function* which maps
   - every $t \in N_O$ to an element $t^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
   - every $C \in N_C$ to a set $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
   - every $R \in N_R$ to a set $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ describes a *possible world*, i. e. a complete description of the things that are true in some assumed situation.

Let $\mathbb{I}_{\mathcal{S}}$ denote the set of all interpretations for $\mathcal{S}$.

WeST People and Knowledge Networks

Let $\mathcal{S} = (N_C, N_R, N_O)$ be a signature and $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ an interpretation.

We abbreviate

$$\begin{aligned}
\top^\mathcal{I} &= \Delta^\mathcal{I} \\
\bot^\mathcal{I} &= \emptyset \\
(C_1 \sqcup C_2)^\mathcal{I} &= C_1^\mathcal{I} \cup C_2^\mathcal{I} \\
(C_1 \sqcap C_2)^\mathcal{I} &= C_1^\mathcal{I} \cap C_2^\mathcal{I} \\
(\neg C)^\mathcal{I} &= \Delta^\mathcal{I} \setminus C^\mathcal{I} \\
(\forall R.C)^\mathcal{I} &= \{t \in \Delta^\mathcal{I} \mid \text{for all } t' \text{ with } (t,t') \in R^\mathcal{I} \text{ it is } t' \in C^\mathcal{I}\} \\
(\exists R.C)^\mathcal{I} &= \{t \in \Delta^\mathcal{I} \mid \text{there is } t' \text{ with } (t,t') \in R^\mathcal{I} \text{ and } t' \in C^\mathcal{I}\}
\end{aligned}$$

Description logic $\mathcal{ALC}$ - Semantics 3/4

Let $\mathcal{S} = (N_C, N_R, N_O)$ be a signature and $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ an interpretation.

$\mathcal{I}$ satisfies an axiom A, written $\mathcal{I} \models A$, if

- $\mathcal{I} \models C_1 \sqsubseteq C_2$ if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Semantics 3/4

Let $\mathcal{S} = (N_C, N_R, N_O)$ be a signature and $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ an interpretation.

$\mathcal{I}$ satisfies an axiom A, written $\mathcal{I} \models A$, if

- $\mathcal{I} \models C_1 \sqsubseteq C_2$ if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
- $\mathcal{I} \models t : C$ if $t^{\mathcal{I}} \in C^{\mathcal{I}}$

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Semantics 3/4

Let $\mathcal{S} = (N_C, N_R, N_O)$ be a signature and $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ an interpretation.

$\mathcal{I}$ satisfies an axiom A, written $\mathcal{I} \models A$, if

- $\mathcal{I} \models C_1 \sqsubseteq C_2$ if $C_1^\mathcal{I} \subseteq C_2^\mathcal{I}$
- $\mathcal{I} \models t : C$ if $t^\mathcal{I} \in C^\mathcal{I}$
- $\mathcal{I} \models (t_1, t_2) : R$ if $(t_1^\mathcal{I}, t_2^\mathcal{I}) \in R^\mathcal{I}$

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfies a set of axioms $\mathcal{F} \subseteq \mathfrak{T}_{\mathcal{S}} \cup \mathfrak{A}_{\mathcal{S}}$, written $\mathcal{I} \models \mathcal{F}$ if

$$\forall A \in \mathcal{F} : \mathcal{I} \models A$$

Description logic $\mathcal{ALC}$ - Semantics 4/4

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfies a set of axioms $\mathcal{F} \subseteq \mathfrak{T}_{\mathcal{S}} \cup \mathfrak{A}_{\mathcal{S}}$, written $\mathcal{I} \models \mathcal{F}$ if

$$\forall A \in \mathcal{F} : \mathcal{I} \models A$$

An interpretation $\mathcal{I}$ satisfies a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, written $\mathcal{I} \models \mathcal{K}$ if

- $\mathcal{I} \models \mathcal{T}$
- $\mathcal{I} \models \mathcal{A}$

If $\mathcal{I} \models \mathcal{K}$ then $\mathcal{I}$ is called a *model* of $\mathcal{K}$.

WeST People and Knowledge Networks

Example 1/2

Let $\mathcal{S}_1 = (N_C, N_R, N_O)$ be the signature defined via

$$N_C = \{\text{Human}, \text{Man}, \text{Woman}, \text{Architect}, \text{Father}\}$$

$$N_R = \{\text{hasChild}, \text{hasJob}\}$$

$$N_O = \{\text{Carl}, \text{Dave}, \text{Anna}\}$$

Example 1/2

Let $\mathcal{S}_1 = (N_C, N_R, N_O)$ be the signature defined via

$$N_C = \{\text{Human}, \text{Man}, \text{Woman}, \text{Architect}, \text{Father}\}$$
$$N_R = \{\text{hasChild}, \text{hasJob}\}$$
$$N_O = \{\text{Carl}, \text{Dave}, \text{Anna}\}$$

Consider the interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ defined via

$$\begin{aligned}
\Delta^{\mathcal{I}} &= \{ \quad \text{CarlCarlson}, \text{DavidDavidson}, \text{AnnaAndrews} \; \} \\
\text{Carl}^{\mathcal{I}} &= \quad\;\; \text{CarlCarlson} \qquad \ldots \\
\text{Human}^{\mathcal{I}} &= \{ \quad \text{CarlCarlson}, \text{DavidDavidson}, \text{AnnaAndrews} \; \} \\
\text{Man}^{\mathcal{I}} &= \{ \quad \text{CarlCarlson}, \text{DavidDavidson} \; \} \\
\text{Father}^{\mathcal{I}} &= \{ \quad \text{CarlCarlson}, \text{DavidDavidson} \; \} \\
\text{Woman}^{\mathcal{I}} &= \{ \quad (\text{AnnaAndrews}) \; \} \\
\text{hasChild}^{\mathcal{I}} &= \{ \quad (\text{DavidDavidson}, \text{AnnaAndrews}), (\text{CarlCarlson}, \text{DavidDavidson}) \; \} \\
\text{hasJob}^{\mathcal{I}} &= \{ \quad (\text{DavidDavidson}, \text{Architect}) \; \}
\end{aligned}$$

Example 2/2

Some observations:

$$(\text{Man} \sqcap \text{Woman})^{\mathcal{I}} = \text{Man}^{\mathcal{I}} \cap \text{Woman}^{\mathcal{I}} = \emptyset$$

$$\begin{aligned}(\text{Man} \sqcup \text{Woman})^{\mathcal{I}} &= \text{Man}^{\mathcal{I}} \cup \text{Woman}^{\mathcal{I}} \\ &= \{\text{CarlCarlson}, \text{DavidDavidson}, \text{AnnaAndrews}\}\end{aligned}$$

$$(\text{Man} \sqcap \exists \text{hasChild}.\top)^{\mathcal{I}} = \text{Man}^{\mathcal{I}} \cap (\exists \text{hasChild}.\top)^{\mathcal{I}}$$

$$= \text{Man}^{\mathcal{I}} \cap \{t \in \Delta^{\mathcal{I}} \mid \text{there is } t' \text{ with } (t, t') \in \text{hasChild}^{\mathcal{I}} \text{ and } t' \in C^{\mathcal{I}}$$

$$= \{\text{CarlCarlson}, \text{DavidDavidson}\}$$

Example 2/2

Some observations:

$$(\mathsf{Man} \sqcap \mathsf{Woman})^{\mathcal{I}} = \mathsf{Man}^{\mathcal{I}} \cap \mathsf{Woman}^{\mathcal{I}} = \emptyset$$

$$(\mathsf{Man} \sqcup \mathsf{Woman})^{\mathcal{I}} = \mathsf{Man}^{\mathcal{I}} \cup \mathsf{Woman}^{\mathcal{I}}$$

$$= \{\mathsf{CarlCarlson}, \mathsf{DavidDavidson}, \mathsf{AnnaAndrews}\}$$

$$(\mathsf{Man} \sqcap \exists \mathsf{hasChild}.\top)^{\mathcal{I}} = \mathsf{Man}^{\mathcal{I}} \cap (\exists \mathsf{hasChild}.\top)^{\mathcal{I}}$$

$$= \mathsf{Man}^{\mathcal{I}} \cap \{t \in \Delta^{\mathcal{I}} \mid \text{there is } t' \text{ with } (t, t') \in \mathsf{hasChild}^{\mathcal{I}} \text{ and } t' \in C^{\mathcal{I}}$$

$$= \{\mathsf{CarlCarlson}, \mathsf{DavidDavidson}\}$$

It follows

- $\mathcal{I} \models \mathsf{Father} \sqsubseteq \mathsf{Man} \sqcap \exists \mathsf{hasChild}.\top$
- $\mathcal{I} \models \mathsf{Dave} : \mathsf{Architect}$
- $\mathcal{I} \models \mathsf{Anna} : \mathsf{Woman}$

Description logic $\mathcal{ALC}$ - Inferences 1/2

Inference describes the process from drawing new conclusions from a knowledge base.

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Inferences 1/2

Inference describes the process from drawing new conclusions from a knowledge base.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base. We say that $\mathcal{K}$ *entails* an axiom $A \in \mathfrak{T}_{\mathcal{S}} \cup \mathfrak{A}_{\mathcal{S}}$, written $\mathcal{K} \models A$, if

> *for every interpretation* $\mathcal{I} \in \mathbb{I}_{\mathcal{S}}$ *we have* $\mathcal{I} \models \mathcal{K}$ *implies* $\mathcal{I} \models A$

Description logic $\mathcal{ALC}$ - Inferences 1/2

Inference describes the process from drawing new conclusions from a knowledge base.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base. We say that $\mathcal{K}$ *entails* an axiom $A \in \mathfrak{T}_{\mathcal{S}} \cup \mathfrak{A}_{\mathcal{S}}$, written $\mathcal{K} \models A$, if

> *for every interpretation $\mathcal{I} \in \mathbb{I}_{\mathcal{S}}$ we have $\mathcal{I} \models \mathcal{K}$ implies $\mathcal{I} \models A$*

Note that for every $C_1 \sqsubseteq C_2 \in \mathcal{T}$ we automatically have $\mathcal{K} \models C_1 \sqsubseteq C_2$ (the same is true for the ABox $\mathcal{A}$).

Description logic $\mathcal{ALC}$ - Inferences 2/2

In description logics one usually distinguishes the following *inference tasks*:

- Subsumption problem: Given concepts C_1, C_2 does $\mathcal{K}$ entail $C_1 \sqsubseteq C_2$?

WeST People and Knowledge Networks

Description logic $\mathcal{ALC}$ - Inferences 2/2

In description logics one usually distinguishes the following *inference tasks*:

- Subsumption problem: Given concepts C_1, C_2 does $\mathcal{K}$ entail $C_1 \sqsubseteq C_2$?
- Instance checking problem: Given concept C and individual t does $\mathcal{K}$ entail $t : C$?

Description logic $\mathcal{ALC}$ - Inferences 2/2

In description logics one usually distinguishes the following *inference tasks*:

- Subsumption problem: Given concepts C_1, C_2 does $\mathcal{K}$ entail $C_1 \sqsubseteq C_2$?
- Instance checking problem: Given concept C and individual t does $\mathcal{K}$ entail $t : C$?
- Relation checking problem: Given relation R and individuals t, t' does $\mathcal{K}$ entail $(t, t') : R$?

Description logic $\mathcal{ALC}$ - Inferences 2/2

In description logics one usually distinguishes the following *inference tasks*:

- Subsumption problem: Given concepts C_1, C_2 does $\mathcal{K}$ entail $C_1 \sqsubseteq C_2$?
- Instance checking problem: Given concept C and individual t does $\mathcal{K}$ entail $t : C$?
- Relation checking problem: Given relation R and individuals t, t' does $\mathcal{K}$ entail $(t, t') : R$?
- Consistency problem: Is there $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$?

Description logic $\mathcal{ALC}$ - Inferences 2/2

In description logics one usually distinguishes the following *inference tasks*:

- Subsumption problem: Given concepts C_1, C_2 does $\mathcal{K}$ entail $C_1 \sqsubseteq C_2$?
- Instance checking problem: Given concept C and individual t does $\mathcal{K}$ entail $t : C$?
- Relation checking problem: Given relation R and individuals t, t' does $\mathcal{K}$ entail $(t, t') : R$?
- Consistency problem: Is there $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$?

$\rightarrow$ more on that next time

WeST People and Knowledge Networks

Example

Consider again $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A}_1)$ given by

$$\begin{aligned}
\mathcal{T}_1 = \{ \quad & \text{Architect} \sqsubseteq \text{Human} \\
& \text{Man} \sqcup \text{Woman} \sqsubseteq \text{Human} \\
& \text{Human} \sqsubseteq \forall \text{hasChild.Human} \\
& \text{Father} \sqsubseteq \text{Man} \sqcap \exists \text{hasChild.}\top \quad \} \\
\mathcal{A}_1 = \{ \quad & \text{Dave} : \text{Architect}, \text{Anna} : \text{Woman}, \\
& \text{Carl} : \text{Father}, (\text{Dave}, \text{Anna}) : \text{hasChild} \quad \}
\end{aligned}$$

Here we have

- $\mathcal{K} \models \text{Dave} : \text{Human}$

Example

Consider again $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A}_1)$ given by

$$
\begin{aligned}
\mathcal{T}_1 = \{ \quad & \text{Architect} \sqsubseteq \text{Human} \\
& \text{Man} \sqcup \text{Woman} \sqsubseteq \text{Human} \\
& \text{Human} \sqsubseteq \forall \text{hasChild.Human} \\
& \text{Father} \sqsubseteq \text{Man} \sqcap \exists \text{hasChild.}\top \quad \} \\
\mathcal{A}_1 = \{ \quad & \text{Dave} : \text{Architect}, \text{Anna} : \text{Woman}, \\
& \text{Carl} : \text{Father}, (\text{Dave}, \text{Anna}) : \text{hasChild} \quad \}
\end{aligned}
$$

Here we have

- $\mathcal{K} \models \text{Dave} : \text{Human}$
- $\mathcal{K} \models \text{Carl} : \exists \text{hasChild.Human}$

Example

Consider again $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A}_1)$ given by

$$\begin{aligned}
\mathcal{T}_1 = \{ \quad & \mathsf{Architect} \sqsubseteq \mathsf{Human} \\
& \mathsf{Man} \sqcup \mathsf{Woman} \sqsubseteq \mathsf{Human} \\
& \mathsf{Human} \sqsubseteq \forall \mathsf{hasChild}.\mathsf{Human} \\
& \mathsf{Father} \sqsubseteq \mathsf{Man} \sqcap \exists \mathsf{hasChild}.\top \quad \} \\
\mathcal{A}_1 = \{ \quad & \mathsf{Dave} : \mathsf{Architect}, \mathsf{Anna} : \mathsf{Woman}, \\
& \mathsf{Carl} : \mathsf{Father}, (\mathsf{Dave}, \mathsf{Anna}) : \mathsf{hasChild} \quad \}
\end{aligned}$$

Here we have

- $\mathcal{K} \models \mathsf{Dave} : \mathsf{Human}$
- $\mathcal{K} \models \mathsf{Carl} : \exists \mathsf{hasChild}.\mathsf{Human}$
- $\mathcal{K} \not\models \mathsf{Man} \sqcap \mathsf{Female} \sqsubseteq \bot$ (!)

$\mathcal{ALC}$ and first-order logic 1/2

- Description logics have been invented as a modeling language for taxonomic/ontological knowledge

$\mathcal{ALC}$ and first-order logic 1/2

- Description logics have been invented as a modeling language for taxonomic/ontological knowledge
- Formally, most of them are (decidable) fragments of first-order logic

$\mathcal{ALC}$ and first-order logic 1/2

- Description logics have been invented as a modeling language for taxonomic/ontological knowledge
- Formally, most of them are (decidable) fragments of first-order logic

Comparison of notation:

$\mathcal{ALC}$ term	FOL term
Concept	(unary) Predicate
Relation	(binary) Predicate
Individual	Individual

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$
$(t, t') : R$	

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$
$(t, t') : R$	$R(t, t')$

WeST People and Knowledge Networks

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$
$(t, t') : R$	$R(t, t')$
$C_1 \sqsubseteq C_2$	

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$
$(t, t') : R$	$R(t, t')$
$C_1 \sqsubseteq C_2$	$\forall X : C_1(X) \Rightarrow C_2(X)$

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$
$(t, t') : R$	$R(t, t')$
$C_1 \sqsubseteq C_2$	$\forall X : C_1(X) \Rightarrow C_2(X)$
$C_1 \sqsubseteq C_2 \sqcap C_3$	

WeST People and Knowledge Networks

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$
$(t, t') : R$	$R(t, t')$
$C_1 \sqsubseteq C_2$	$\forall X : C_1(X) \Rightarrow C_2(X)$
$C_1 \sqsubseteq C_2 \sqcap C_3$	$\forall X : C_1(X) \Rightarrow (C_2(X) \wedge C_3(X))$

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$
$(t, t') : R$	$R(t, t')$
$C_1 \sqsubseteq C_2$	$\forall X : C_1(X) \Rightarrow C_2(X)$
$C_1 \sqsubseteq C_2 \sqcap C_3$	$\forall X : C_1(X) \Rightarrow (C_2(X) \wedge C_3(X))$
$C_1 \sqsubseteq \forall R.C_2$	

WeST People and Knowledge Networks

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$
$(t, t') : R$	$R(t, t')$
$C_1 \sqsubseteq C_2$	$\forall X : C_1(X) \Rightarrow C_2(X)$
$C_1 \sqsubseteq C_2 \sqcap C_3$	$\forall X : C_1(X) \Rightarrow (C_2(X) \wedge C_3(X))$
$C_1 \sqsubseteq \forall R.C_2$	$\forall X : C_1(X) \Rightarrow (\forall Y : R(X, Y) \Rightarrow C_2(Y))$

WeST People and Knowledge Networks

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$
$(t, t') : R$	$R(t, t')$
$C_1 \sqsubseteq C_2$	$\forall X : C_1(X) \Rightarrow C_2(X)$
$C_1 \sqsubseteq C_2 \sqcap C_3$	$\forall X : C_1(X) \Rightarrow (C_2(X) \wedge C_3(X))$
$C_1 \sqsubseteq \forall R.C_2$	$\forall X : C_1(X) \Rightarrow (\forall Y : R(X, Y) \Rightarrow C_2(Y))$
$C_1 \sqsubseteq \exists R.C_2$	

WeST People and Knowledge Networks

$\mathcal{ALC}$ and first-order logic 2/2

Comparison of formulas

$\mathcal{ALC}$ axiom	FOL formula
$t : C$	$C(t)$
$(t, t') : R$	$R(t, t')$
$C_1 \sqsubseteq C_2$	$\forall X : C_1(X) \Rightarrow C_2(X)$
$C_1 \sqsubseteq C_2 \sqcap C_3$	$\forall X : C_1(X) \Rightarrow (C_2(X) \wedge C_3(X))$
$C_1 \sqsubseteq \forall R.C_2$	$\forall X : C_1(X) \Rightarrow (\forall Y : R(X, Y) \Rightarrow C_2(Y))$
$C_1 \sqsubseteq \exists R.C_2$	$\forall X : C_1(X) \Rightarrow (\exists Y : R(X, Y) \wedge C_2(Y))$

Outline

Summary

- Ontologies: definition and applications
- Structure of an ontology
- The description logic $\mathcal{ALC}$:
  - Syntax: signature, concepts, relations, axioms
  - Semantic: interpretations, models
  - Inference: entailment
  - $\mathcal{ALC}$ and first-order logics

WeST People and Knowledge Networks

Pointers to further reading

- John F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, 2000.
- Steffen Staab, Rudi Studer (Editors). Handbook on Ontologies, Springer Verlag, 2009.
- The Description Logic Complexity Navigator `http://www.cs.man.ac.uk/~ezolin/dl/`

Exercises

1. Imagine you have to model the domain "University Life". What are suitable concepts, relations, individuals? What axioms hold in this domain?

Exercises

1. Imagine you have to model the domain "University Life". What are suitable concepts, relations, individuals? What axioms hold in this domain?
2. For the knowledge base $\mathcal{K}_1$ (slide 22) find an interpretation other than $\mathcal{I}$ from slide 27 that is a model of $\mathcal{K}_1$.

Exercises

1. Imagine you have to model the domain "University Life". What are suitable concepts, relations, individuals? What axioms hold in this domain?
2. For the knowledge base $\mathcal{K}_1$ (slide 22) find an interpretation other than $\mathcal{I}$ from slide 27 that is a model of $\mathcal{K}_1$.
3. Translate the knowledge base $\mathcal{K}_1$ (slide 22) into first-order logic.

Exercises

1. Imagine you have to model the domain "University Life". What are suitable concepts, relations, individuals? What axioms hold in this domain?
2. For the knowledge base $\mathcal{K}_1$ (slide 22) find an interpretation other than $\mathcal{I}$ from slide 27 that is a model of $\mathcal{K}_1$.
3. Translate the knowledge base $\mathcal{K}_1$ (slide 22) into first-order logic.
4. Consider $\mathcal{K}_2 = (\mathcal{T}_2, \mathcal{A}_2)$ given via

$$\mathcal{T}_2 = \{ \quad B \sqsubseteq D \quad \}$$
$$\mathcal{A}_2 = \{ \quad d : B, (c, d) : S \quad \}$$

Is the entailment $\mathcal{K}_2 \models c : \forall S.D$ valid? (Home assignment)