Logical Aspects of Artificial Intelligence: Knowledge Logics

Stéphane Demri demri@lsv.fr
https://cv.archives-ouvertes.fr/stephane-demri

September 14th, 2022

What is in this part of the course?

Introduction to Description Logics and Temporal Logics for Multi-Agent Systems

- Today: Introduction to description logics (DLs).
- 21/09, 28/09: Introduction (Part II), Tableaux calculi.
- 05/10, 12/10, 26/10: Introduction to alternating-time temporal logics, concurrent game structures, complexity of ATL^{+}, reasoning with resources, other variants and fragments.
- No lecture on October 19th.

What is in this part of the course?

Introduction to Description Logics and Temporal Logics for Multi-Agent Systems

- Today: Introduction to description logics (DLs).
- 21/09, 28/09: Introduction (Part II), Tableaux calculi.
- 05/10, 12/10, 26/10: Introduction to alternating-time temporal logics, concurrent game structures, complexity of ATL^{+}, reasoning with resources, other variants and fragments.
- No lecture on October 19th.
- Exam during the week including Nov 9th (to be confirmed).
- Final grade: 20% participation+TDs – 80% exam

Current organisation of sessions

- Today: 1h (L) + 10min (B) + 1h (L) + 10min (B) + 1h30 (TDs).
- Next sessions: 30min (corrections TD) + 1h (L) + 10min (B) + 1h (L) + 10min (B) + 1h10 (TD).

What can you expect to learn?

- Basics of description logics including $\mathcal{ALC}$ as well as alternating-time temporal logic ATL and variants.
- Tableaux for $\mathcal{ALC}$, model-checking techniques for ATL-like logics.
- Complexity, decidability, expressive power results for logics dedicated to knowledge representation.

Background

1. Necessary background
   - Basics of first-order logic.
   - Basics of complexity theory.
2. Optional background
   - Basics of modal logics, temporal logics
   - Sequent-style proof systems.
   - Basics of model-checking, verification.

Course material

- Slides, exercises and corrections available on

  `https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-1-39`

  `http://www.lsv.fr/~demri/notes-de-cours.html`

- Do not hesitate to contact me (`demri@lsv.fr`).

Material mainly based on the following documents

- F. Baader, I. Horrocks, C. Lutz and U. Sattler.
  An introduction to Description Logic.
  Cambridge University Press, 2017.
- Ivan Varzinczak's slides (ESSLLI'18)
- Ulrike Sattler & Thomas Schneider's slides (ESSLLI'15).
- Stefan Borgwardt's slides 2018.
- S. Demri, V. Goranko, M. Lange.
  Temporal Logics in Computer Science.
  Cambridge University Press, 2016.

Other (online) ressources

- Description Logic Complexity Navigator by Evgeny Zolin.

  `http://www.cs.man.ac.uk/~ezolin/dl/`

- Proceedings of the Description Logic Workshops

  `http://dl.kr.org/workshops/`

- See also the proceedings of the international conferences:
  - Int. Joint Conference on Artificial Intelligence. (IJCAI)
  - European Conference on Artificial Intelligence. (ECAI)
  - Int. Conference on Principles of Knowledge Representation and Reasoning. (KR)

Plan of the lecture

- Knowledge representation.
- Basic description logic $\mathcal{ALC}$.
- Several extensions of $\mathcal{ALC}$.
- Exercises session.

Knowledge Representation

DLs: where they come from

- First-order logic is not always the most natural language.

$$\forall x \, \exists \, y \, \forall z \, ((\mathtt{P}(x, y) \wedge \mathtt{Q}(y, z) \Rightarrow (\neg \mathtt{Q}(a, y) \vee \mathtt{P}(x, z))))$$

$$\forall \, x \, (\mathtt{Teacher}(x) \Leftrightarrow (\mathtt{Person}(x) \wedge \exists \, y \, (\mathtt{Teaches}(x, y) \wedge \mathtt{Course}(y))))$$

DLs: where they come from

- First-order logic is not always the most natural language.

$$\forall x \, \exists \, y \, \forall z \, ((\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge \mathtt{Q}(\mathtt{y},\mathtt{z}) \Rightarrow (\neg \mathtt{Q}(\mathtt{a},\mathtt{y}) \vee \mathtt{P}(\mathtt{x},\mathtt{z}))))$$

$$\forall \, \mathtt{x} \, (\mathtt{Teacher}(\mathtt{x}) \Leftrightarrow (\mathtt{Person}(\mathtt{x}) \wedge \exists \, \mathtt{y} \, (\mathtt{Teaches}(\mathtt{x},\mathtt{y}) \wedge \mathtt{Course}(\mathtt{y}))))$$

- How to design user-friendly languages for knowledge representation ?

DLs: where they come from

- First-order logic is not always the most natural language.

$$\forall x\, \exists\, y\, \forall z\, ((\mathtt{P}(x, y) \wedge \mathtt{Q}(y, z) \Rightarrow (\neg \mathtt{Q}(a, y) \vee \mathtt{P}(x, z))))$$

$$\forall\, x\, (\mathtt{Teacher}(x) \Leftrightarrow (\mathtt{Person}(x) \wedge \exists\, y\, (\mathtt{Teaches}(x, y) \wedge \mathtt{Course}(y))))$$

- How to design user-friendly languages for knowledge representation ?

- Concept definition from Description Logics.

$$\mathtt{Teacher} \equiv \mathtt{Person} \sqcap \exists \mathtt{Teaches}.\mathtt{Course}$$

Reasoning about ...

- Knowledge — Epistemic Logics
- Rules and obligations — Deontic Logics
- Programs — Hoare Logics
- Time — Temporal Logics

but also separation logics, many-valued logics, non-monotonic logics, etc.

Ontologies

- Formal specification of some domain with concepts, objects, relationships between concepts, objects, etc.
- Backbone of ontologies includes:
  - taxonomy (classification of objects),
  - axioms (to constrain the models of the defined terms).

Ontologies

- Formal specification of some domain with concepts, objects, relationships between concepts, objects, etc.
- Backbone of ontologies includes:
  - taxonomy (classification of objects),
  - axioms (to constrain the models of the defined terms).
- Classification of medical terms: diseases, body parts, drugs, etc.
- Well-known ontologies:
  - Medical ontology SNOMED-CT formalised with description logic $\mathcal{EL}++$.
  - NCI Thesaurus (National Cancer Institute, USA).
  - Gene ontology (world largest source of information on the functions of genes).

Ontologies

- Formal specification of some domain with concepts, objects, relationships between concepts, objects, etc.
- Backbone of ontologies includes:
  - taxonomy (classification of objects),
  - axioms (to constrain the models of the defined terms).
- Classification of medical terms: diseases, body parts, drugs, etc.
- Well-known ontologies:
  - Medical ontology SNOMED-CT formalised with description logic $\mathcal{EL}++$.
  - NCI Thesaurus (National Cancer Institute, USA).
  - Gene ontology (world largest source of information on the functions of genes).
- Free ontology editor Protégé

  `http://protege.stanford.edu/`

The classical student ontology

- Natural language specification:
  - Employed students are students and employees.
  - Students are not taxpayers.
  - Employed students are taxpayers.
  - Employed students who are parents are not taxpayers.
  - To work for is to be employed by.
  - John is an employed student, John works for IBM.
- Classes/relations/individuals.

Main ingredients in formal ontologies

- Model of the world with classes within a domain, relationships between classes and instantiations of classes.

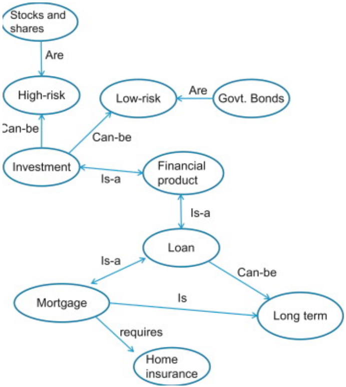
- Formal: abstract model of some domain with (mathematical) semantics and reasoning tasks.

Main ingredients in formal ontologies

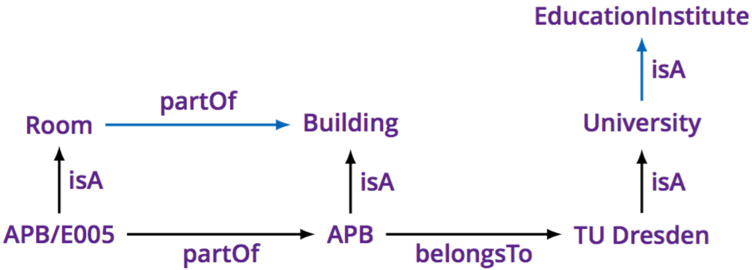
- Model of the world with classes within a domain, relationships between classes and instantiations of classes.
- Formal: abstract model of some domain with (mathematical) semantics and reasoning tasks.
- Classes or concepts: classes of objects with the domain of interest. (Employed student, Parent, Course)
- Relations or roles: relationships between concepts. (being employed by, sibling-of)
- Instances of classes and relations.
  - John is an employed student.
  - Mary works for IBM.

Early KR formalisms

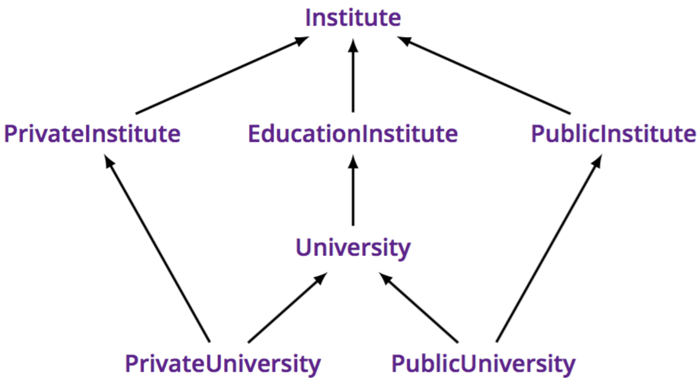
- Graphical formalisms easier to grasp and supposedly close to how knowledge is represented by human beings.
- Large variety of semantical networks.
- Often, lack of formal semantics (see tentatives with the knowledge representation system KL-ONE).



Knowledge graph



Example of concept hierarchy / taxonomy



Challenges with ontologies

- How to define ontologies and to reason on it?

Challenges with ontologies

- How to define ontologies and to reason on it?
- How to relate distinct ontologies?

Challenges with ontologies

- How to define ontologies and to reason on it?
- How to relate distinct ontologies?
- How to repair faulty ontologies ?

Challenges with ontologies

- How to define ontologies and to reason on it?
- How to relate distinct ontologies?
- How to repair faulty ontologies ?
- How to add new concepts or axioms without affecting the old inferences?

Why description logics?

- Formal languages for concepts, relations and instances.
- DLs have all one needs to formalise ontologies.
- Computational properties.
  - Acceptable trade-off between expressivity and complexity.
  - Decidability and often tractability.
  - Implementation in tools of the main reasoning tasks.

Why description logics?

- Formal languages for concepts, relations and instances.
- DLs have all one needs to formalise ontologies.
- Computational properties.
  - Acceptable trade-off between expressivity and complexity.
  - Decidability and often tractability.
  - Implementation in tools of the main reasoning tasks.
- A remarkable suite of languages and tools.
  See e.g.,
  - OWL: Web Ontology Language.
  - Protégé: ontology editor.
  - FaCT++: DL reasoner supporting OWL DL.

Description logics and knowledge representation

- Description logics is a subfield of “Knowledge representation”, itself a subfield of “Artificial intelligence”.

Description logics and knowledge representation

- Description logics is a subfield of “Knowledge representation”, itself a subfield of “Artificial intelligence”.

- Description logic(s):
  - a research field,
  - a family of knowledge representation languages,
  - a member of the family.

Description logics and knowledge representation

- Description logics is a subfield of “Knowledge representation”, itself a subfield of “Artificial intelligence”.
- Description logic(s):
  - a research field,
  - a family of knowledge representation languages,
  - a member of the family.
- Well-defined syntax with formal semantics, decision problems, algorithms, etc.

Basic Description Logic $\mathcal{ALC}$

DLs: the core

- Concept language.

$$\texttt{Person} \sqcap \exists \texttt{Teaches}.\texttt{Course}$$

DLs: the core

- Concept language.

$$\texttt{Person} \sqcap \exists \texttt{Teaches}.\texttt{Course}$$

- Syntactic ingredients of the concept language:
  - **Concept names** for sets of elements, e.g. `Person`.
  - **Role names** interpreted by binary relations between objects, e.g. `EmployedBy`.
  - **Concept constructors** to build **complex concepts**, e.g. $\neg$, $\sqcap$, $\sqcup$, $\exists$.

DLs: the core

- Concept language.

$$\texttt{Person} \sqcap \exists \texttt{Teaches}.\texttt{Course}$$

- Syntactic ingredients of the concept language:
  - **Concept names** for sets of elements, e.g. `Person`.
  - **Role names** interpreted by binary relations between objects, e.g. `EmployedBy`.
  - **Concept constructors** to build **complex concepts**, e.g. $\neg$, $\sqcap$, $\sqcup$, $\exists$.

- Basic terminology stored in a **TBox.**

- Facts about individuals stored in an **ABox.**

(do not worry, you will learn the new terminology)

Basic elements of the language

- **Concept names.**

$$\mathsf{N_C} \stackrel{\text{def}}{=} \{A_1, A_2, B_1, B_2, \ldots\}$$

  Examples: `Parent`, `Sister`, `Student`, `Animal`

Basic elements of the language

- **Concept names.**

$$\mathsf{N}_{\mathbf{C}} \stackrel{\text{def}}{=} \{A_1, A_2, B_1, B_2, \ldots\}$$

Examples: `Parent`, `Sister`, `Student`, `Animal`

- **Role names.**

$$\mathsf{N}_{\mathbf{R}} \stackrel{\text{def}}{=} \{r_1, r_2, s_1, s_2, \ldots\}$$

Examples: `EmployedBy`, `MotherOf`

Basic elements of the language

- **Concept names.**

$$\mathsf{N}_{\mathbf{C}} \stackrel{\text{def}}{=} \{A_1, A_2, B_1, B_2, \ldots\}$$

Examples: `Parent`, `Sister`, `Student`, `Animal`

- **Role names.**

$$\mathsf{N}_{\mathbf{R}} \stackrel{\text{def}}{=} \{r_1, r_2, s_1, s_2, \ldots\}$$

Examples: `EmployedBy`, `MotherOf`

- **Individual names.**

$$\mathsf{N}_{\mathbf{I}} \stackrel{\text{def}}{=} \{a_1, a_2, b_1, b_2, \ldots\}$$

Examples: `Mary`, `Alice`, `John`, `Felix`

Boolean constructors and role restrictions

- Boolean constructors.
  - **Concept negation $\neg$** (class complement)
  - **Concept conjunction $\sqcap$** (class intersection)
  - **Concept disjunction $\sqcup$** (class union)

Boolean constructors and role restrictions

- Boolean constructors.
  - **Concept negation** $\neg$ (class complement)
  - **Concept conjunction** $\sqcap$ (class intersection)
  - **Concept disjunction** $\sqcup$ (class union)
- Role restrictions.
  - **Existential restriction** $\exists$ (at least one related individual)
  - **Value restriction** $\forall$ (all related individuals)
- Many more constructors exist, see forthcoming $\mathcal{ALC}$ extensions.

Boolean constructors and role restrictions

- Boolean constructors.
  - **Concept negation** $\neg$ (class complement)
  - **Concept conjunction** $\sqcap$ (class intersection)
  - **Concept disjunction** $\sqcup$ (class union)
- Role restrictions.
  - **Existential restriction** $\exists$ (at least one related individual)
  - **Value restriction** $\forall$ (all related individuals)
- Many more constructors exist, see forthcoming $\mathcal{ALC}$ extensions.
- Correspondence with modal language:

$$\neg, \sqcap, \sqcup, \exists, \forall \quad \approx \quad \neg, \wedge, \vee, \Diamond, \Box$$

- Correspondence with temporal language:

$$\neg, \sqcap, \sqcup, \exists, \forall \quad \approx \quad \neg, \wedge, \vee, \textbf{EX}, \textbf{AX}$$

Complex concepts in $\mathcal{ALC}$

- $\mathcal{ALC}$: Attributive concept Language with Complements.
- **Complex concepts.**

  $$C ::= \top \mid \bot \mid A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.C \mid \forall r.C,$$

  where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$.
- Examples of complex concepts:
  - $\mathtt{Student} \sqcap \neg\exists\mathtt{Pays.Tax}$
  - $\exists\mathtt{MotherOf}.(\exists\mathtt{MotherOf}.A)$

Complex concepts in $\mathcal{ALC}$

- $\mathcal{ALC}$: Attributive concept Language with Complements.
- **Complex concepts.**

  $$C ::= \top \mid \bot \mid A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.C \mid \forall r.C,$$

  where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$.
- Examples of complex concepts:
  - $\texttt{Student} \sqcap \neg \exists \texttt{Pays.Tax}$
  - $\exists \texttt{MotherOf}.(\exists \texttt{MotherOf}.A)$
- Syntax errors in

  $$\texttt{Student} \sqcup \forall \neg \texttt{Tax} \quad \forall \exists \texttt{MotherOf.Mary}$$

- $C \Rightarrow D \stackrel{\text{def}}{=} \neg C \sqcup D$. (as in pro

Interpretation

Concept name/role/individual

$\approx$

unary predicate/binary predicate/constant

Interpretation

Concept name/role/individual

$\approx$

unary predicate/binary predicate/constant

- **Interpretation** $\mathcal{I} \stackrel{\text{def}}{=} (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

  (usual notation from the literature)

- $\Delta^{\mathcal{I}}$: non-empty set (the **domain**).

- $\cdot^{\mathcal{I}}$: **interpretation function** such that

$$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \quad r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \quad a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$$

Interpretation

Concept name/role/individual

$\approx$

unary predicate/binary predicate/constant

- **Interpretation** $\mathcal{I} \stackrel{\text{def}}{=} (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

  (usual notation from the literature)

- $\Delta^{\mathcal{I}}$: non-empty set (the **domain**).

- $\cdot^{\mathcal{I}}$: **interpretation function** such that

$$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \quad r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \quad a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$$

- A priori, $\Delta^{\mathcal{I}}$ is arbitrary (not necessarily finite).

- $\mathcal{I}$ can be viewed as a first-order model for unary and binary predicate symbols and constants.

Semantics for complex concepts

$$
\begin{aligned}
\top^{\mathcal{I}} &\stackrel{\text{def}}{=} \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &\stackrel{\text{def}}{=} \emptyset \\
(\neg C)^{\mathcal{I}} &\stackrel{\text{def}}{=} \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C_1 \sqcup C_2)^{\mathcal{I}} &\stackrel{\text{def}}{=} C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}} \\
(C_1 \sqcap C_2)^{\mathcal{I}} &\stackrel{\text{def}}{=} C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\
(\exists r.C)^{\mathcal{I}} &\stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(\mathfrak{a}) \cap C^{\mathcal{I}} \neq \emptyset\} \\
(\forall r.C)^{\mathcal{I}} &\stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(\mathfrak{a}) \subseteq C^{\mathcal{I}}\}
\end{aligned}
$$

$$\mathcal{R}(\mathfrak{a}) \stackrel{\text{def}}{=} \{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in \mathcal{R}\}$$

Semantics for complex concepts

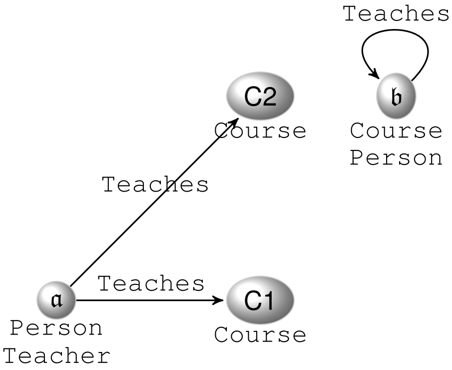
$$
\begin{array}{lcl}
\top^{\mathcal{I}} & \stackrel{\text{def}}{=} & \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} & \stackrel{\text{def}}{=} & \emptyset \\
(\neg C)^{\mathcal{I}} & \stackrel{\text{def}}{=} & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C_1 \sqcup C_2)^{\mathcal{I}} & \stackrel{\text{def}}{=} & C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}} \\
(C_1 \sqcap C_2)^{\mathcal{I}} & \stackrel{\text{def}}{=} & C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\
(\exists r.C)^{\mathcal{I}} & \stackrel{\text{def}}{=} & \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(\mathfrak{a}) \cap C^{\mathcal{I}} \neq \emptyset\} \\
(\forall r.C)^{\mathcal{I}} & \stackrel{\text{def}}{=} & \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(\mathfrak{a}) \subseteq C^{\mathcal{I}}\}
\end{array}
$$

$$\mathcal{R}(\mathfrak{a}) \stackrel{\text{def}}{=} \{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in \mathcal{R}\}$$

- In modal logic lingua, $\mathfrak{a} \in C^{\mathcal{I}}$ corresponds to $\mathcal{I}, \mathfrak{a} \models C$.

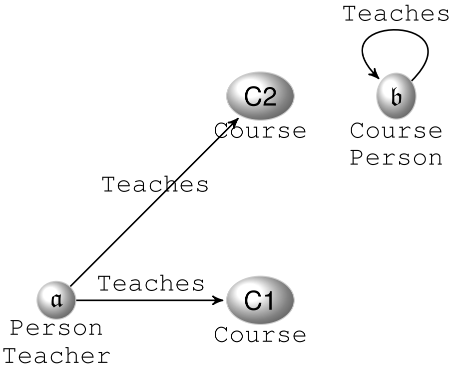
(standard duality applie $(\exists r.C)^{\mathcal{I}} = (\neg\forall r.\neg C)^{\mathcal{I}}$)

Graphical representation



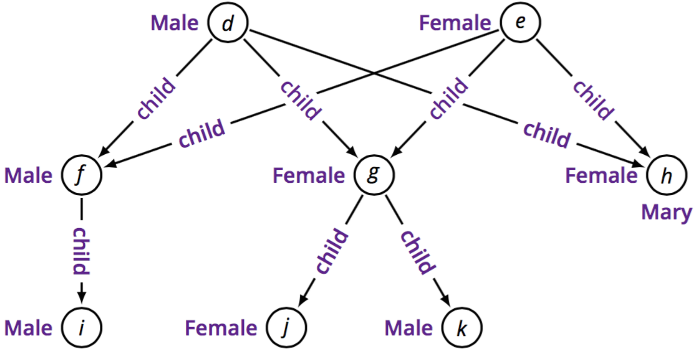
- $\Delta^{\mathcal{I}} = \{\mathfrak{a}, \mathfrak{b}, C1, C2\}$.
- $\texttt{Teaches}^{\mathcal{I}} = \{(\mathfrak{a}, C1), (\mathfrak{a}, C2), (\mathfrak{b}, \mathfrak{b})\}$.
- $\texttt{Person}^{\mathcal{I}} = \{\mathfrak{a}, \mathfrak{b}\}$, $\texttt{Course}^{\mathcal{I}} = \{C1, C2, \mathfrak{b}\}$.

Graphical representation



- $\Delta^{\mathcal{I}} = \{\mathfrak{a}, \mathfrak{b}, C1, C2\}$.
- $\texttt{Teaches}^{\mathcal{I}} = \{(\mathfrak{a}, C1), (\mathfrak{a}, C2), (\mathfrak{b}, \mathfrak{b})\}$.
- $\texttt{Person}^{\mathcal{I}} = \{\mathfrak{a}, \mathfrak{b}\}$, $\texttt{Course}^{\mathcal{I}} = \{C1, C2, \mathfrak{b}\}$.
- $\mathfrak{a} \in (\forall \texttt{Teaches}.\texttt{Course})^{\mathcal{I}}$. (why?)

Another example



- $(\exists \texttt{child}.\top)^{\mathcal{I}} = \{d, e, f, g\}$.
- $(\texttt{Female} \sqcap \exists \texttt{child}.\top)^{\mathcal{I}} = \{e, g\}$.
- $(\exists \texttt{child}.\texttt{Mary})^{\mathcal{I}} = \{d, e\}$.
  ⚠ Herein, `Mary` understood as a concept name (on this slide).

Concept satisfiability problem

- Concept satisfiability problem:

  Input: A (complex) concept C in $\mathcal{ALC}$.

  Question: Is there an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that $C^{\mathcal{I}} \neq \emptyset$?

- This corresponds to the standard formulation for the satisfiability problem.

  (in modal logics, temporal logics, etc.).

Concept satisfiability problem

- Concept satisfiability problem:

  Input: A (complex) concept C in $\mathcal{ALC}$.

  Question: Is there an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that $C^{\mathcal{I}} \neq \emptyset$?

- This corresponds to the standard formulation for the satisfiability problem.

  (in modal logics, temporal logics, etc.).

- The concept satisfiability problem for $\mathcal{ALC}$ is PSPACE-complete.

- $\mathcal{ALC}$ has the finite interpretation property: every satisfiable concept has an interpretation with a finite domain.

Statements

- Concept inclusion.
  Teachers are persons. Employed students are employees.

- Concept and role membership.
  Mary is a student. Alice is a teacher.
  Laura teaches the course “Automata Theory”.

Statements

- Concept inclusion.
  Teachers are persons. Employed students are employees.
- Concept and role membership.
  Mary is a student. Alice is a teacher.
  Laura teaches the course “Automata Theory”.
- Statements are not concepts and express properties of concepts, roles and individuals.

General concept inclusion (GCI)

- Expressions of the form

$$C \sqsubseteq D$$

  are called **general concept inclusions.**

- Intuitive meaning:
  - D subsumes C.
  - C is more specific than D.

General concept inclusion (GCI)

- Expressions of the form

$$C \sqsubseteq D$$

  are called **general concept inclusions.**

- Intuitive meaning:
  - D subsumes C.
  - C is more specific than D.

- Example: $\texttt{Employee} \sqsubseteq \exists \texttt{WorksFor}.\top$

General concept inclusion (GCI)

- Expressions of the form

$$C \sqsubseteq D$$

  are called **general concept inclusions.**

- Intuitive meaning:
  - D subsumes C.
  - C is more specific than D.

- Example: $\texttt{Employee} \sqsubseteq \exists \texttt{WorksFor}.\top$

- Satisfaction relation: $\mathcal{I} \models C \sqsubseteq D \overset{\text{def}}{\Leftrightarrow} C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

- $C \sqsubseteq D$ understood as a global statement about $\mathcal{I}$.

Concept equivalence

- $C \sqsubseteq D$ and $D \sqsubseteq C$ abbreviated by

$$C \equiv D$$

  called **concept equivalence.**

- Satisfaction relation: $\mathcal{I} \models C \equiv D \overset{\text{def}}{\Leftrightarrow} C^{\mathcal{I}} = D^{\mathcal{I}}$.

- $\top \equiv (\neg \mathtt{Student} \sqcup \mathtt{Student})$.

Concept equivalence

- $C \sqsubseteq D$ and $D \sqsubseteq C$ abbreviated by

  $$C \equiv D$$

  called **concept equivalence.**

- Satisfaction relation: $\mathcal{I} \models C \equiv D \overset{\text{def}}{\Leftrightarrow} C^{\mathcal{I}} = D^{\mathcal{I}}$.

- $\top \equiv (\neg \texttt{Student} \sqcup \texttt{Student})$.

- **Concept definition** ($A \in \mathsf{N}_{\mathbf{C}}$ is a concept name)

  $$A \equiv C$$

Subsumption problem

- Subsumption problem:

  Input: A GCI $C \sqsubseteq D$ with $C, D \in \mathcal{ALC}$.

  Question: Is it the case that for all interpretations $\mathcal{I}$, we have $\mathcal{I} \models C \sqsubseteq D$?

Subsumption problem

- Subsumption problem:

  Input: A GCI $C \sqsubseteq D$ with $C, D \in \mathcal{ALC}$.

  Question: Is it the case that for all interpretations $\mathcal{I}$, we have $\mathcal{I} \models C \sqsubseteq D$?

- $C \sqsubseteq D$ is “not valid” iff $C \sqcap \neg D$ is satisfiable.

Subsumption problem

- Subsumption problem:

  Input: A GCI $C \sqsubseteq D$ with $C, D \in \mathcal{ALC}$.

  Question: Is it the case that for all interpretations $\mathcal{I}$, we have $\mathcal{I} \models C \sqsubseteq D$?

- $C \sqsubseteq D$ is “not valid” iff $C \sqcap \neg D$ is satisfiable.

- As coPSPACE =PSPACE, the subsumption problem for $\mathcal{ALC}$ is PSPACE-complete too.

Assertions

- **Concept assertion**: stating that an individual is an instance of a concept.

$$a : C$$

Assertions

- **Concept assertion**: stating that an individual is an instance of a concept.

$$a : C$$

- Satisfaction relation: $\mathcal{I} \models a : C \overset{\text{def}}{\Leftrightarrow} a^{\mathcal{I}} \in C^{\mathcal{I}}$.
  ⚠ In $\mathcal{ALC}$, a does not occur in concepts !

Assertions

- **Concept assertion**: stating that an individual is an instance of a concept.

$$a : C$$

- Satisfaction relation: $\mathcal{I} \models a : C \overset{\text{def}}{\Leftrightarrow} a^{\mathcal{I}} \in C^{\mathcal{I}}$.
  ⚠ In $\mathcal{ALC}$, a does not occur in concepts !

- **Role assertion**: two individuals are in a relation.

$$(a, b) : r$$

- Satisfaction relation: $\mathcal{I} \models (a, b) : r \overset{\text{def}}{\Leftrightarrow} (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$.

- Examples:
  - $\texttt{Alice} : \texttt{Student} \sqcap \neg\exists \texttt{Pays}.\texttt{Tax}$.
  - $(\texttt{Laura}, \texttt{CNRS}) : \texttt{WorksFor}$.

The validity problem

- Validity problem:

  Input: A statement α in $\mathcal{ALC}$.

  Question: Is the case that for all interpretations $\mathcal{I}$, we have $\mathcal{I} \models \alpha$?

- Validity of α is written $\models \alpha$.

The validity problem

- Validity problem:

  Input: A statement α in $\mathcal{ALC}$.

  Question: Is the case that for all interpretations $\mathcal{I}$, we have $\mathcal{I} \models \alpha$?

- Validity of α is written $\models \alpha$.

- Validity of $\top \sqsubseteq C$ corresponds to the usual notion of validity for C, i.e. for all interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, we have $C^{\mathcal{I}} = \Delta^{\mathcal{I}}$.

The validity problem

- Validity problem:

  Input: A statement α in $\mathcal{ALC}$.

  Question: Is the case that for all interpretations $\mathcal{I}$, we have $\mathcal{I} \models \alpha$?

- Validity of α is written $\models \alpha$.

- Validity of $\top \sqsubseteq C$ corresponds to the usual notion of validity for C, i.e. for all interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, we have $C^{\mathcal{I}} = \Delta^{\mathcal{I}}$.

- Examples of valid statements:
  - $\models \forall r.(C \sqcap D) \sqsubseteq \forall r.C$.
  - $\models a : C \sqcup \neg C$.
  - $\models \top \sqsubseteq (\neg(C \sqcap D) \sqcup (C \sqcup D))$.

- The validity problem for $\mathcal{ALC}$ is PSPACE-complete.

What is a knowledge base (a.k.a. ontology) ?

- **Terminological Box** (**TBox**) $\mathcal{T}$: finite collection of GCIs.
  - I.e., a finite set of concept inclusions.
  - This provides definitions of concepts (a terminology).

What is a knowledge base (a.k.a. ontology) ?

- **Terminological Box** (**TBox**) $\mathcal{T}$: finite collection of GCIs.
  - I.e., a finite set of concept inclusions.
  - This provides definitions of concepts (a terminology).
- **Assertional** Box (**ABox**) $\mathcal{A}$: finite collection of assertions.
  - I.e., a finite set of concept and role assertions.
  - This provides a partial view on the interpretations and can be understood as a incomplete finite database.

What is a knowledge base (a.k.a. ontology) ?

- **Terminological Box** (**TBox**) $\mathcal{T}$: finite collection of GCIs.
  - I.e., a finite set of concept inclusions.
  - This provides definitions of concepts (a terminology).

- **Assertional** Box (**ABox**) $\mathcal{A}$: finite collection of assertions.
  - I.e., a finite set of concept and role assertions.
  - This provides a partial view on the interpretations and can be understood as a incomplete finite database.

- **Knowledge base** $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{A})$.

- Knowledge bases are also called **ontologies**.

A knowledge base $\mathcal{K}_\star$

- TBox $\mathcal{T}$:

$$
\begin{array}{lcl}
\texttt{Course} & \sqsubseteq & \neg\texttt{Person} \\
\texttt{Teacher} & \sqsubseteq & \texttt{Person} \sqcap \exists\texttt{Teaches}.\texttt{Course} \\
\exists\texttt{Teaches}.\top & \sqsubseteq & \texttt{Person} \\
\texttt{Student} & \sqsubseteq & \texttt{Person} \sqcap \exists\texttt{Attends}.\texttt{Course} \\
\exists\texttt{Attends}.\top & \sqsubseteq & \texttt{Person}
\end{array}
$$

A knowledge base $\mathcal{K}_\star$

- TBox $\mathcal{T}$:

$$\begin{array}{lcl}
\texttt{Course} & \sqsubseteq & \neg\texttt{Person} \\
\texttt{Teacher} & \sqsubseteq & \texttt{Person} \sqcap \exists\texttt{Teaches}.\texttt{Course} \\
\exists\texttt{Teaches}.\top & \sqsubseteq & \texttt{Person} \\
\texttt{Student} & \sqsubseteq & \texttt{Person} \sqcap \exists\texttt{Attends}.\texttt{Course} \\
\exists\texttt{Attends}.\top & \sqsubseteq & \texttt{Person}
\end{array}$$

- ABox $\mathcal{A}$:

$$\begin{array}{rl}
\texttt{Mary}: & \texttt{Person} \\
\texttt{CS600}: & \texttt{Course} \\
\texttt{Alice}: & \texttt{Person} \sqcap \texttt{Teacher} \\
(\texttt{Alice}, \texttt{CS600}): & \texttt{Teaches} \\
(\texttt{Mary}, \texttt{CS600}): & \texttt{Attends}
\end{array}$$

Consequences from knowledge bases

- Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.
  - $\mathcal{I} \models \mathcal{A} \overset{\text{def}}{\Leftrightarrow}$ for all $\alpha \in \mathcal{A}$, we have $\mathcal{I} \models \alpha$.
  - $\mathcal{I} \models \mathcal{T} \overset{\text{def}}{\Leftrightarrow}$ for all $\alpha \in \mathcal{T}$, we have $\mathcal{I} \models \alpha$.
  - $\mathcal{I} \models \mathcal{K} \overset{\text{def}}{\Leftrightarrow} \mathcal{I} \models \mathcal{A}$ and $\mathcal{I} \models \mathcal{T}$.

- $\mathcal{K} \models \alpha \overset{\text{def}}{\Leftrightarrow}$ for all interpretations $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$, we have $\mathcal{I} \models \alpha$.

Consequences from knowledge bases

- Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.
  - $\mathcal{I} \models \mathcal{A} \overset{\text{def}}{\Leftrightarrow}$ for all $\alpha \in \mathcal{A}$, we have $\mathcal{I} \models \alpha$.
  - $\mathcal{I} \models \mathcal{T} \overset{\text{def}}{\Leftrightarrow}$ for all $\alpha \in \mathcal{T}$, we have $\mathcal{I} \models \alpha$.
  - $\mathcal{I} \models \mathcal{K} \overset{\text{def}}{\Leftrightarrow} \mathcal{I} \models \mathcal{A}$ and $\mathcal{I} \models \mathcal{T}$.

- $\mathcal{K} \models \alpha \overset{\text{def}}{\Leftrightarrow}$ for all interpretations $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$, we have $\mathcal{I} \models \alpha$.

- $\mathcal{K}_\star \models \texttt{CS600} : \neg\texttt{Person}$ and $\mathcal{K}_\star \models \texttt{Alice} : \texttt{Teacher}$.

Decision problems relatively to a knowlegde base

- $\mathcal{K}$ is **consistent** $\overset{\text{def}}{\Leftrightarrow}$ there is some $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$.

- C is **satisfiable with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ there is $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $C^{\mathcal{I}} \neq \emptyset$.

Decision problems relatively to a knowlegde base

- $\mathcal{K}$ is **consistent** $\overset{\text{def}}{\Leftrightarrow}$ there is some $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$.

- C is **satisfiable with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ there is $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $C^{\mathcal{I}} \neq \emptyset$.

- C **is subsumed by** D **with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ $\mathcal{K} \models C \sqsubseteq D$.

Decision problems relatively to a knowlegde base

- $\mathcal{K}$ is **consistent** $\overset{\text{def}}{\Leftrightarrow}$ there is some $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$.

- C is **satisfiable with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ there is $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $C^{\mathcal{I}} \neq \emptyset$.

- C **is subsumed by** D **with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ $\mathcal{K} \models C \sqsubseteq D$.

- C and D **are equivalent with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ $\mathcal{T} \models C \equiv D$.

- a is an **instance** of C **with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ $\mathcal{K} \models a : C$.

Decision problems relatively to a knowlegde base

- $\mathcal{K}$ is **consistent** $\overset{\text{def}}{\Leftrightarrow}$ there is some $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$.

- C is **satisfiable with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ there is $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $C^{\mathcal{I}} \neq \emptyset$.

- C **is subsumed by** D **with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ $\mathcal{K} \models C \sqsubseteq D$.

- C and D **are equivalent with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ $\mathcal{T} \models C \equiv D$.

- a is an **instance** of C **with respect to** $\mathcal{K}$ $\overset{\text{def}}{\Leftrightarrow}$ $\mathcal{K} \models a : C$.

- $\mathcal{K} \models C \sqsubseteq D$ also written $C \sqsubseteq_{\mathcal{K}} D$.
  $\mathcal{K} \models C \equiv D$ also written $C \equiv_{\mathcal{K}} D$.

Subsumption problem w.r.t a TBox

- $\mathcal{T} \models C \sqsubseteq D \overset{\text{def}}{\Leftrightarrow}$ for all interpretations $\mathcal{I}$, $\mathcal{I} \models \mathcal{T}$ implies $\mathcal{I} \models C \sqsubseteq D$.
- $\mathcal{T} \models C \sqsubseteq D$ also written $C \sqsubseteq_{\mathcal{T}} D$.
- **Subsumption problem w.r.t. a TBox:**

  Input: TBox $\mathcal{T}$, concepts C, D

  Question: Does $\mathcal{T} \models C \sqsubseteq D$?

Relationships between reasoning problems

- C and D are equivalent w.r.t. $\mathcal{K}$ iff C is subsumed by D w.r.t. $\mathcal{K}$ and D is subsumed by C w.r.t. $\mathcal{K}$.

Relationships between reasoning problems

- C and D are equivalent w.r.t. $\mathcal{K}$ iff C is subsumed by D w.r.t. $\mathcal{K}$ and D is subsumed by C w.r.t. $\mathcal{K}$.
- $C \sqsubseteq_{\mathcal{K}} D$ iff $C \sqcap \neg D$ is not satisfiable w.r.t. $\mathcal{K}$.

Relationships between reasoning problems

- C and D are equivalent w.r.t. $\mathcal{K}$ iff C is subsumed by D w.r.t. $\mathcal{K}$ and D is subsumed by C w.r.t. $\mathcal{K}$.
- $C \sqsubseteq_{\mathcal{K}} D$ iff $C \sqcap \neg D$ is not satisfiable w.r.t. $\mathcal{K}$.
- C is satisfiable w.r.t. $\mathcal{K}$ iff $C \not\sqsubseteq_{\mathcal{K}} \bot$.

Relationships between reasoning problems

- C and D are equivalent w.r.t. $\mathcal{K}$ iff C is subsumed by D w.r.t. $\mathcal{K}$ and D is subsumed by C w.r.t. $\mathcal{K}$.

- $C \sqsubseteq_{\mathcal{K}} D$ iff $C \sqcap \neg D$ is not satisfiable w.r.t. $\mathcal{K}$.

- C is satisfiable w.r.t. $\mathcal{K}$ iff $C \not\sqsubseteq_{\mathcal{K}} \bot$.

- C is satisfiable w.r.t. $\mathcal{K}$ iff $(\mathcal{T}, \mathcal{A} \cup \{b : C\})$ is consistent. (b is fresh)

Relationships between reasoning problems

- C and D are equivalent w.r.t. $\mathcal{K}$ iff C is subsumed by D w.r.t. $\mathcal{K}$ and D is subsumed by C w.r.t. $\mathcal{K}$.
- $C \sqsubseteq_{\mathcal{K}} D$ iff $C \sqcap \neg D$ is not satisfiable w.r.t. $\mathcal{K}$.
- C is satisfiable w.r.t. $\mathcal{K}$ iff $C \not\sqsubseteq_{\mathcal{K}} \bot$.
- C is satisfiable w.r.t. $\mathcal{K}$ iff $(\mathcal{T}, \mathcal{A} \cup \{b : C\})$ is consistent. (b is fresh)
- $\mathcal{K} \models a : C$ iff $(\mathcal{T}, \mathcal{A} \cup \{a : \neg C\})$ is not consistent.

(this works fine because $\neg$ admitted in $\mathcal{ALC}$)

C is satisfiable w.r.t. $\mathcal{K}$ iff $(\mathcal{T}, \mathcal{A} \cup \{b : C\})$ is consistent

- Suppose that C is satisfiable w.r.t. $\mathcal{K}$.
  - There is $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $C^{\mathcal{I}} \neq \emptyset$, say $\mathfrak{a} \in C^{\mathcal{I}}$.
  - Let $\mathcal{I}'$ be the variant of $\mathcal{I}$ such that $b^{\mathcal{I}'} \stackrel{\text{def}}{=} \mathfrak{a}$.

C is satisfiable w.r.t. $\mathcal{K}$ iff $(\mathcal{T}, \mathcal{A} \cup \{b : C\})$ is consistent

- Suppose that C is satisfiable w.r.t. $\mathcal{K}$.
  - There is $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $C^{\mathcal{I}} \neq \emptyset$, say $\mathfrak{a} \in C^{\mathcal{I}}$.
  - Let $\mathcal{I}'$ be the variant of $\mathcal{I}$ such that $b^{\mathcal{I}'} \stackrel{\text{def}}{=} \mathfrak{a}$.
  - As b does not appear in $\mathcal{K}$ and C, we have $\mathcal{I}' \models \mathcal{K}$.
  - Furthermore, $\mathcal{I}' \models b : C$ as $C^{\mathcal{I}} = C^{\mathcal{I}'}$.

C is satisfiable w.r.t. $\mathcal{K}$ iff $(\mathcal{T}, \mathcal{A} \cup \{b : C\})$ is consistent

- Suppose that C is satisfiable w.r.t. $\mathcal{K}$.
  - There is $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $C^{\mathcal{I}} \neq \emptyset$, say $\mathfrak{a} \in C^{\mathcal{I}}$.
  - Let $\mathcal{I}'$ be the variant of $\mathcal{I}$ such that $b^{\mathcal{I}'} \stackrel{\text{def}}{=} \mathfrak{a}$.
  - As b does not appear in $\mathcal{K}$ and C, we have $\mathcal{I}' \models \mathcal{K}$.
  - Furthermore, $\mathcal{I}' \models b : C$ as $C^{\mathcal{I}} = C^{\mathcal{I}'}$.
  - Consequently, $\mathcal{I}' \models (\mathcal{T}, \mathcal{A} \cup \{b : C\})$.

C is satisfiable w.r.t. $\mathcal{K}$ iff $(\mathcal{T}, \mathcal{A} \cup \{b : C\})$ is consistent

- Suppose that C is satisfiable w.r.t. $\mathcal{K}$.
  - There is $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $C^{\mathcal{I}} \neq \emptyset$, say $\mathfrak{a} \in C^{\mathcal{I}}$.
  - Let $\mathcal{I}'$ be the variant of $\mathcal{I}$ such that $b^{\mathcal{I}'} \stackrel{\text{def}}{=} \mathfrak{a}$.
  - As b does not appear in $\mathcal{K}$ and C, we have $\mathcal{I}' \models \mathcal{K}$.
  - Furthermore, $\mathcal{I}' \models b : C$ as $C^{\mathcal{I}} = C^{\mathcal{I}'}$.
  - Consequently, $\mathcal{I}' \models (\mathcal{T}, \mathcal{A} \cup \{b : C\})$.
- Now, suppose that $(\mathcal{T}, \mathcal{A} \cup \{b : C\})$ is consistent.
  - There is $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}$, $\mathcal{I} \models \mathcal{A}$ and $\mathcal{I} \models b : C$.
  - Consequently, $b^{\mathcal{I}} \in C^{\mathcal{I}}$.
  - So, there is some $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and $C^{\mathcal{I}}$ is non-empty.

Classification

- Deduce implicit knowledge from the explicitly represented knowledge.
- For all A, B in $\mathcal{K}$, check whether $A \sqsubseteq_{\mathcal{K}} B$.
- For all A in $\mathcal{K}$, check whether A is satisfiable w.r.t. $\mathcal{K}$.
  If not for some B, a modelling error is probable.
- For all a and C in $\mathcal{K}$, check whether $\mathcal{K} \models a : C$.

Classification

- Deduce implicit knowledge from the explicitly represented knowledge.

- For all A, B in $\mathcal{K}$, check whether $A \sqsubseteq_{\mathcal{K}} B$.

- For all A in $\mathcal{K}$, check whether A is satisfiable w.r.t. $\mathcal{K}$.
  If not for some B, a modelling error is probable.

- For all a and C in $\mathcal{K}$, check whether $\mathcal{K} \models a : C$.

- Classifying a knowledge base $\mathcal{K}$.
  1. Check whether $\mathcal{K}$ is consistent, if yes, go 2.
  2. For each pair A, B of concept names (plus $\top$, $\bot$), check whether $\mathcal{K} \models A \sqsubseteq B$.
  3. For all individual names a and concepts C in $\mathcal{K}$, check whether $\mathcal{K} \models a : C$.

  leading to $\mathcal{K}$'s **inferred class hierarchy**.

Complexity results for $\mathcal{ALC}$

- Concept satisfiability and subsumption problems are PSPACE-complete. (no knowledge base involved)

Complexity results for $\mathcal{ALC}$

- Concept satisfiability and subsumption problems are PSPACE-complete. (no knowledge base involved)

- Knowlegde base consistency problem is EXPTIME-complete.

$$\text{NP} \subseteq \text{PSPACE} \subseteq \text{EXPTIME} \subset \text{2EXPTIME} \subseteq \text{N2EXPTIME}$$

- Recall that $C \sqsubseteq_{\mathcal{K}} D$ iff $(\mathcal{T}, \mathcal{A} \cup \{b : C \sqcap \neg D\})$ is not consistent.

Digression: closed world / open world assumptions

- Standard semantics for $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ makes an **Open World Assumption** (OWA).
  - No assumption that all information is known about all individuals in a domain.
  - Elements in the interpretation domain may not correspond to interpretations of individual names.
- **Closed World Assumption** (CWA) enforces that the only elements in the domain are named elements (by individual names).
- Standard databases make the CWA: facts that are not explicitly stated are false.

Several Extensions of $\mathcal{ALC}$ (Part I)

Extensions: a feature of DLs

- Concepts/assertions in $\mathcal{ALC}$ have a limited expressive power.
    - How to express simple arithmetical constraints such as "Alice teaches at least three courses"?
    - How to enforce constraints between roles?
      For instance, $r^{\mathcal{I}} = (s^{\mathcal{I}})^{-1}$ or $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$.

Extensions: a feature of DLs

- Concepts/assertions in $\mathcal{ALC}$ have a limited expressive power.
  - How to express simple arithmetical constraints such as "Alice teaches at least three courses"?
  - How to enforce constraints between roles?
    For instance, $r^{\mathcal{I}} = (s^{\mathcal{I}})^{-1}$ or $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$.

- The expressive power of $\mathcal{ALC}$ concepts can be characterised precisely, thanks to the notion of bisimulation (not presented today).

⚠ Trade-off between the expressive power and the computational properties of the extensions.

Extensions: a feature of DLs

- Concepts/assertions in $\mathcal{ALC}$ have a limited expressive power.
  - How to express simple arithmetical constraints such as "Alice teaches at least three courses"?
  - How to enforce constraints between roles?
    For instance, $r^{\mathcal{I}} = (s^{\mathcal{I}})^{-1}$ or $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$.

- The expressive power of $\mathcal{ALC}$ concepts can be characterised precisely, thanks to the notion of bisimulation (not presented today).

⚠ Trade-off between the expressive power and the computational properties of the extensions.

- In the other direction: study of $\mathcal{ALC}$ fragments to reduce the complexity while preserving the expression of interesting properties, see e.g. $\mathcal{EL}$, $\mathcal{FL}_0$ or DL-Lite.

Inverse roles

$$\begin{array}{lcl}
\texttt{Course} & \sqsubseteq & \neg\texttt{Person} \\
\texttt{Teacher} & \sqsubseteq & \texttt{Person} \sqcap \exists\texttt{Teaches.Course} \\
\exists\texttt{Teaches}.\top & \sqsubseteq & \texttt{Person} \\
\texttt{Student} & \sqsubseteq & \texttt{Person} \sqcap \exists\texttt{Attends.Course} \\
\exists\texttt{Attends}.\top & \sqsubseteq & \texttt{Person}
\end{array}$$

Inverse roles

$$\begin{array}{lcl}
\texttt{Course} & \sqsubseteq & \neg\texttt{Person} \\
\texttt{Teacher} & \sqsubseteq & \texttt{Person} \sqcap \exists\texttt{Teaches.Course} \\
\exists\texttt{Teaches.}\top & \sqsubseteq & \texttt{Person} \\
\texttt{Student} & \sqsubseteq & \texttt{Person} \sqcap \exists\texttt{Attends.Course} \\
\exists\texttt{Attends.}\top & \sqsubseteq & \texttt{Person} \\
\texttt{Professor} & \sqsubseteq & \texttt{Teacher} \\
\texttt{Course} & \sqsubseteq & \forall\texttt{TaughtBy.}\neg\texttt{Professor}
\end{array}$$

Inverse roles

$$\begin{array}{lcl}
\mathtt{Course} & \sqsubseteq & \neg\mathtt{Person} \\
\mathtt{Teacher} & \sqsubseteq & \mathtt{Person} \sqcap \exists\mathtt{Teaches}.\mathtt{Course} \\
\exists\mathtt{Teaches}.\top & \sqsubseteq & \mathtt{Person} \\
\mathtt{Student} & \sqsubseteq & \mathtt{Person} \sqcap \exists\mathtt{Attends}.\mathtt{Course} \\
\exists\mathtt{Attends}.\top & \sqsubseteq & \mathtt{Person} \\
\mathtt{Professor} & \sqsubseteq & \mathtt{Teacher} \\
\mathtt{Course} & \sqsubseteq & \forall\mathtt{TaughtBy}.\neg\mathtt{Professor}
\end{array}$$

- Extending $\mathsf{N}_{\mathbf{R}}$ with **inverse roles**:

$$\mathsf{N}_{\mathbf{R}} \cup \{r^- \mid r \in \mathsf{N}_{\mathbf{R}}\}$$

- Given $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, $(r^-)^{\mathcal{I}} \stackrel{\text{def}}{=} (r^{\mathcal{I}})^{-1}$ where

$$\mathcal{R}^{-1} \stackrel{\text{def}}{=} \{(\mathfrak{b}, \mathfrak{a}) \mid (\mathfrak{a}, \mathfrak{b}) \in \mathcal{R}\}$$

Elimination of the role name $\texttt{TaughtBy}$

- Back to the previous example.

$$\begin{array}{lcl} \texttt{Professor} & \sqsubseteq & \texttt{Teacher} \\ \texttt{Course} & \sqsubseteq & \forall \texttt{Teaches}^{-}.\neg \texttt{Professor} \end{array}$$

Elimination of the role name $\texttt{TaughtBy}$

- Back to the previous example.

$$\begin{array}{lll} \texttt{Professor} & \sqsubseteq & \texttt{Teacher} \\ \texttt{Course} & \sqsubseteq & \forall \texttt{Teaches}^{-}.\neg \texttt{Professor} \end{array}$$

- Given a logic $\mathfrak{L}$, $\mathfrak{L}\mathcal{I}$ is defined as $\mathfrak{L}$ except that inverse roles are added. ⚠ Symbol '$\mathcal{I}$' overloaded here

Elimination of the role name `TaughtBy`

- Back to the previous example.

$$\begin{array}{lll} \texttt{Professor} & \sqsubseteq & \texttt{Teacher} \\ \texttt{Course} & \sqsubseteq & \forall \texttt{Teaches}^{-}.\neg\texttt{Professor} \end{array}$$

- Given a logic $\mathfrak{L}$, $\mathfrak{L}\mathcal{I}$ is defined as $\mathfrak{L}$ except that inverse roles are added. ⚠ Symbol '$\mathcal{I}$' overloaded here

(Name
accumulation of symbol $\mathfrak{L}$.)

Elimination of the role name TaughtBy

- Back to the previous example.

$$\begin{array}{lcl} \texttt{Professor} & \sqsubseteq & \texttt{Teacher} \\ \texttt{Course} & \sqsubseteq & \forall \texttt{Teaches}^{-}.\neg\texttt{Professor} \end{array}$$

- Given a logic $\mathfrak{L}$, $\mathfrak{L}\mathcal{I}$ is defined as $\mathfrak{L}$ except that inverse roles are added. ⚠ Symbol '$\mathcal{I}$' overloaded here

(Name

accumulation of symbol $\mathfrak{L}$.)

- Concept satisfiability for $\mathcal{ALCI}$ remains PSPACE-complete and knowledge consistency remains EXPTIME-complete.

Number restrictions

- How to express in $\mathcal{ALC}$ that a student attends to at least three courses?

Number restrictions

- How to express in $\mathcal{ALC}$ that a student attends to at least three courses?

$$
\begin{array}{lll}
\texttt{Student} & \sqsubseteq & \exists\texttt{Attends}.(\texttt{Course} \sqcap A) \\
\texttt{Student} & \sqsubseteq & \exists\texttt{Attends}.(\texttt{Course} \sqcap \neg A \sqcap B) \\
\texttt{Student} & \sqsubseteq & \exists\texttt{Attends}.(\texttt{Course} \sqcap \neg A \sqcap \neg B)
\end{array}
$$

(Why isn't it satisfactory?)

Number restrictions

- How to express in $\mathcal{ALC}$ that a student attends to at least three courses?

$$\begin{array}{lcl} \texttt{Student} & \sqsubseteq & \exists\texttt{Attends}.(\texttt{Course} \sqcap A) \\ \texttt{Student} & \sqsubseteq & \exists\texttt{Attends}.(\texttt{Course} \sqcap \neg A \sqcap B) \\ \texttt{Student} & \sqsubseteq & \exists\texttt{Attends}.(\texttt{Course} \sqcap \neg A \sqcap \neg B) \end{array}$$

(Why isn't it satisfactory?)

- How to express in $\mathcal{ALC}$ that a student attends to at most 10 courses?

Number restrictions

- How to express in $\mathcal{ALC}$ that a student attends to at least three courses?

$$\begin{array}{lcl} \texttt{Student} & \sqsubseteq & \exists\texttt{Attends}.(\texttt{Course} \sqcap A) \\ \texttt{Student} & \sqsubseteq & \exists\texttt{Attends}.(\texttt{Course} \sqcap \neg A \sqcap B) \\ \texttt{Student} & \sqsubseteq & \exists\texttt{Attends}.(\texttt{Course} \sqcap \neg A \sqcap \neg B) \end{array}$$

(Why isn't it satisfactory?)

- How to express in $\mathcal{ALC}$ that a student attends to at most 10 courses?

- There is no concept C in $\mathcal{ALC}$ such that for all interpretations $\mathcal{I}$, for all $\mathfrak{a} \in \Delta^{\mathcal{I}}$,

$$\mathfrak{a} \in C^{\mathcal{I}} \text{ iff } \mathrm{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in \texttt{Attends}^{\mathcal{I}}\}) \geq 3$$

(Unqualified) number restriction

- Extending the concepts with **number restrictions** $(\leq n\ r)$ and $(\geq m\ r)$.

- Given $\mathcal{I} \stackrel{\text{def}}{=} (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$,

$$(\leq n\ r)^{\mathcal{I}} \stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \ \mid \mathrm{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}}\}) \leq n\}$$

$$(\geq m\ r)^{\mathcal{I}} \stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \ \mid \mathrm{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}}\}) \geq m\}$$

(Unqualified) number restriction

- Extending the concepts with **number restrictions** $(\leq n\ r)$ and $(\geq m\ r)$.

- Given $\mathcal{I} \stackrel{\text{def}}{=} (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$,

$$(\leq n\ r)^{\mathcal{I}} \stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid \text{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}}\}) \leq n\}$$

$$(\geq m\ r)^{\mathcal{I}} \stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid \text{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}}\}) \geq m\}$$

- Given a logic $\mathfrak{L}$, $\mathfrak{L}\mathcal{N}$ is defined as $\mathfrak{L}$ except that (unqualified) number restrictions are added.

- In $\mathcal{ALCN}$, $(\geq 3\ \texttt{Attends}) \sqcap (\leq 10\ \texttt{Attends})$ does the job.

Qualified number restriction

- Generalising the number restrictions $(\sim\ n\, r)$.
- **Qualified number restrictions**: $(\leq n\, r \cdot C)$, $(\geq m\, r \cdot C)$.
- Given $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$,

$$(\leq n\, r{\cdot}C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid \text{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}} \text{ and } \mathfrak{b} \in C^{\mathcal{I}}\}) \leq n\}$$

$$(\geq m\, r{\cdot}C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid \text{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}} \text{ and } \mathfrak{b} \in C^{\mathcal{I}}\}) \geq m\}$$

Qualified number restriction

- Generalising the number restrictions $(\sim\ n\, r)$.
- **Qualified number restrictions**: $(\leq n\, r \cdot C)$, $(\geq m\, r \cdot C)$.
- Given $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$,

  $$(\leq n\, r \cdot C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid \text{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}} \text{ and } \mathfrak{b} \in C^{\mathcal{I}}\}) \leq n\}$$

  $$(\geq m\, r \cdot C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid \text{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}} \text{ and } \mathfrak{b} \in C^{\mathcal{I}}\}) \geq m\}$$

- $(\sim n\, r) = (\sim n\, r \cdot \top)$.
- Given a logic $\mathfrak{L}$, $\mathfrak{L}\mathcal{Q}$ is defined as $\mathfrak{L}$ except that qualified number restrictions are added.

Qualified number restriction

- Generalising the number restrictions ($\sim n\ r$).
- **Qualified number restrictions**: $(\leq n\ r \cdot C)$, $(\geq m\ r \cdot C)$.
- Given $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$,

  $(\leq n\ r{\cdot}C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid \text{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}} \text{ and } \mathfrak{b} \in C^{\mathcal{I}}\}) \leq n\}$

  $(\geq m\ r{\cdot}C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid \text{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}} \text{ and } \mathfrak{b} \in C^{\mathcal{I}}\}) \geq m\}$
- $(\sim n\ r) = (\sim n\ r \cdot \top)$.
- Given a logic $\mathfrak{L}$, $\mathfrak{L}\mathcal{Q}$ is defined as $\mathfrak{L}$ except that qualified number restrictions are added.
- Concept satisfiability for $\mathcal{ALCIQ}$ is PSPACE-complete and knowledge base consistency is EXPTIME-complete.

Recapitulation

Recapitulation: concept and role constructors

Name	Syntax	Semantics
Top	$\top$	$\Delta^{\mathcal{I}}$
Bottom	$\bot$	$\emptyset$
Conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Existential restr.	$\exists r.C$	$\{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(\mathfrak{a}) \cap C^{\mathcal{I}} \neq \emptyset\}$
Value restr.	$\forall r.C$	$\{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(\mathfrak{a}) \subseteq C^{\mathcal{I}}\}$
Unqual. nb. restr.	$(\leq n\ r)$	$\{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid \text{card}(\{\mathfrak{b} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}}\}) \leq n\}$
Qual. nb. restr.	$(\leq n\ r \cdot C)$	$\{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid \text{card}(\{\mathfrak{b} \in C^{\mathcal{I}} \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}}\}) \leq n\}$
Nominal*	$\{a\}$	$\{a^{\mathcal{I}}\}$
Role value map*	$r \sqsubseteq s$	$\{\mathfrak{a} \in \Delta^{\mathcal{I}} \mid r^{\mathcal{I}}(\mathfrak{a}) \subseteq s^{\mathcal{I}}(\mathfrak{a})\}$
Inverse role	r^-	$\{(\mathfrak{b}, \mathfrak{a}) \mid (\mathfrak{a}, \mathfrak{b}) \in r^{\mathcal{I}}\}$
Role composition*	$r \circ s$	$\{(\mathfrak{a}, \mathfrak{b}) \mid \exists\ \mathfrak{a}'\ (\mathfrak{a}, \mathfrak{a}') \in r^{\mathcal{I}} \text{ and } (\mathfrak{a}', \mathfrak{b}) \in s^{\mathcal{I}}\}$

*: See next lecture

Recapitulation: Terminological and assertional axioms

Name	Syntax	Semantics
General concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Concept definition	$A \equiv C$	$A^{\mathcal{I}} = C^{\mathcal{I}}$
Role inclusion*	$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$
Role transitivity*	$\mathtt{Trans}(r)$	$r^{\mathcal{I}}$ is transitive
Concept assertion	$a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
Role assertion	$(a, b) : r$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

*: See next lecture

Conclusion

- Today lecture: Introduction to description logics
  - Getting familiar with DL terminology.
  - Playing with syntax and decision problems.
- Next week lecture: Introduction and Properties.
  - Extensions of $\mathcal{ALC}$ (Part II).
  - Relationships with first-order logic.
  - Tree model property.