Description Logics: a Nice Family of Logics — Introduction, Part 1 —

Uli Sattler[1] *Thomas Schneider*[2]

[1]School of Computer Science, University of Manchester, UK

[2]Department of Computer Science, University of Bremen, Germany

ESSLLI, 6 August 2012

What's in this course?

1. Introduction
   - Origins, the basic DL $\mathcal{ALC}$, reasoning problems
   - Relation with other logics, ontologies, examples and exercises
2. Tableau algorithms
3. Automata-based decision procedures
4. Complexity of selected DLs
   - upper bounds, lower bounds, undecidability
   - a polynomial DL: $\mathcal{EL}$
5. Other reasoning problems
   - Justifications
   - Modularity

Welcome!

Let us know if you

... have questions. – **Do ask** them at any time.

... have difficulties understanding us/reading our writing/...

... find this course too slow/boring.

... find this course too fast/difficult.

In this course, we'll

... ask you to **think** a lot

... ask you to **work** through numerous examples

... talk about **complex** stuff with many **fascinating** facets!

Plan for today

1. Origins of DLs
2. DL basics
3. Relationship with other logics
4. Ontologies
5. OWL and DLs

And now ...

1. Origins of DLs
2. DL basics
3. Relationship with other logics $\rightsquigarrow$ Uli!
4. Ontologies
5. OWL and DLs

DLs: where they come from

DLs as **knowledge representation (KR)** formalisms

- Common perception: **logic** is difficult for human conception
  - e.g., how long does it take you to read

$$\forall x \exists y \forall z((r(x,y) \wedge s(y,z)) \Rightarrow (\neg s(a,y) \vee r(x,z)))$$

  - or check that it is equivalent to

$$\forall x \exists y \forall z(r(x,z) \vee \neg r(x,y) \vee \neg s(y,z) \vee \neg s(a,y))$$

$\leadsto$ It's like a new language to learn!

Only for the "mathematically gifted"

- Are there better suited alternatives?
- Can we help users learn/speak/interact with logic?

They might not even have to see it.

Early KR formalisms

. . . were mostly **graphical** because graphics are

- easier to grasp:

  "A picture says more than a thousand words."

- close to how knowledge is represented in human beings (?)

Most graphical KR formalisms represent knowledge as **graphs** with

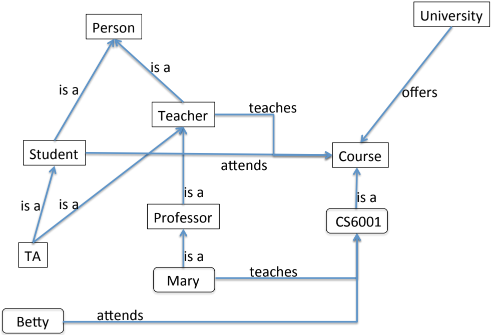
- vertices (possibly labelled)

  mostly representing concepts, classes, individuals etc.

- edges (possibly labelled)

  mostly representing properties, relationships etc.

A Semantic Network



What does it represent/say? Is Betty a Student?

Problem: missing semantics
Remedy: base your picture on logic or use logic directly

Terminological Knowledge

DLs: designed to represent **terminological** or **conceptual knowledge**

Goal

- Formalise basic terminology of an application domain; store it in a **TBox**
- Enable reasoning about **concepts** – e.g.:
  - Can there be Mammals?
  - Is every Mammal an Animal?
  - Are Frogs Reptiles?
- Store facts about individuals in an **ABox**
- Enable reasoning about **individuals** and **concepts** – e.g.:
  - Are my facts consistent with my terminology?
  - Is Kermit a Frog?

Applications

Medical

- **SNOMED CT**
  (Systematized Nomenclature of Medicine – Clinical Terms)
  - clinical terminology, used internationally
  - 450,000 terms
- **NCI Thesaurus** (NCI = National Cancer Institute of the USA)
  - vocabulary for clinical care, translational and basic research, public information, administrative activities
  - 75,000 terms
- **ICD 11** (International Classification of Diseases)
  used worldwide for health statistics

Applications

Biology

- **GO** (Gene Ontology)

  controlled vocabulary of terms for gene product characteristics and gene product annotation data

  Bioportal

  website that provides access to 255 bio-health ontologies

Semantic Web

- Use terms defined in a TBox to annotate (linked open) data
- Use TBox when querying data

And now . . .

1. Origins of DLs
2. DL basics
3. Relationship with other logics $\rightsquigarrow$ Uli!
4. Ontologies
5. OWL and DLs

DLs: the core

Core part of a DL: its **concept language**, e.g.:

$$\texttt{Animal} \sqcap \exists\, \texttt{hasPart.Feather}$$

describes all animals that are related via "`hasPart`" to a feather.

Syntactic ingredients of a concept language:

- **Concept names stand for sets of elements**, e.g., `Animal`
- **Role names** stand for binary relations between elements, e.g., `hasPart`
- **Constructors** to build **concept expressions**, e.g., $\sqcap$, $\exists$

Syntax and semantics of $\mathcal{ALC}$

Semantics given by means of an **interpretation** $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where

- $\Delta^{\mathcal{I}}$ is a nonempty set (the **domain**), and
- $\cdot^{\mathcal{I}}$ is a mapping (the **interpretation function**) as follows:

Constructor	Syntax	Example	Semantics
concept name	A	`Human`	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
role name	r	`likes`	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

For C, D concepts and R a role name:

conjunction	$C \sqcap D$	$\texttt{Human} \sqcap \texttt{Male}$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$\texttt{Nice} \sqcup \texttt{Rich}$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation	$\neg C$	$\neg\texttt{Meat}$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
restrictions:			
existential	$\exists r.C$	$\exists\texttt{hasChild.Human}$	$\{x \mid \exists y.(x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
value	$\forall r.C$	$\forall\texttt{hasChild.Blond}$	$\{x \mid \forall y.(x,y) \in r^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$

Understanding syntax and semantics of $\mathcal{ALC}$

We can “draw” interpretations . . .
(similarly to Kripke models if you happen to know modal logic)

Exercise 1: Formulate $\mathcal{ALC}$ concepts that describe

1. happy pet owners
2. unhappy pet owners who own an old cat
3. pet owners who own a cat, a dog, and only cats and dogs
4. pet owners who own a cat, a dog, and no other animals
5. everything (abbreviated by $\top$ with $\top^\mathcal{I} = \Delta^\mathcal{I}$)
6. nothing (abbreviated by $\bot$ with $\bot^\mathcal{I} = \emptyset^\mathcal{I}$)

For each of your concepts (1)–(4),
“draw” an interpretation with an instance of that concept.

Basic reasoning problems in $\mathcal{ALC}$

Definition: let C, D be $\mathcal{ALC}$ concepts. We say that

- $e \in C^{\mathcal{I}}$ is **an instance of** C in $\mathcal{I}$.
- C is **satisfiable** if there is an interpretation $\mathcal{I}$ with $C^{\mathcal{I}} \neq \emptyset$.
- C is **subsumed by** D (written $\emptyset \models C^{\mathcal{I}} \sqsubseteq D^{\mathcal{I}}$) if: for every interpretation $\mathcal{I}$, we have that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Exercise 2: Which of the following concepts is satisfiable? Which is subsumed by which?

(1) $\exists r.(A \sqcap B)$ (2) $\exists r.(A \sqcup B)$

(3) $\forall r.(A \sqcap B)$ (4) $\exists r.(A \sqcap \neg A)$

(5) $\exists r.A \sqcap \forall r.B$ (6) $\exists r.A$

(7) $\exists r.A \sqcap \forall r.\neg A$ (8) $\exists r.A \sqcap \forall s.\neg A$

The TBox

Definition

- A **general concept inclusion (GCI)** has the form $C \sqsubseteq D$, for C, D (possibly complex) concepts
- A **general TBox** is a finite set of GCIs: $\mathcal{T} = \{C_i \sqsubseteq D_i \mid 1 \leqslant i \leqslant n\}$
- $\mathcal{I}$ **satisfies** $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$ (written $\mathcal{I} \models C \sqsubseteq D$)
- $\mathcal{I}$ is a **model of TBox** $\mathcal{T}$ if $\mathcal{I}$ satisfies every $C_i \sqsubseteq D_i$
- We use $C \equiv D$ to abbreviate $C \sqsubseteq D$, $D \sqsubseteq C$

Example:

$$
\begin{aligned}
\{\ \mathtt{Father} &\equiv \mathtt{Man} \sqcap \exists\,\mathtt{hasChild.Human}\,, \\
\mathtt{Human} &\equiv \mathtt{Mammal} \sqcap \forall\,\mathtt{hasParent.Human}\,, \\
\exists\,\mathtt{favourite.Brewery} &\sqsubseteq \exists\,\mathtt{drinks.Beer}\ \}
\end{aligned}
$$

Exercise 3: Draw a model of the above TBox.
Draw an interpretation that is **not** a model of it.

Reasoning problems with respect to a TBox

Definition: let C, D be concepts, $\mathcal{T}$ a TBox. We say that

- C is **satisfiable w.r.t. $\mathcal{T}$**
  if there is a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$
- C is **subsumed by D w.r.t. $\mathcal{T}$** (written $\mathcal{T} \models C \sqsubseteq D$)
  if, for every model $\mathcal{I}$ of $\mathcal{T}$, we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

Example:

$$\mathcal{T} = \{ \begin{array}{rcl} A & \sqsubseteq & B \sqcap \exists r.C\,, \\ \exists r.\top & \sqsubseteq & \neg A \end{array} \}$$

Exercise 4: Does $\mathcal{T}$ have a model?
Are all concept names in $\mathcal{T}$ satisfiable?
Any subsumptions that you can point out?

How many models does a TBox have?

The ABox

TBox
- captures knowledge on a general, conceptual level
- contains concept def.s + general axioms about concepts

ABox
- captures knowledge on an individual level
- is a finite set of
  - **concept assertions** $a : C$ e.g., $\texttt{John} : \texttt{Man}$, and
  - **role assertions** $(a, b) : r$ e.g., $(\texttt{John}, \texttt{Mary}) : \texttt{hasChild}$

Semantics: an interpretation $\mathcal{I}$

- maps each **individual name** e to some $e^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
- satisfies a concept assertion $a : C$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- satisfies a role assertion $(a, b) : r$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$
- is a **model** of an ABox $\mathcal{A}$ if $\mathcal{I}$ satisfies each assertion in $\mathcal{A}$

$a : C$ is **entailed by** $\mathcal{A}$ if every model of $\mathcal{A}$ satisfies $a : C$

The ABox

Semantics: an interpretation $\mathcal{I}$

repeated from previous slide

- maps each **individual name** e to some $e^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
- satisfies a concept assertion $a : C$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- satisfies a role assertion $(a, b) : r$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$
- is a **model** of an ABox $\mathcal{A}$ if $\mathcal{I}$ satisfies each assertion in $\mathcal{A}$

$a : C$ is **entailed by** $\mathcal{A}$ if every model of $\mathcal{A}$ satisfies $a : C$

Example:

$$\begin{aligned} \mathcal{A} = \{ \quad & a : B \sqcap \exists r.C\,, \\ & b : A \sqcap \neg P \sqcap \forall s.\forall r.F\,, \\ (b, a) & : s \ \} \end{aligned}$$

Exercise 5: Does $\mathcal{A}$ have a model? – Describe some of them.
Can you see any entailments?

(Later) Can you translate this into FOL? ML?

Ontologies: TBox and ABox

Definition: an **ontology** consists of

- a TBox that captures knowledge on a general, conceptual level
- an ABox that captures knowledge on an individual level
  and **uses terms described in the TBox**

Notation: $(\mathcal{T}, \mathcal{A})$ or $\mathcal{T} \cup \mathcal{A}$ – no difference!

Semantics:

- Int. $\mathcal{I}$ is a **model** of $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ (written $\mathcal{I} \models \mathcal{O}$)
  if $\mathcal{I}$ satisfies each assertion and axiom in $\mathcal{O}$
  alternatively: $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$
- $\mathcal{O}$ is **consistent** if it has a model
- $\mathcal{O}$ is **coherent** if each conc. name A in $\mathcal{O}$ is satisfiable w.r.t. $\mathcal{O}$
- $C \sqsubseteq D$ is **entailed by** $\mathcal{O}$ if every model of $\mathcal{O}$ satisfies $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- $a : C$ is **entailed by** $\mathcal{O}$ if every model of $\mathcal{O}$ satisfies $a^{\mathcal{I}} \in C^{\mathcal{I}}$

Ontologies: TBox and ABox

Semantics:

repeated from previous slide

- Int. $\mathcal{I}$ is a **model** of $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ if $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$
- $\mathcal{O}$ is **consistent** if it has a model
- $\mathcal{O}$ is **coherent** if each conc. name A in $\mathcal{O}$ is satisfiable w.r.t. $\mathcal{O}$
- $C \sqsubseteq D$ is **entailed by** $\mathcal{O}$ if every model of $\mathcal{O}$ satisfies $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- $a : C$ is **entailed by** $\mathcal{O}$ if every model of $\mathcal{O}$ satisfies $a^{\mathcal{I}} \in C^{\mathcal{I}}$

Example:

$$\mathcal{O} = \{ \quad \begin{array}{rcll} A & \sqsubseteq & B \sqcap \exists r.C\,, & \qquad a : B\,, \\ \exists r.\top & \sqsubseteq & \neg A\,, & \qquad (a,b) : r \quad \} \end{array}$$

Exercise 6: Does $\mathcal{O}$ have a model? – Describe some of them.
Can you see any entailments?
What about $\mathcal{O} \cup \{b : C\}$ or $\mathcal{O} \cup \{b : A\}$?

Ontologies: TBox and ABox

Semantics:

repeated from previous slide

- Int. $\mathcal{I}$ is a **model** of $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ if $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$
- $\mathcal{O}$ is **consistent** if it has a model
- $\mathcal{O}$ is **coherent** if each conc. name A in $\mathcal{O}$ is satisfiable w.r.t. $\mathcal{O}$
- $C \sqsubseteq D$ is **entailed by** $\mathcal{O}$ if every model of $\mathcal{O}$ satisfies $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- $a : C$ is **entailed by** $\mathcal{O}$ if every model of $\mathcal{O}$ satisfies $a^{\mathcal{I}} \in C^{\mathcal{I}}$

Lemma

$C \sqsubseteq D$ is entailed by $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ iff $C \sqsubseteq D$ is entailed by $\mathcal{T}$.

Proof: for "$\Leftarrow$", note that every model of $\mathcal{O}$ is one of $\mathcal{T}$.

For "$\Rightarrow$", use contraposition; distinguish between $\mathcal{O}$ being inconsistent (trivial) and consistent (combine a model witnessing $\mathcal{T} \not\models C \sqsubseteq D$ and one of $\mathcal{O}$ to one witnessing $\mathcal{O} \not\models C \sqsubseteq D$). $\square$

And now . . .