Description Logics

Franz Baader
Theoretical Computer Science
TU Dresden
Germany

1. Motivation and introduction to Description Logics
2. Tableau-based reasoning procedures
3. Automata-based reasoning procedures
4. Complexity of reasoning in Description Logics
5. Reasoning in inexpressive Description Logics
Description Logics

Franz Baader
Theoretical Computer Science
TU Dresden
Germany

Literature:
The Description Logic Handbook
Cambridge University Press
Knowledge Representation

general goal

“develop formalisms for providing high-level descriptions of the world that can be effectively used to build intelligent applications”  
[Brachman & Nardi, 2003]

- formalism: well-defined syntax and formal, unambiguous semantics
- high-level description: only relevant aspects represented, others left out
- intelligent applications: must be able to reason about the knowledge, and infer implicit knowledge from the explicitly represented knowledge
- effectively used: need for practical reasoning tools and efficient implementations
Terminological knowledge

formalize the terminology of the application domain:

- define important notions (classes, relations, objects) of the domain
- state constraints on the way these notions can be interpreted
- deduce consequences of definitions and constraints:
  subclass relationships, instance relationships

Example: domain summer school

- classes (concepts) like Person, Lecturer, Course, Student, …
- relations (roles) like teaches, attends, likes, …
- objects (individuals) like Franz, Raj, …
- constraints like: every course must have a student, courses are only taught by lecturers, …
Description Logics

class of logic-based knowledge representation formalisms
tailed towards representing terminological knowledge

Descended from semantic networks and frames via the system KL-ONE [Brachman&Schmolze 85]. Emphasis on well-defined basic inference procedures:
subsumption and instance problem.

Phase 1:
- implementation of incomplete systems (Back, Classic, Loom)
- based on structural subsumption algorithms

Phase 2:
- development of tableau-based algorithms and complexity results
- first implementation of tableau-based systems (Kris, Crack)
- first formal investigation of optimization methods

Phase 3:
- tableau-based algorithms for very expressive DLs
- highly optimized tableau-based systems (FaCT, Racer)
- relationship to modal logic and decidable fragments of FOL
Description logic system

1. description language
   - constructors for building complex concepts out of atomic concepts and roles
   - formal, logic-based semantics

2. TBox
   - defines the terminology of the application domain

3. ABox
   - states facts about a specific “world”

4. reasoning component
   - derive implicitly represented knowledge (e.g., subsumption)
   - “practical” algorithms

knowledge base
The description language

Prototypical DL $\mathcal{ALC}$

Set $N_C$ of concept names and disjoint set $N_R$ of role names.

$\mathcal{ALC}$-concept descriptions are defined by induction:

- If $A \in N_C$, then $A$ is an $\mathcal{ALC}$-concept description.
- If $C$, $D$ are $\mathcal{ALC}$-concept descriptions, and $r \in N_R$, then the following are $\mathcal{ALC}$-concept descriptions:
  - $C \sqcap D$ (conjunction)
  - $C \sqcup D$ (disjunction)
  - $\neg C$ (negation)
  - $\forall r.C$ (value restriction)
  - $\exists r.C$ (existential restriction)

Abbreviations:
- $\top := A \sqcup \neg A$ (top)
- $\bot := A \sqcap \neg A$ (bottom)
- $C \Rightarrow D := \neg C \sqcup D$ (implication)
The description language

examples of $\mathcal{ALC}$-concept descriptions

Person $\sqcap$ Female

Person $\sqcap$ $\exists$attends.Course

Person $\sqcap$ $\forall$attends.(Course $\sqcap$ $\neg$Easy)

Person $\sqcap$ $\exists$teaches.(Course $\sqcap$ $\forall$topic.DL)

Person $\sqcap$ $\forall$teaches.(Course $\sqcap$ $\exists$topic.(DL $\sqcup$ NMR))
The description language

An interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a non-empty domain $\Delta^\mathcal{I}$ and an interpretation function $\cdot^\mathcal{I}$:

- $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ for all $A \in N_C$, concepts interpreted as sets
- $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ for all $r \in N_R$, roles interpreted as binary relations

The interpretation function is extended to $ALC$-concept descriptions as follows:

- $(C \cap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$
- $(C \cup D)^\mathcal{I} := C^\mathcal{I} \cup D^\mathcal{I}$
- $(\neg C)^\mathcal{I} := \Delta^\mathcal{I} \setminus C^\mathcal{I}$
- $(\forall r.C)^\mathcal{I} := \{ d \in \Delta^\mathcal{I} \mid \text{for all } e \in \Delta^\mathcal{I} : (d, e) \in r^\mathcal{I} \text{ implies } e \in C^\mathcal{I} \}$
- $(\exists r.C)^\mathcal{I} := \{ d \in \Delta^\mathcal{I} \mid \text{there is } e \in \Delta^\mathcal{I} : (d, e) \in r^\mathcal{I} \text{ and } e \in C^\mathcal{I} \}$
Example of an interpretation

\[
(\text{Person } \sqsubset \exists \text{teaches}. (\text{Course } \sqsubset \forall \text{topic}. \text{DL})) \uparrow = \{F\}
\]

\[
(\text{Person } \sqsubset \forall \text{teaches}. (\text{Course } \sqsubset \exists \text{topic}. (\text{DL } \sqcup \text{NMR}))) \uparrow = \{F, M\}
\]
\textit{ALC} can be seen as a fragment of first-order logic:

- Concept names are unary predicates, and role names are binary predicates.

- Concept descriptions $C$ yield formulae with one free variable $\tau_x(C)$:
  
  - $\tau_x(A) := A(x)$ for $A \in N_C$
  - $\tau_x(C \cap D) := \tau_x(C) \land \tau_x(D)$
  - $\tau_x(C \cup D) := \tau_x(C) \lor \tau_x(D)$
  - $\tau_x(\neg C) := \neg \tau_x(C)$
  - $\tau_x(\forall r. C) := \forall y. (r(x, y) \rightarrow \tau_y(C))$ \quad \text{\small \text{\it y variable different from} x}
  - $\tau_x(\exists r. C) := \exists y. (r(x, y) \land \tau_y(C))$

$C$ and $\tau_x(C)$ have the same semantics:

$$C^I = \{ d \in \Delta^I \mid I \models \tau_x(C)[x \leftarrow d] \}$$
Relationship with First-order Logic

$\mathcal{ALC}$ can be seen as a fragment of first-order logic:

- Concept names are unary predicates, and role names are binary predicates.
- Concept descriptions $C$ yield formulae with one free variable $\tau_x(C)$:

  These formulae belong to known decidable subclasses of first-order logic:
  - two-variable fragment
  - guarded fragment

\[
\begin{align*}
\tau_x(\forall r.(A \sqcap \exists r.B)) &= \forall y.(r(x, y) \rightarrow \tau_y(A \sqcap \exists r.B)) \\
&= \forall y.(r(x, y) \rightarrow (A(y) \land \exists z.(r(y, z) \land B(z)))))
\end{align*}
\]
\textbf{Relationship with Modal Logic}

\(\mathcal{ALC}\) is a syntactic variant of the basic modal logic \(K\):

- Concept names are propositional variables, and role names are names for transition relations.

- Concept descriptions \(C\) yield modal formulae \(\theta(C')\):
  - \(\theta(A) := a\) for \(A \in N_C\)
  - \(\theta(C \land D) := \theta(C') \land \theta(D)\)
  - \(\theta(C \lor D) := \theta(C') \lor \theta(D)\)
  - \(\theta(\neg C') := \neg \theta(C')\)
  - \(\theta(\forall r. C') := \Box_r \theta(C')\)
  - \(\theta(\exists r. C') := \Diamond_r \theta(C')\)

\(C'\) and \(\theta(C')\) have the same semantics: \(C' \models \theta(C')\) is the set of worlds that make \(\theta(C')\) true in the Kripke structure described by \(\mathcal{I}\).
**Additional constructors**

\( ALC \) is only an example of a description logic.

DL researchers have introduced and investigated many additional constructors.

**Example**

Number restrictions: \( (\geq n \text{ r}.C) \), \( (\leq n \text{ r}.C) \) with semantics

\[
(\geq n \text{ r}.C) := \{ d \in \Delta^I \mid \text{card}(\{ e \mid (d, e) \in r^I \land e \in C^I \}) \geq n \} \\
(\leq n \text{ r}.C) := \{ d \in \Delta^I \mid \text{card}(\{ e \mid (d, e) \in r^I \land e \in C^I \}) \leq n \}
\]

Persons that attend at most 3 courses, of which at least 2 have the topic DL:

\[
\text{Person} \sqcap (\leq 3 \text{ attends.COURSE}) \sqcap (\geq 2 \text{ attends.}(\text{Course} \sqcap \exists \text{topic.DL}))
\]
Additional constructors

In addition to concept constructors, one can also introduce role constructors.

Example

Inverse roles: if $r$ is a role, then $r^{-1}$ denotes its inverse

$$(r^{-1})^I := \{(e, d) \mid (d, e) \in r^I\}$$

Inverse roles can be used like role names in value and existential restrictions.

Teacher of a boring course:

Person $\sqcap \exists$teaches.(Course $\sqcap \forall$attends$^{-1}$.(Bored $\sqcup$ Sleeping))
**Terminologies**

introduce names for complex descriptions

A concept definition is of the form $A \equiv C$ where

- $A$ is a concept name;
- $C$ is a concept description.

A **TBox** is a finite set of concept definitions that

- does not contain multiple definitions;
- does not contain cyclic definitions.

$\begin{align*}
A &\equiv C \\
A &\equiv D \\
A &\equiv B \cap \forall r.P \\
B &\equiv P \cap \forall r.C \\
C &\equiv \exists r.A
\end{align*}$

**Defined concept** occurs on left-hand side of a definition

**Primitive concept** does not occur on left-hand side of a definition
An interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T}$ if it satisfies all its concept definitions:

$$A^\mathcal{I} = C^\mathcal{I} \text{ for all } A \equiv C \in \mathcal{T}$$

- Woman $\equiv$ Person $\sqcap$ Female
- Man $\equiv$ Person $\sqcap$ $\neg$Female
- Course $\equiv$ $\exists$topic.$\top$
- Lecturer $\equiv$ Person $\sqcap$ $\exists$teaches.Course
- Student $\equiv$ Person $\sqcap$ $\exists$attends.Course
- BusyLecturer $\equiv$ Lecturer $\sqcap$ ($\geq 3$ teaches.Course)
- BadLecturer $\equiv$ Lecturer $\sqcap$ $\forall$teaches.($\forall$attends$^{-1}$.Bored $\sqcup$ Sleeping)
Terminologies

Modern DL systems allow their users to state more general constraints for the interpretation of concepts.

A general concept inclusion axiom (GCI) is of the form $C \subseteq D$ where $C$, $D$ may be complex concept descriptions.

**general TBox**

An interpretation $\mathcal{I}$ is a model of a set of GCIs $\mathcal{T}$ if it satisfies all its concept inclusions:

$$C^\mathcal{I} \subseteq D^\mathcal{I} \text{ for all } C \subseteq D \in \mathcal{T}$$

- Course $\cap \forall\text{attends}^{-1}.\text{Sleeping} \subseteq \text{Boring}$
- Lecturer $\cap \text{Student} \subseteq \bot$
ABox assertions

state properties of individuals

An assertion is of the form

\[ C(a) \text{ (concept assertion)} \quad \text{or} \quad r(a, b) \text{ (role assertion)} \]

where \( C \) is a concept description, \( r \) is a role, and \( a, b \) are individual names from a set \( N_I \) of such names.

An ABox is a finite set of assertions.

An interpretation \( \mathcal{I} \) is a model of an ABox \( \mathcal{A} \) if it satisfies all its assertions:

\[ a^\mathcal{I} \in C^\mathcal{I} \quad \text{for all } C(a) \in \mathcal{A} \]
\[ (a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I} \quad \text{for all } r(a, b) \in \mathcal{A} \]

\( \mathcal{I} \) assigns elements of \( \Delta^\mathcal{I} \) to individual names

Lecturer(FRANZ), teaches(FRANZ, C1), Course(C1), topic(C1, T1), DL(T1)
Reasoning makes implicitly represented knowledge explicit, provided as service by the DL system, e.g.:

- **Subsumption**: Is $C$ a subconcept of $D$?
  \[ C \sqsubseteq_T D \text{ iff } C^I \subseteq D^I \text{ for all models } I \text{ of the TBox } T. \]

- **Satisfiability**: Is the concept $C$ non-contradictory?
  \[ C \text{ is satisfiable w.r.t. } T \text{ iff } C^I \neq \emptyset \text{ for some model } I \text{ of } T. \]

- **Consistency**: Is the ABox $A$ non-contradictory?
  \[ A \text{ is consistent w.r.t. } T \text{ iff it has a model that is also a model of } T. \]

- **Instantiation**: Is $e$ an instance of $C$?
  \[ A \models_T C(e) \text{ iff } e^I \in C^I \text{ for all models } I \text{ of } T \text{ and } A. \]
Reductions between inference problems

Subsumption to satisfiability:
\[ C \sqsubseteq_T D \text{ iff } C \cap \neg D \text{ is unsatisfiable w.r.t. } T \]

Satisfiability to subsumption:
\[ C \text{ is satisfiable w.r.t. } T \text{ iff } \neg C \sqsubseteq_T \bot \]

Satisfiability to consistency:
\[ C \text{ is satisfiable w.r.t. } T \text{ iff } \{ C(a) \} \text{ is consistent w.r.t. } T \]

Instance to consistency:
\[ a \text{ is an instance of } C \text{ w.r.t. } T \text{ and } A \text{ iff } A \cup \{ \neg C(a) \} \text{ is inconsistent w.r.t. } T \]

Consistency to instance:
\[ A \text{ is consistent w.r.t. } T \text{ iff } a \text{ is not an instance of } \bot \text{ w.r.t. } T \text{ and } A \]
Reduction

getting rid of the TBox

Expansion of concepts:

For a given TBox $\mathcal{T}$ and concept description $C$, the expansion $C^\mathcal{T}$ of $C$ w.r.t. $\mathcal{T}$ is obtained from $C$ by

- replacing defined concepts by their definitions
- until no more defined concepts occur.

$$
\begin{array}{ll}
\text{Woman} & \equiv \text{Person} \sqcap \text{Female} \\
\text{Course} & \equiv \exists \text{topic. } \mathcal{T} \\
\text{Lecturer} & \equiv \text{Person} \sqcap \exists \text{teaches. } \text{Course}
\end{array}
$$

Woman $\sqcap$ Lecturer expands to

\[
\text{Person} \sqcap \text{Female} \sqcap \text{Person} \sqcap \exists \text{teaches.}(\exists \text{topic. } \mathcal{T})
\]
Reduction

getting rid of the TBox

Since TBoxes are acyclic, expansion always terminates,
but the expanded concept may be exponential in the size of $\mathcal{T}$.

\[
\begin{align*}
A_0 & \equiv \forall r. A_1 \land \forall s. A_1 \\
A_1 & \equiv \forall r. A_2 \land \forall s. A_2 \\
& \vdots \\
A_{n-1} & \equiv \forall r. A_n \land \forall s. A_n
\end{align*}
\]

The size of $\mathcal{T}$ is linear in $n$,
but the expansion $A_0^T$ contains $A_n$ $2^n$ times.

Reductions:

- $C$ is satisfiable w.r.t. $\mathcal{T}$ \IFF $C^T$ is satisfiable w.r.t. the empty TBox $\emptyset$.
- $C \sqsubseteq_{\mathcal{T}} D$ \IFF $C^T \sqsubseteq_{\emptyset} D^T$.
- Consistency and the instance problem can be treated similarly.
Classification

Computing the subsumption hierarchy of all concept names occurring in the TBox.

\[
\begin{align*}
\text{Man} & \equiv \text{Person} \sqcap \neg \text{Female} \\
\text{Woman} & \equiv \text{Person} \sqcap \text{Female} \\
\text{MaleLecturer} & \equiv \text{Man} \sqcap \exists \text{teaches.Course} \\
\text{FemaleLecturer} & \equiv \text{Woman} \sqcap \exists \text{teaches.Course} \\
\text{Lecturer} & \equiv \text{FemaleLecturer} \sqcup \text{MaleLecturer} \\
\text{BusyLecturer} & \equiv \text{Lecturer} \sqcap (\geq 3 \text{teaches.Course})
\end{align*}
\]

Diagram:

- Course
  - Person
    - Male
    - Female
    - MaleLecturer
    - Lecturer
    - BusyLecturer
    - FemaleLecturer
  - Woman
Realization

Computing the most specific concept names in the TBox to which an ABox individual belongs.

\[
\begin{align*}
\text{Man} & \equiv \text{Person} \cap \neg \text{Female} \\
\text{Woman} & \equiv \text{Person} \cap \text{Female} \\
\text{MaleLecturer} & \equiv \text{Man} \cap \exists \text{teaches}.\text{Course} \\
\text{FemaleLecturer} & \equiv \text{Woman} \cap \exists \text{teaches}.\text{Course} \\
\text{Lecturer} & \equiv \text{FemaleLecturer} \sqcup \text{MaleLecturer} \\
\text{BusyLecturer} & \equiv \text{Lecturer} \cap (\geq 3 \text{ teaches}.\text{Course})
\end{align*}
\]

\[
\text{Man(FRANZ), teaches(} FRANZ, \text{C1), Course(C1)}
\]

FRANZ is an instance of \text{Man, Lecturer, MaleLecturer.}

most specific