Description Logics—Basics, Applications, and More

Ian Horrocks
Information Management Group
University of Manchester, UK

Ulrike Sattler
Teaching and Research Area for Theoretical Computer Science
RWTH Aachen, Germany

Overview of the Tutorial

- **History and Basics:** Syntax, Semantics, ABoxes, Tboxes, Inference Problems and their interrelationship, and Relationship with other (logical) formalisms
- **Applications of DLs:** ER-diagrams with i.com demo, ontologies, etc. including system demonstration
- **Reasoning Procedures:** simple tableaux and why they work
- **Reasoning Procedures II:** more complex tableaux, non-standard inference problems
- **Complexity issues**
- **Implementing/Optimising DL systems**

Description Logics

- family of logic-based knowledge representation formalisms well-suited for the representation of and reasoning about

➠ **terminological knowledge**

➠ **configurations**

➠ **ontologies**

➠ **database schemata**

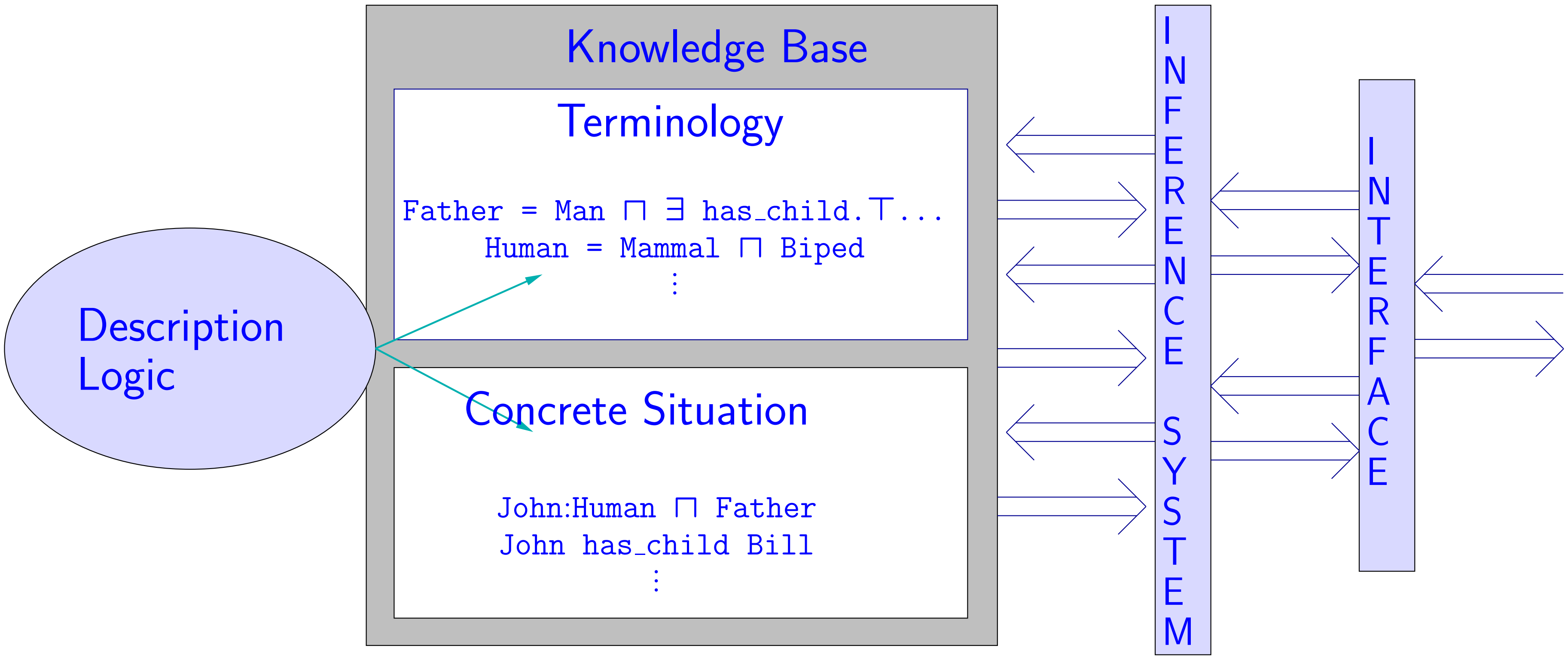
  – schema design, evolution, and query optimisation
  – source integration in heterogeneous databases/data warehouses
  – conceptual modelling of multidimensional aggregation

➠ ...

- descendents of semantics networks, frame-based systems, and KL-ONE

- aka terminological KR systems, concept languages, etc.

Architecture of a Standard DL System

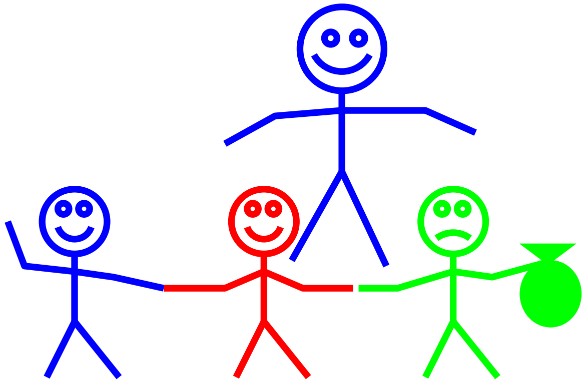


A Description Logic - mainly characterised by a **set of constructors** that allow to build complex **concepts** and **roles** from atomic ones,

concepts correspond to classes / are interpreted as **sets** of objects,

roles correspond to relations / are interpreted as **binary relations** on objects,

Example: Happy Father in the DL $\mathcal{ALC}$



$$\begin{aligned}\texttt{Man} \sqcap\ & (\exists \texttt{has-child.Blue}) \sqcap \\ & (\exists \texttt{has-child.Green}) \sqcap \\ & (\forall \texttt{has-child.Happy} \sqcup \texttt{Rich})\end{aligned}$$

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

Constructor	Syntax	Example	Semantics
atomic concept	A	Human	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
atomic role	R	likes	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
For C, D concepts and R a role name			
conjunction	$C \sqcap D$	Human $\sqcap$ Male	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	Nice $\sqcup$ Rich	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation	$\neg C$	$\neg$ Meat	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
exists restrict.	$\exists R.C$	$\exists$has-child.Human	$\{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
value restrict.	$\forall R.C$	$\forall$has-child.Blond	$\{x \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$

Constructor	Syntax	Example	Semantics
number restriction	$(\geq n\ R)$	$(\geq 7$ has-child$)$	$\{x \mid \lvert\{y.\langle x,y\rangle \in R^{\mathcal{I}}\}\rvert \geq n\}$
$(\rightsquigarrow \mathcal{ALCN})$	$(\leq n\ R)$	$(\leq 1$ has-mother$)$	$\{x \mid \lvert\{y.\langle x,y\rangle \in R^{\mathcal{I}}\}\rvert \leq n\}$
inverse role	R^{-}	has-child^{-}	$\{\langle x,y\rangle \mid \langle y,x\rangle \in R^{\mathcal{I}}\}$
trans. role	R^{*}	has-child*	$(R^{\mathcal{I}})^{*}$
concrete domain	$u_1,\ldots,u_n.P$	h-father·age, age. $>$	$\{x \mid \langle u_1^{\mathcal{I}}(x),\ldots,u_n^{\mathcal{I}}(x)\rangle \in P\}$
etc.			

Many other different DLs/DL constructors have been investigated

For terminological knowledge: **TBox** contains

Concept definitions $A \doteq C$ (A a concept name, C a complex concept)

Father $\doteq$ Man $\sqcap$ $\exists$has-child.Human

Human $\doteq$ Mammal $\sqcap$ $\forall$has-child^{-}.Human

$\leadsto$ introduce macros/names for concepts, can be (a)cyclic

Axioms $C_1 \sqsubseteq C_2$ (C_i complex concepts)

$\exists$favourite.Brewery $\sqsubseteq$ $\exists$drinks.Beer

$\leadsto$ restrict your models

An interpretation $\mathcal{I}$ **satisfies**

a **concept definition** $A \doteq C$ iff $A^{\mathcal{I}} = C^{\mathcal{I}}$

an **axiom** $C_1 \sqsubseteq C_2$ iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$

a **TBox** $\mathcal{T}$ iff $\mathcal{I}$ satisfies all definitions and axioms in $\mathcal{T}$

$\leadsto$ $\mathcal{I}$ is a **model** of $\mathcal{T}$

For assertional knowledge: **ABox** contains

Concept assertions	a : C	(a an individual name, C a complex concept)
	John : Man $\sqcap$ $\forall$has-child.(Male $\sqcap$ Happy)	
Role assertions	$\langle a_1, a_2 \rangle$: R	(a_i individual names, R a role)
	$\langle$John, Bill$\rangle$: has-child	

An interpretation $\mathcal{I}$ **satisfies**

a **concept assertion**	$a : C$	iff	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
a **role assertion**	$\langle a_1, a_2 \rangle : R$	iff	$\langle a_1^{\mathcal{I}}, a_2^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
an **ABox**	$\mathcal{A}$	iff	$\mathcal{I}$ satisfies all assertions in $\mathcal{A}$
		$\leadsto$	$\mathcal{I}$ is a **model** of $\mathcal{A}$

Subsumption: $C \sqsubseteq D$ Is $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$?

w.r.t. TBox $\mathcal{T}$: $C \sqsubseteq_{\mathcal{T}} D$ Is $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{T}$?

$\leadsto$ structure your knowledge, compute taxonomy

Consistency: Is C consistent w.r.t. $\mathcal{T}$? Is there a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$?

of ABox $\mathcal{A}$: Is $\mathcal{A}$ consistent? Is there a model of $\mathcal{A}$?

of KB ($\mathcal{T}$,$\mathcal{A}$): Is ($\mathcal{T}$,$\mathcal{A}$) consistent? Is there a model of both $\mathcal{T}$ and $\mathcal{A}$?

Inference Problems are **closely related:**

$C \sqsubseteq_{\mathcal{T}} D$ iff $C \sqcap \neg D$ is **in**consistent w.r.t. $\mathcal{T}$,
(no model of $\mathcal{I}$ has an instance of $C \sqcap \neg D$)

C is consistent w.r.t. $\mathcal{T}$ iff **not** $C \sqsubseteq_{\mathcal{T}} A \sqcap \neg A$

$\leadsto$ **Decision Procdures for consistency (w.r.t. TBoxes) suffice**

Introduction to DL: Basic Inference Problems II

For most DLs, the basic inference problems are **decidable**,
with complexities between **P** and **ExpTime**.

Why is decidability important? Why does semi-decidability not suffice?

If subsumption (and hence consistency) is undecidable, and

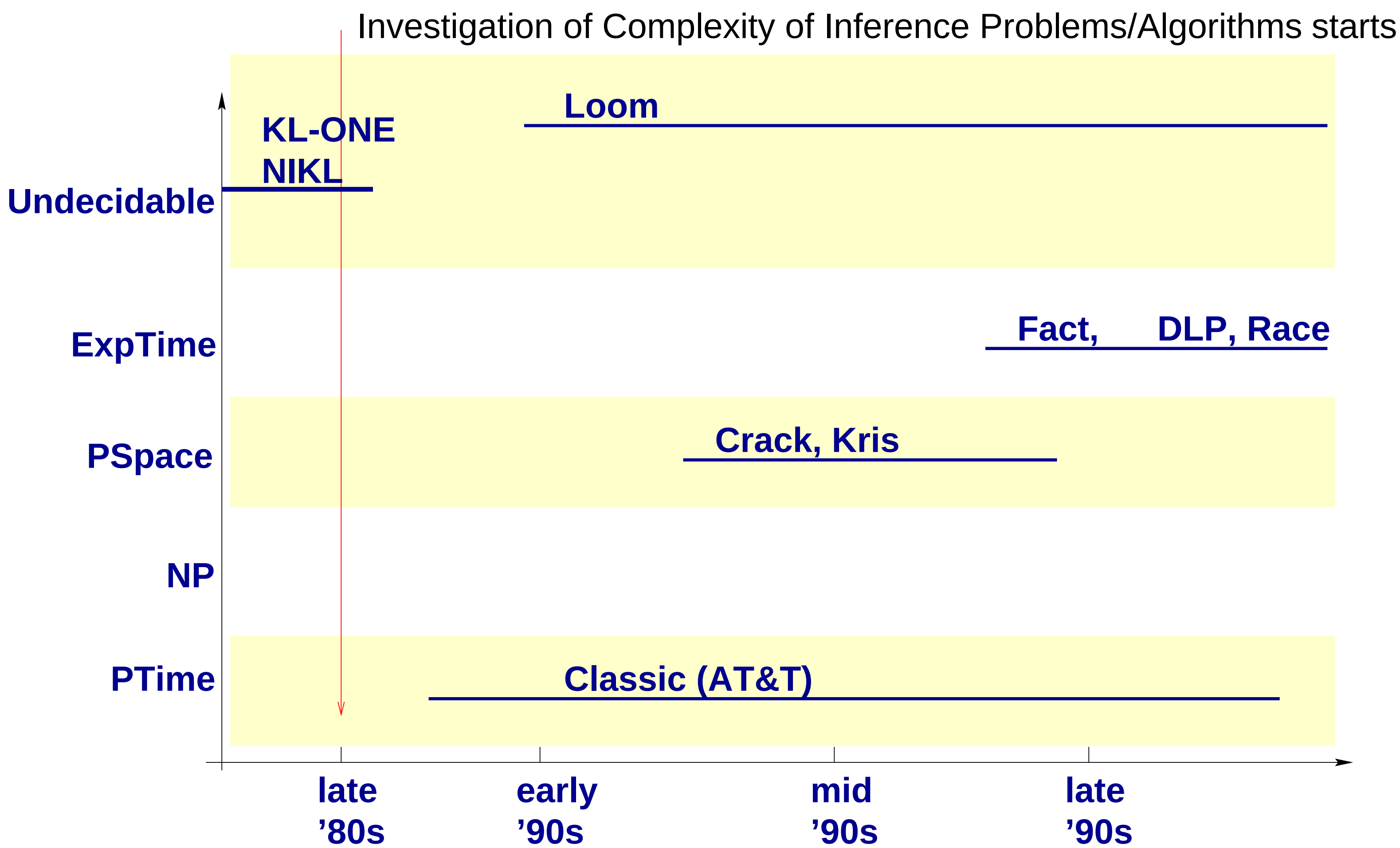
➠ subsumption is semi-decidable, then consistency is **not** semi-decidable

➠ consistency is semi-decidable, then subsumption is **not** semi-decidable

➠ Quest for a "highly expressive" DL with decidable/"practicable" inference problems

where **expressiveness** depends on the application
practicability changed over the time

Complexity of Inferences provided by DL systems over the time



In the last 5 years, DL-based systems were built that

✔ can handle DLs far more expressive than $\mathcal{ALC}$ (close relatives of converse-DPDL)

- Number restrictions: "people having at most 2 children"
- Complex roles: inverse ("has-child" — "child-of"),
  transitive closure ("offspring" — "has-child"),
  role inclusion ("has-daughter" — "has-child"), etc.

✔ implement provably sound and complete inference algorithms
(for ExpTime-complete problems)

✔ can handle large knowledge bases
(e.g., Galen medical terminology ontology: 2,740 concepts, 413 roles, 1,214 axioms)

✔ are highly optimised versions of **tableau-based** algorithms

✔ perform (surprisingly well) on benchmarks for modal logic reasoners
(Tableaux'98, Tableaux'99)

Relationship with Other Logical Formalisms: First Order Predicate Logic

Most DLs are decidable fragments of FOL: Introduce

a unary predicate ϕ_A for a concept name A
a binary relation ρ_R for a role name R

Translate complex concepts C, D as follows:

$$
\begin{aligned}
t_x(A) &= \phi_A(x), & t_y(A) &= \phi_A(y),\\
t_x(C \sqcap D) &= t_x(C) \land t_x(D), & t_y(C \sqcap D) &= t_y(C) \land t_y(D),\\
t_x(C \sqcup D) &= t_x(C) \lor t_x(D), & t_y(C \sqcup D) &= t_y(C) \lor t_y(D),\\
t_x(\exists R.C) &= \exists y.\rho_R(x,y) \land t_y(C), & t_y(\exists R.C) &= \exists x.\rho_R(y,x) \land t_x(C),\\
t_x(\forall R.C) &= \forall y.\rho_R(x,y) \Rightarrow t_y(C), & t_y(\forall R.C) &= \forall x.\rho_R(y,x) \Rightarrow t_x(C).
\end{aligned}
$$

A TBox $\mathcal{T} = \{C_i \sqsubseteq D_i\}$ is translated as

$$\Phi_{\mathcal{T}} = \forall x. \bigwedge_{1 \le i \le n} t_x(C_i) \Rightarrow t_x(D_i)$$

Relationship with Other Logical Formalisms: First Order Predicate Logic II

$$\begin{aligned}
C \text{ is consistent} &\text{ iff its translation } t_x(C) \text{ is satisfiable,}\\
C \text{ is consistent w.r.t. } \mathcal{T} &\text{ iff its translation } t_x(C) \wedge \Phi_{\mathcal{T}} \text{ is satisfiable,}\\
C \sqsubseteq D &\text{ iff } t_x(C) \Rightarrow t_x(D) \text{ is valid}\\
C \sqsubseteq_{\mathcal{T}} D &\text{ iff } \Phi_t \Rightarrow \forall x.(t_x(C) \Rightarrow t_x(D)) \text{ is valid.}
\end{aligned}$$

$\leadsto$ $\mathcal{ALC}$ is a fragment of FOL with 2 variables (L2), known to be decidable

$\leadsto$ $\mathcal{ALC}$ with inverse roles and Boolean operators on roles is a fragment of L2

$\leadsto$ further adding number restrictions yields a fragment of C2 (L2 with "counting quantifiers"), known to be decidable

- in contrast to most DLs, adding transitive roles/transitive closure operator to L2 leads to **undecidability**
- many DLs (like many modal logics) are fragments of the **Guarded Fragment**
- most DLs are less complex than L2: L2 is NExpTime-complete, most DLs are in ExpTime

DLs and Modal Logics are closely related:

$\mathcal{ALC} \rightleftarrows$ multi-modal $\mathbf{K}$:

$$C \sqcap D \rightleftarrows C \land D, \qquad C \sqcup D \rightleftarrows C \lor D$$
$$\neg C \rightleftarrows \neg C \quad ,$$
$$\exists R.C \rightleftarrows \langle R \rangle C \quad , \qquad \forall R.C \rightleftarrows [R]C$$

transitive roles $\stackrel{\bullet}{\rightleftarrows}$ transitive frames (e.g., in $\mathbf{K4}$)

regular expressions on roles $\stackrel{\bullet}{\rightleftarrows}$ regular expressions on programs (e.g., in PDL)

inverse roles $\stackrel{\bullet}{\rightleftarrows}$ converse programs (e.g., in C-PDL)

number restrictions $\stackrel{\bullet}{\rightleftarrows}$ deterministic programs (e.g., in D-PDL)

➪ no TBoxes available in modal logics
  $\leadsto$ "internalise" axioms using a universal role u: $C \doteq D \rightleftarrows [u](C \Leftrightarrow D)$

➪ no ABox available in modal logics $\leadsto$ use nominals