A Description Logic Journey

Franz Baader
Theoretical Computer Science
TU Dresden, Germany

Navy Art Collection
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
Description Logics

- Family of logic-based knowledge representation languages tailored towards representing terminological knowledge
- Many DLs are decidable fragments of first-order logic
- Close relationship to propositional modal logics
- Design goal: good compromise between expressiveness and complexity
- Decidability and complexity results for a great variety of DLs and various inference problems, but also implementation of practical systems
  - very expressive DLs of high worst-case complexity, but with highly optimized “practical” reasoning procedures
  - inexpressive DLs with tractable inference problems, which are expressive enough for certain applications
- Applications: natural language processing, configuration, databases, modelling in engineering domains, ontologies (Web ontology language OWL, biomedical ontologies)

© Franz Baader
Description Logics \textbf{from a general point of view}

\textbf{Concepts}

- **Constructors** for building complex concept descriptions out of atomic concepts (unary predicates) and roles (binary predicates).
- **Interpretation** $\mathcal{I}$ assigns sets $C^\mathcal{I}$ to concept descriptions $C$ according to the semantics of the constructors.

\textbf{TBoxes}

- Finite set of general concept inclusions (GCIs) of the form $C \subseteq D$ where $C, D$ are concept descriptions.
- The interpretation $\mathcal{I}$ is a \textbf{model} of a TBox $\mathcal{T}$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$ holds for all GCIs $C \subseteq D$ in $\mathcal{T}$.

\textbf{ABoxes}

- Finite set of assertions of the form $C(a)$ and $r(a, b)$ where $C$ is a concept description, $r$ a role, and $a, b$ individual names.
- The interpretation $\mathcal{I}$ is a \textbf{model} of an ABox $\mathcal{A}$ if $a^\mathcal{I} \in C^\mathcal{I}$ and $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$ holds for all assertions $C(a)$ and $r(a, b)$ in $\mathcal{A}$.
Description Logics from a general point of view

**Concepts**

- Constructors for building complex concept descriptions out of atomic concepts (unary predicates) and roles (binary predicates).
- Interpretation $\mathcal{I}$ assigns sets $C^\mathcal{I}$ to concept descriptions $C$ according to the semantics of the constructors.

**Restricted TBoxes**

- Finite set of general concept inclusions (GCIs) of the form $A \equiv D$ where $A$ is a concept name occurring only once as left-hand side.
- The interpretation $\mathcal{I}$ is a model of a restricted TBox $\mathcal{T}$ if $A^\mathcal{I} = D^\mathcal{I}$ holds for all definitions $A \equiv D$ in $\mathcal{T}$.

**ABoxes**

- Finite set of assertions of the form $C(a)$ and $r(a, b)$ where $C$ is a concept description, $r$ a role, and $a, b$ individual names.
- The interpretation $\mathcal{I}$ is a model of an ABox $\mathcal{A}$ if $a^\mathcal{I} \in C^\mathcal{I}$ and $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$ holds for all assertions $C(a)$ and $r(a, b)$ in $\mathcal{A}$.
Constructors of the DL $\mathcal{ALC}$

- top concept $\top$, negation $\neg C$
- conjunction $C \sqcap D$, disjunction $C \sqcup D$
- existential restriction $\exists r.C$, value restriction $\forall r.C$

An advanced course that has a smart or studious student, no easy topic, and a teacher

- $Course \sqcap Advanced \sqcap$
- $\exists has\_student. (Smart \sqcap Studious) \sqcap$
- $\forall has\_topic. \neg Easy \sqcap$
- $\exists has\_teacher. \top$

**TBox**
- Concept definition
  - $Good\_course \equiv Course \sqcap \ldots$
- General concept inclusion (GCI)
  - $\exists has\_student. \top \sqsubseteq Course$

**ABox**
- properties of individuals
  - $Good\_Course(Course123)$
  - $has\_teacher(Course123, Franz)$
  - $has\_topic(Course123, DL)$
Constructors of the DL $\mathcal{ALC}$

- top concept $\top$, negation $\neg C$
- conjunction $C \sqcap D$, disjunction $C \sqcup D$
- existential restriction $\exists r.C$
- value restriction $\forall r.C$

An advanced course that $\text{Course} \sqcap \text{Advanced}$

has a smart or studious student, $\exists \text{has\_student}.(\text{Smart} \sqcup \text{Studious})$

no easy topic, $\forall \text{has\_topic}.\neg \text{Easy}$

and a teacher $\exists \text{has\_teacher}.\top$

The semantics of the constructors is defined through identities:

- $(C \sqcap D)^I = C^I \cap D^I$,
- $(\exists r.C)^I = \{ d \mid \exists e. (d, e) \in r^I \land e \in C^I \}$,
- $(\forall r.C)^I = \{ d \mid \forall e. (d, e) \in r^I \rightarrow e \in C^I \}$,
- ...
Reasoning makes implicitly represented knowledge explicit, provided as service by the DL system, e.g.:

**Subsumption:** Is $C$ a subconcept of $D$?

$$C \subseteq_{\mathcal{T}} D \iff C^\mathcal{I} \subseteq D^\mathcal{I} \text{ for all models } \mathcal{I} \text{ of the TBox } \mathcal{T}. $$

**Satisfiability:** Is the concept $C$ non-contradictory?

$C$ is satisfiable w.r.t. $\mathcal{T}$ iff $C^\mathcal{I} \neq \emptyset$ for some model $\mathcal{I}$ of $\mathcal{T}$.

**Consistency:** Is the ABox $\mathcal{A}$ non-contradictory?

$\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$ iff it has a model that is also a model of $\mathcal{T}$.

**Instantiation:** Is $x$ an instance of $C$?

$\mathcal{A} \models_{\mathcal{T}} C(x)$ iff $x^\mathcal{I} \in C^\mathcal{I}$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$.
Focus of DL research: develop and investigate reasoning procedures

- Expressivity of language
  - application relevant concepts must be definable
  - some domains require very expressive DLs

- Complexity of reasoning
  - reasoning must be decidable/of low complexity
  - requires restricted description languages

- Practical algorithms
  - easy to implement and optimize
  - behave well in practice
A Description Logic Journey

Starting small
In the early days of DL research, $\mathcal{FL}_0$ was considered to be the smallest possible DL.

- value restriction: green is the only possible color;
  \[
  \text{Frog} \sqsubseteq \text{Animal} \sqcap \forall \text{color}.\text{Green}
  \]

- existential restriction: green is one of its colors.
  \[
  \text{Frog} \sqsubseteq \text{Animal} \sqcap \exists \text{color}.\text{Green}
  \]
In the early days of DL research, $\mathcal{FL}_0$ was considered to be the smallest possible DL.

- Value restriction: green is the only possible color;
- Existential restriction: green is one of its colors.

$\mathcal{FL}_0 \quad Frog \sqsubseteq Animal \sqcap \forall color. Green$

Chosen by KL-ONE and other early DL systems.
Some bad news in the late 1980ies

A commonly held belief in the 1980ies:

- reasoning in KR systems should be **tractable**, i.e., of **polynomial time complexity**

- **KL-ONE** and its early successor systems (BACK, MESON, K-Rep, ...) employed **polynomial-time algorithms**
  - sound, but **incomplete**

- reasoning in **KL-ONE** is **undecidable**  
  - [Schmidt-Schauß; 1989]

- reasoning w.r.t. a TBox is **intractable** even in the minimal DL $\mathcal{FL}_0$  
  - [Nebel; 1989]
### Complexity of subsumption reasoning $C \subseteq D$

<table>
<thead>
<tr>
<th>restricted TBox</th>
<th>general TBox</th>
</tr>
</thead>
<tbody>
<tr>
<td>no TBox</td>
<td>acyclic TBox</td>
</tr>
<tr>
<td>$\mathcal{FL}_0$</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{ALC}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $P \subseteq coNP \subseteq P$Space $\subseteq ExpTime$
  - tractable
  - not tractable unless $P \neq NP$
  - not tractable

There is no reason for restricting to $\mathcal{FL}_0$. © Franz Baader
A Description Logic Journey

Getting larger

\textit{ALC} and beyond
Extensions of $\mathcal{ALC}$ motivated by applications

- Concrete domains: refer to concrete objects (e.g., numbers) and predicates on these objects (e.g., numerical comparisons) when defining concepts:

  \[
  \text{Teenager} \equiv \text{Human} \cap (\text{age} \geq 10) \cap (\text{age} \leq 19)
  \]

  \[
  \text{Human} \subseteq (\text{mother} \circ \text{age} > \text{age}) \cap (\text{father} \circ \text{age} > \text{age})
  \]

Extensions of \textit{ALC} motivated by applications

- Concrete domains: refer to concrete objects (e.g., numbers) and predicates on these objects (e.g., numerical comparisons) when defining concepts.

- Local and global cardinality constraints: restrict the number of role successors of an object (number restrictions) or the cardinality of a concept:

  At most two sons and at least one daughter:  
  \[(\leq 2 \text{child.Male}) \sqcap (\geq 1 \text{child.Female})\]  
  [Hollunder & Baader; 1991]  
  [Tobies; 2000]

  At most 45 million cars are registered all over Germany:  
  \[(\leq 45000000 (\text{Car} \sqcap \exists \text{registered_in.German_district}))\]  
  [Baader et al.; 1996]
Extensions of $\mathcal{ALC}$ motivated by applications

- Concrete domains: refer to concrete objects (e.g., numbers) and predicates on these objects (e.g., numerical comparisons) when defining concepts.

- Local and global cardinality constraints: restrict the number of role successors of an object (number restrictions) or the cardinality of a concept.

- Transitive roles, subroles, and inverse roles: describe complex objects that are composed of different parts:

$$\text{Engine} \sqcap \exists \text{part_of}. \text{Car} \sqcap \exists \text{has_part}. \text{Distributor} \sqcap \ldots$$

The role $\text{has_part}$ is transitive, the inverse of $\text{part_of}$, and has $\text{has_strict_part}$ as a subrole.

[Sattler; 1996] [Horrocks & Sattler; 1999]
Extensions of \textit{ALC} motivated by applications

- Concrete domains: refer to concrete objects (e.g., numbers) and predicates on these objects (e.g., numerical comparisons) when defining concepts.

- Local and global cardinality constraints: restrict the number of role successors of an object (number restrictions) or the cardinality of a concept.

- Transitive roles, subroles, and inverse roles: describe complex objects that are composed of different parts.

- These and some additional features are available in the Web Ontology Language OWL 2 DL.

Highly optimized reasoning systems: \textit{FaCT, Racer, Pellet, HermiT, Sequoia, \ldots}
Concrete domains

our original motivation

[Baader & Hanschke; 1991]

describe geometric structure of rotational-symmetric lathe workpieces
Concrete domains

our original motivation

[Baader & Hanschke; 1991]

describe geometric structure of
rotational-symmetric lathe workpieces

- decompose workpiece into simple geometric components
Concrete domains  

our original motivation  

[Baader & Hanschke; 1991]

describe geometric structure of rotational-symmetric lathe workpieces

- decompose workpiece into simple geometric components
- described geometric shape of single components and how neighbouring components fit together using concrete domain predicates  
  real arithmetics
- described whole workpiece as sequence of its components using transitive closure
**Good news**

- extending $\mathcal{ALC}$ with “admissible” concrete domains leaves reasoning
  (without TBox or w.r.t. an acyclic TBox) decidable
  [Baader & Hanschke; 1991]

- extending $\mathcal{ALC}$ with transitive closure of roles leaves reasoning
  decidable
  [Baader; 1991]

**Bad news**

- combining the two extensions causes undecidability

[Baader & Hanschke; 1992]
More bad news

- adding an acyclic TBox to $\mathcal{ALC}$ with an admissible concrete domain may increase the complexity considerably
- adding a cyclic or general TBox may cause undecidability
- even for quite simple admissible concrete domains

Some good news

- extending $\mathcal{ALC}$ with $\omega$-admissible concrete domains leaves reasoning decidable even in the presence of general TBoxes
  
  [Lutz; 2002]  [Lutz & Milicic; 2007]

- model-theoretic characterizations of $\omega$-admissible concrete domains that facilitate finding new $\omega$-admissible concrete domains
  
  [Baader & Rydval; 2020]
A Description Logic Journey

Getting smaller again

EL

Navy Art Collection
In the early days of DL research, $\mathcal{FL}_0$ was considered to be the smallest possible DL.

- value restriction: green is the only possible color;
  \[
  \mathcal{FL}_0 \quad \text{Frog} \sqsubseteq \text{Animal} \land \forall \text{color}. \text{Green}
  \]

- existential restriction: green is one of its colors.
  \[
  \mathcal{EL} \quad \text{Frog} \sqsubseteq \text{Animal} \land \exists \text{color}. \text{Green}
  \]

Source: Wikimedia Author: Carey James Balboa

© Franz Baader
### Complexity of subsumption reasoning $C \subseteq D$

<table>
<thead>
<tr>
<th></th>
<th>no TBox</th>
<th>acyclic TBox</th>
<th>cyclic TBox</th>
<th>general TBox</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{FL}_0$</td>
<td>$P$</td>
<td>$\text{coNP}$</td>
<td>$\text{PSpace}$</td>
<td>$\text{ExpTime}$</td>
</tr>
<tr>
<td>$\mathcal{ALC}$</td>
<td>$\text{PSpace}$</td>
<td>$\text{PSpace}$</td>
<td>$\text{ExpTime}$</td>
<td>$\text{ExpTime}$</td>
</tr>
<tr>
<td>$\mathcal{EL}$</td>
<td>$P$</td>
<td>$P$</td>
<td>$P$</td>
<td>$P$</td>
</tr>
<tr>
<td>[Baader, Küsters, Molitor; 1999]</td>
<td>[Baader; 2003]</td>
<td>[Baader; 2003]</td>
<td>[Brandt; 2004]</td>
<td></td>
</tr>
</tbody>
</table>

Is $\mathcal{EL}$ useful in practice?
\( \mathcal{E\ell} \) a practically useful DL for which
TBox and ABox reasoning is tractable

- subsumption in \( \mathcal{E\ell} \) is polynomial
  even in the presence of general TBoxes;

- \( \mathcal{E\ell} \) is used in several biomedical ontologies
  (e.g., SNOMED CT, The Gene Ontology);

- tractability even holds for ABox reasoning and
  for interesting extensions of \( \mathcal{E\ell} \);

- a maximal such extension is the basis of the OWL 2 EL profile
  of the Web Ontology Language standard of the W3C;

- availability of several highly efficient \( \mathcal{E\ell} \) reasoners
  CEL
  Snorocket, ELK, Konclude

[Brandt; 2004]
[Suntisrivaporn; 2009]
[Baader, Brandt, Lutz; 2005]
A Description Logic Journey

Making in larger again

Horn DLs and beyond
Horn DLs and beyond

- The algorithmic approach (consequence-based reasoning) employed for $\mathcal{EL}$ can be extended to non-tractable DLs that share a certain model-theoretic property with $\mathcal{EL}$

Horn $\approx$ existence of canonical models
Horn DLs and beyond

- The algorithmic approach (consequence-based reasoning) employed for $\mathcal{EL}$ can be extended to non-tractable DLs that share a certain model-theoretic property with $\mathcal{EL}$

$\mathcal{ELI}$: extension of $\mathcal{EL}$ with inverse roles

  - The subsumption problem in $\mathcal{ELI}$ is ExpTime-complete.  
    [Baader, Brandt, Lutz; 2008]

  - The consequence-based reasoner CB is much faster on $\mathcal{ELI}$ than highly-optimized reasoners for more expressive DLs  
    [Kazakov; 2009]

- This approach has recently been extended also to non-Horn DLs up to OWL 2 DL

Consequence-based reasoner Sequoia  
[Cucala, Grau, Horrocks; 2021]
End of Journey

- Great variety of DLs with different expressive power
- Formal and algorithmic properties well investigated
- Highly optimized reasoning system
- Formal basis of OWL 2 standard
- Employed in diverse application domains