Expressivity and Complexity of Description Logics with Concrete Domains

Carsten Lutz

TU Dresden, Germany

Description Logics

- Description Logics (DLs) originated in Artificial Intelligence as a tool for the representation of conceptual knowledge.

  e.g. "a father is a human which is male and has at least one child"

- Technically, DLs are very closely related to modal logics

  Today lively exchange between these two fields

- Strong emphasis on reasoning: DLs should be decidable,
  preferably of low complexity, implementable

- Many interesting applications: KR, semantic web, databases,...

Description Logics II

Knowledge represented in terms of concepts: ($\approx$ formulas)

- Human $\sqcap$ Male $\sqcap$ $\exists$child.Human
- Workshop $\sqcap$ $\forall$papers.$\exists$has-topic.(ML $\sqcup$ DL)

Most important reasoning tasks:

- concept satisfiability

  $\Longrightarrow$ check for modelling mistakes

- subsumption

  is each instance of a concept C also an instance of a concept D?

  ($\approx$ validity of implications $C \to D$)

  $\Longrightarrow$ arrange concepts in a hierarchy w.r.t. generality

Subsumption can easily be reduced to (un)satisfiability.

Motivating Concrete Domains

Many applications need to define concepts with reference to

concrete qualities of real-world objects

such as sizes, heights, and temporal and spatial extensions

Examples:

- People that are 2 meters tall
- People which are taller than their parents
- Countries whose territory is crossed by the equator

$\Longrightarrow$ Appropriate extension of “classical” DLs needed!

Nowadays generally accepted approach: Concrete Domains [BaaderHanschke91].

The Description Logic $\mathcal{ALC}$

$\mathcal{ALC}$ is the smallest propositionally closed Description Logic

Atomic types:	concept names $A, B, \ldots$	(unary predicates)
	role names $R, S, \ldots$	(binary relations)
Constructors:	- $\neg C$	(negation)
	- $C \sqcap D$	(conjunction)
	- $C \sqcup D$	(disjunction)
	- $\exists R.C$	(existential restriction)
	- $\forall R.C$	(universal restriction)

Example concept: Human $\sqcap$ Male $\sqcap$ $\exists$child.Human

The Description Logic $\mathcal{ALC}$

$\mathcal{ALC}$ is the smallest propositionally closed Description Logic

and a notational variant of multi-modal K.

Atomic types: concept names $A, B, \ldots$ $\approx$ propositional variables

role names $R, S, \ldots$ $\approx$ names for accessibility relations

Constructors:
- $\neg C$
- $C \sqcap D$
- $C \sqcup D$
- $\exists R.C$ $\approx$ modal diamond $\Diamond_R \varphi$
- $\forall R.C$ $\approx$ modal box $\Box_R \varphi$

Example concept: Human $\sqcap$ Male $\sqcap$ $\exists$child.Human

$\approx$ Human $\wedge$ Male $\wedge$ $\Diamond_{\text{child}}$Human

$\mathcal{ALC}$ Semantics

Semantics based on interpretations $(\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\cdot^\mathcal{I}$ maps

- each concept name A to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$.
- each role name R to a binary relation $R^\mathcal{I}$ over $\Delta^\mathcal{I}$.

$\approx$ Kripke structure with set of worlds $\Delta^\mathcal{I}$

Semantics of complex concepts:

$$\begin{aligned}
(\neg C)^\mathcal{I} &= \Delta^\mathcal{I} \setminus C^\mathcal{I} \\
(C \sqcap D)^\mathcal{I} &= C^\mathcal{I} \cap D^\mathcal{I} \\
(C \sqcup D)^\mathcal{I} &= C^\mathcal{I} \cup D^\mathcal{I} \\
(\exists R.C)^\mathcal{I} &= \{d \in \Delta^\mathcal{I} \mid \exists e : (d, e) \in R^\mathcal{I} \text{ and } e \in C^\mathcal{I}\} \\
(\forall R.C)^\mathcal{I} &= \{d \in \Delta^\mathcal{I} \mid \forall e : (d, e) \in R^\mathcal{I} \text{ implies } e \in C^\mathcal{I}\}
\end{aligned}$$

Concrete Domains

A concrete domain $\mathcal{D} = (\Delta_\mathcal{D}, \Phi_\mathcal{D})$ consists of

- a set $\Delta_\mathcal{D}$ and
- a set $\Phi_\mathcal{D}$ of predicate names; each $P \in \Phi_\mathcal{D}$ is equipped with an arity n a fixed extension $P^\mathcal{D} \subseteq \Delta_\mathcal{D}^n$.

Examples:

1. the natural numbers $\mathbb{N}$ and predicates $=_n$, $<$, $>$, $=$
2. the real numbers $\mathbb{R}$ and predicates $<$, $>$, $=$, $+$, $*$
3. the subsets of $\mathbb{R}^2$ and "spatial" predicates $=_{\text{polygon}}$, overlaps, etc.

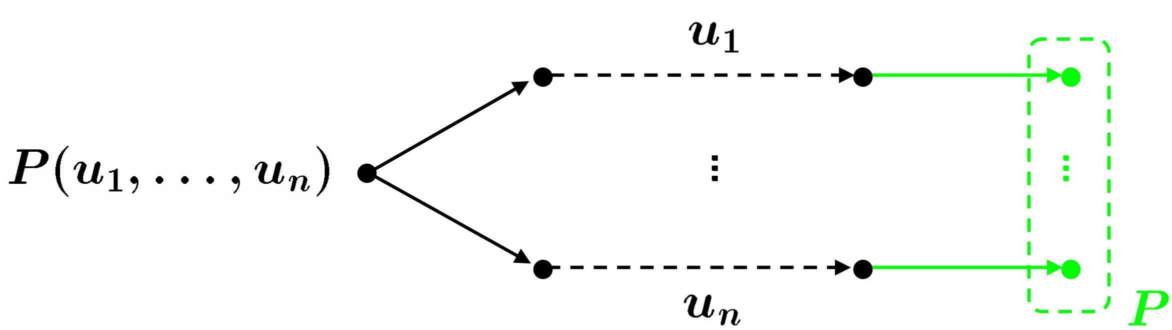
The Description Logic $\mathcal{ALC}(\mathcal{D})$

New atomic types:

- abstract features f are functional roles
- concrete features g are interpreted as partial functions $g^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow \Delta_{\mathcal{D}}$

Path: sequence of features $u = f_1 \cdots f_n g$ with $f_1, \ldots, f_n$ abstract and g concrete

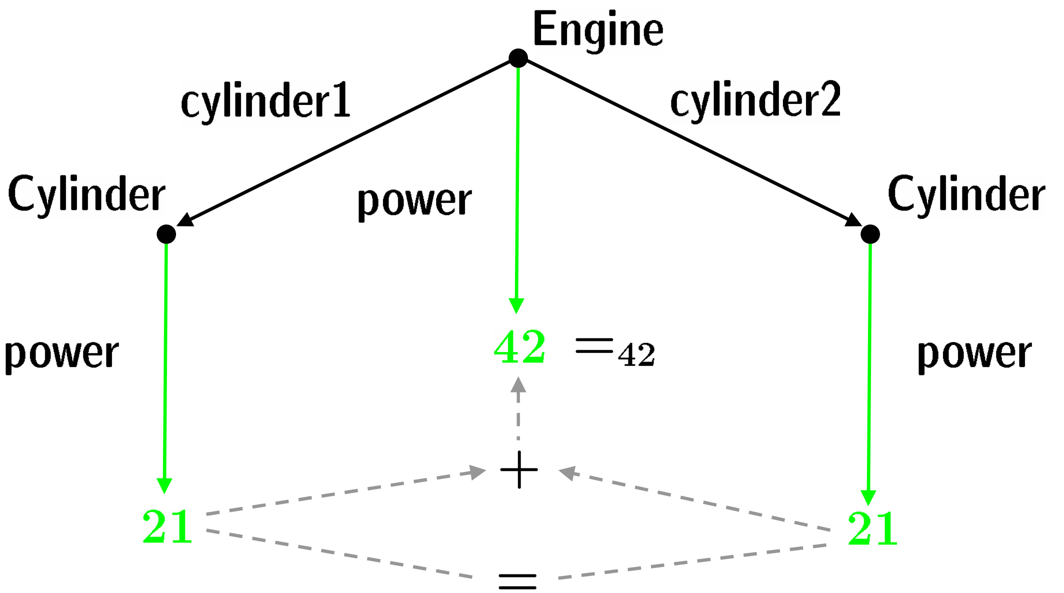
New concept constructor:



$\mathcal{ALC}(\mathcal{D})$: extension of $\mathcal{ALC}$ with concrete domain $\mathcal{D}$

An Example $\mathcal{ALC}(\mathcal{D})$ Concept

$$\mathsf{Engine} \sqcap \exists \mathsf{cylinder1}.\mathsf{Cylinder} \sqcap \exists \mathsf{cylinder2}.\mathsf{Cylinder}$$
$$\sqcap =_{42}(\mathsf{power})$$
$$\sqcap =((\mathsf{cylinder1} \circ \mathsf{power}), (\mathsf{cylinder2} \circ \mathsf{power}))$$
$$\sqcap +((\mathsf{cylinder1} \circ \mathsf{power}), (\mathsf{cylinder2} \circ \mathsf{power}), \mathsf{power})$$



Expressivity and Complexity of DLs

DL research goal:

> Identify DLs that are sufficiently expressive for a given application and for which reasoning is "practicable".

"practicable": implemented algorithms perform sufficiently well on realistic problems.

$\Longrightarrow$ Reasoning should be decidable! But what complexity is ok?

CLASSIC 1988 — $\mathcal{KRIS}$ 1991 — FaCT 1997 — . . . ?

PTime — PSpace ($\mathcal{ALC}$) — ExpTime

Tradeoff between expressivity and computational complexity!

Reasoning with $\mathcal{ALC}(\mathcal{D})$

Decidability/complexity results should be general.

Theorem [BaaderHanschke91]. $\mathcal{ALC}(\mathcal{D})$-concept satisfiability is decidable if $\mathcal{D}$-satisfiability is decidable.

$\mathcal{D}$-satisfiability: satisfiability of finite conjunctions

$$P_1(x_1^{(1)}, \dots, x_{n_1}^{(1)}) \wedge \cdots \wedge P_k(x_1^{(k)}, \dots, x_{n_k}^{(k)})$$

$\Longrightarrow$ interface between DL and concrete domain

Theorem [L__99]. $\mathcal{ALC}(\mathcal{D})$-concept satisfiability is PSPACE-complete if $\mathcal{D}$-satisfiability is in PSPACE.

$\Longrightarrow$ Reasoning with $\mathcal{ALC}(\mathcal{D})$ is not harder than reasoning with $\mathcal{ALC}$.

$\mathcal{ALC}(\mathcal{D})$ with Numerical Concrete Domains

All concrete domains based on the rational numbers:

$=_q$	int	$<, \leq, =, \ldots$	$+$	$*$	$\mathcal{D}$-sat.	$\mathcal{ALC}(\mathcal{D})$ sat.
✗	✗	✗	✗	✗	undecidable	undecidable
✗		✗	✗	✗	ExpTime-c.	in NExpTime
✗	✗	✗	✗		NP-c.	PSpace-c.
✗		✗	✗		PTime	PSpace-c.

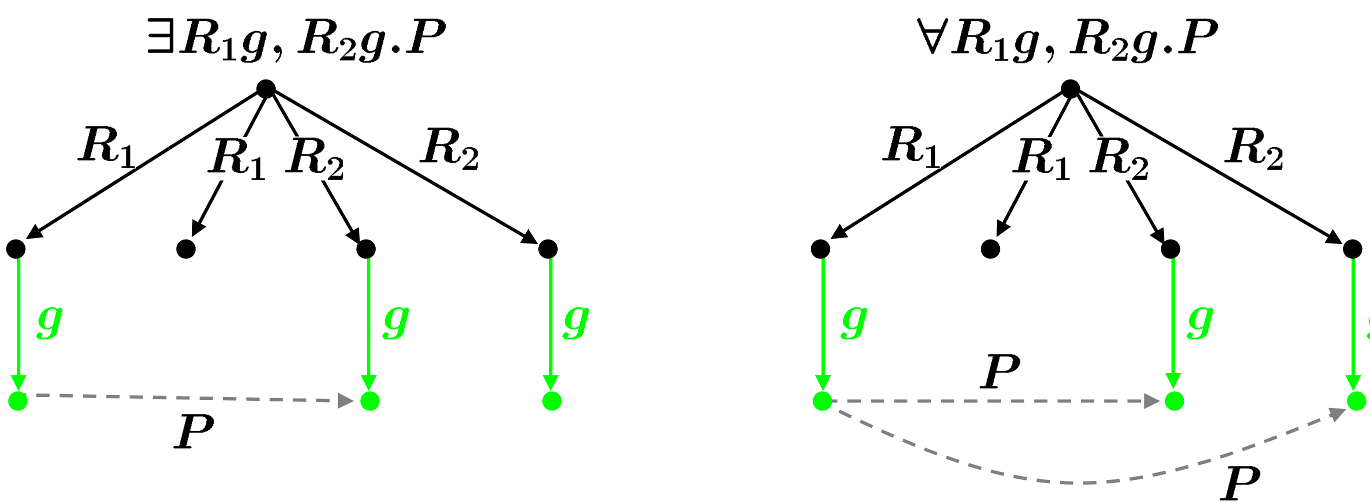
Why Functionality?

Why restrict the concrete domain constructor to functional roles?

"A man that is older than all of his neighbors"

Alternative: admit operators

$\exists U_1, \ldots, U_n.P$ and $\forall U_1, \ldots, U_n.P$ with U_i of the form $R_1 \cdots R_k g$.



Resulting logic: $\mathcal{ALCP}(\mathcal{D})$

Two Problems of $\mathcal{ALCP}(\mathcal{D})$ I

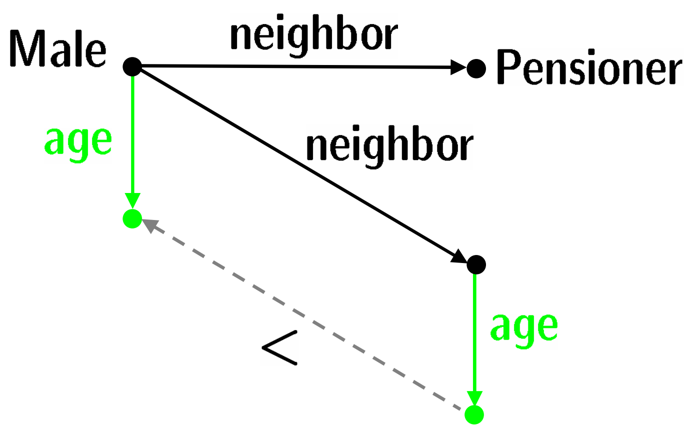
Expressive power of $\mathcal{ALCP}(\mathcal{D})$ not satisfactory:

- "A man that is older than all of his neighbors"

$$\mathsf{Male} \sqcap \forall \mathsf{neighbor} \circ \mathsf{age}, \mathsf{age}.<$$

- "A man that has a neighbor who is a pensioner and younger than him."

$$\mathsf{Male} \sqcap \exists \mathsf{neighbor}.\mathsf{Pensioner} \sqcap \exists \mathsf{neighbor} \circ \mathsf{age}, \mathsf{age}.<$$



Possible solution: role hierarchies

Two Problems of $\mathcal{ALCP}(\mathcal{D})$ Ib

Role hierarchy (RH): finite set of role inclusions $R \sqsubseteq S$

Interpretation $\mathcal{I}$ is a model of RH $\mathcal{H}$ if

$$R^{\mathcal{I}} \subseteq S^{\mathcal{I}} \text{ for all } R \sqsubseteq S \in \mathcal{H}.$$

Concept satisfiability w.r.t. role hierarchies:

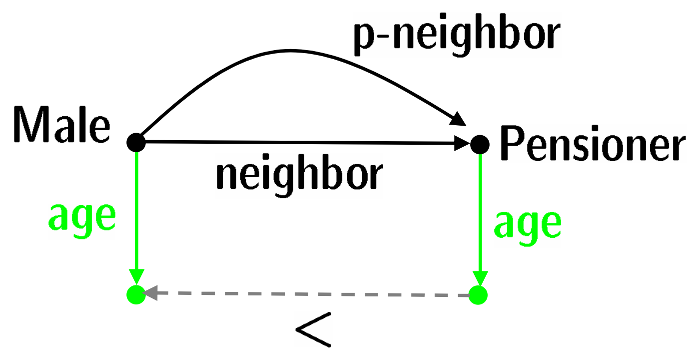
Do C and $\mathcal{H}$ have a common model?

"A man that has a neighbor who is a pensioner and younger than him."

Male $\sqcap$ $\exists$p-neighor.Pensioner $\sqcap$ $\exists$p-neighbor $\circ$ age, age.$<$

RH: {p-neighbor $\sqsubseteq$ neighbor}

functional



Two Problems of $\mathcal{ALCP}(\mathcal{D})$ II

Theorem [L__02]. There exists a concrete domain $\mathcal{D}$ such that $\mathcal{D}$-satisfiability is in $\textsc{PTime}$ and $\mathcal{ALCP}(\mathcal{D})$-concept satisfiability is $\textsc{NExpTime}$-hard.

Proof by reduction of a $\textsc{NExpTime}$-complete variant of the Post Correspondence Problem (PCP).

Matching upper bound if $\mathcal{D}$-satisfiability is in NP [L__02].

$\textsc{NExpTime}$-hardness also arises with "natural" concrete domains:

- $\Delta_{\mathcal{D}}$ contains the natural numbers
- there is a unary predicate $=_0$
- there is a binary predicate $=$
- there are ternary predicates "$+$" and "$*$"

$\Longrightarrow$ arithmetic concrete domains

Concrete Functionality

Do concrete features have to be functional?

"My telephone numbers are 0351/46339171 and 0160/98386681"

- Just replacing concrete features by concrete roles is not convincing:

  People have only one age, weight, size, etc.

- General approach: allow number restrictions on concrete roles [HorrocksPan02]

  - $(\leq n\, F)$ $(\geq n\, F)$ F: concrete role
  - $(\leq n\, F_1, \ldots, F_k\, .\, P)$ $(\geq n\, F_1, \ldots, F_k\, .\, P)$

Decidability and complexity:

Resulting logic "behaves" like $\mathcal{ALC}(\mathcal{D})$ / $\mathcal{ALCP}(\mathcal{D})$.

Back to the roots!

Let's go back to the basic formalism:

- $\mathcal{ALC}(\mathcal{D})$ with basic concrete domain constructor $P(u_1, \ldots, u_n)$
- only abstract and concrete features (both functional!) in concrete domain constructor
- no role hierachies

Next important step: extend our logic with TBoxes

TBoxes

TBoxes are used to capture background knowledge:

$$\mathsf{Father} \sqsubseteq \mathsf{Human} \sqcap \mathsf{Male} \sqcap \exists \mathsf{child}.\mathsf{Human}$$

$$\mathsf{Human} \sqcap \neg\exists \mathsf{parent}.\mathsf{Human} \sqsubseteq \mathsf{Orphan} \sqcup \mathsf{Adam} \sqcup \mathsf{Eve}$$

TBox: finite set of concept implications $C \sqsubseteq D$

Interpretation $\mathcal{I}$ is a model of TBox $\mathcal{T}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all $C \sqsubseteq D \in \mathcal{T}$.

Concept satisfiability w.r.t. TBoxes:

Do C and $\mathcal{T}$ have a common model?

Closely related to the universal modality:

$$C \text{ sat w.r.t. } \mathcal{T} \quad \text{iff} \quad C \wedge \bigwedge_{D \sqsubseteq E \in \mathcal{T}} \Box_u (D \rightarrow E) \text{ sat}$$

Reasoning with TBoxes

Theorem [BaaderHanschke92/L__01] $\mathcal{ALC}(\mathcal{D})$-concept satisfiability w.r.t. TBoxes is undecidable if $\mathcal{D}$ satisfies the following:

- $\Delta_{\mathcal{D}}$ contains the natural numbers
- there is a unary predicate $=_0$
- there are binary predicates $=$ and $+1$

Proof by reduction of the unconstrained PCP.

There are (at least) three ways around this problem:

1. use acyclic TBoxes
2. disallow paths in concrete domain constructor
3. carefully choose the concrete domain

Acyclic TBoxes

A TBox $\mathcal{T}$ is called acyclic if

- in each concept implication $C \sqsubseteq D$, C is a concept name
- it contains no cycles:

$$\{A_0 \sqsubseteq A_1 \sqcap C$$
$$A_1 \sqsubseteq \exists R.A_2$$
$$A_2 \sqsubseteq A_0\}$$

Acyclic TBoxes can be unfolded:

Father $\sqsubseteq$ Human $\sqcap$ Male $\sqcap$ $\exists$child.Human

Male $\sqsubseteq$ $\neg$Female

Female $\sqcap$ $\exists$married-to.(Father $\sqcap$ Human $\sqcap$ Male $\sqcap$ $\neg$Female $\sqcap$ $\exists$child.Human)

Theorem. $\mathcal{ALC}(\mathcal{D})$-concept satisfiability w.r.t. acyclic TBoxes is decidable if $\mathcal{D}$-satisfiability is decidable.

Acyclic TBoxes: Complexity

Acyclic TBoxes are usually believed to be "harmless" w.r.t. complexity:

$\mathcal{ALC}$-concept satisfiability w.r.t. acyclic TBoxes is $\textsc{PSpace}$ complete.

But this is not true:

Theorem [L__01]. For every arithmetic concrete domain $\mathcal{D}$, $\mathcal{ALC}(\mathcal{D})$-concept satisfiability w.r.t. acyclic TBoxes is $\textsc{NExpTime}$-hard.

Proof: reduction of a $\textsc{NExpTime}$-complete variant of the PCP.

Matching upper bound if $\mathcal{D}$-satisfiability is in NP [L__01].

Note: Although acyclic TBoxes don't really contribute to the expressive power, they yield succinctness

Path-freeness

A pragmatic approach for reasoning with full TBoxes:

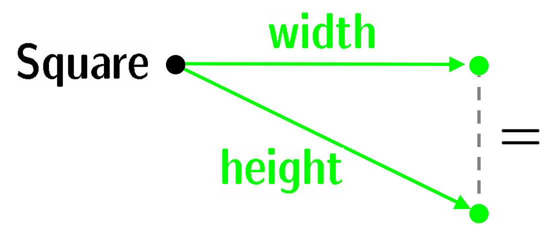
disallow paths of length >1 in concrete domain constructor.

Approach has been taken in the **RACER** system.

Theorem [HaarslevMöller01/L__02]. Path-free $\mathcal{ALC}(\mathcal{D})$-concept satisfiability w.r.t. general TBoxes is ExpTime-complete if $\mathcal{D}$-satisfiability is in ExpTime.

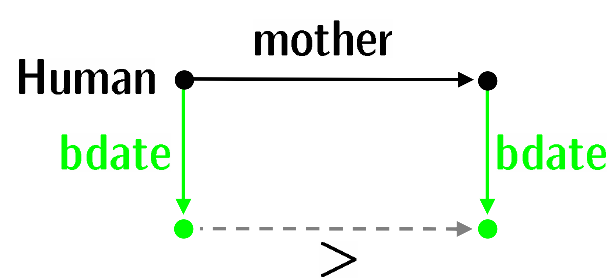
- "A square has same width and height."

$$\text{Square} \sqsubseteq =(\text{width}, \text{height})$$



- "Humans are born after their mothers"

$$\text{Human} \sqsubseteq >(\text{bdate}, (\text{mother} \circ \text{bdate}))$$



Expressivity of Path-freeness

Path free concrete domains can be "simulated" by concept names:

- let input concept C and TBox $\mathcal{T}$ be given
- replace each concept $P(g_1, \ldots, g_k)$ with a concept name $A_{P(g_1,\ldots,g_k)}$
- call a set Φ of "relevant" concepts $P(g_1, \ldots, g_k)$ inconsistent if

$$\bigwedge_{P(g_1,\ldots,g_k)\in\Phi} P(x_{g_1}, \ldots, x_{g_k}) \quad \text{is unsatisfiable}$$

- for each inconsistent Φ, add the concept equation

$$\bot \doteq \bigsqcap_{P(g_1,\ldots,g_k)\in\Phi} A_{P(g_1,\ldots,g_k)}$$

$\Longrightarrow$ In some sense, path-free concrete domains do not extend expressive power

Reasoning with Full TBoxes

What if we really need more expressive power?

Consider a "weak" concrete domain **C**:

- $\Delta_{\mathsf{C}} = \mathbb{Q}$
- unary predicate $=_q$ for each $q \in \mathbb{Q}$
- binary comparison predicates $<, \leq, =, \neq, \geq, >$

No arithmetics!

Theorem [L__01]. $\mathcal{ALC}(\mathsf{C})$-concept satisfiability w.r.t. TBoxes is ExpTime-complete.

Proof via automata-theoretic approach.

Can be (simultaneously) extended with

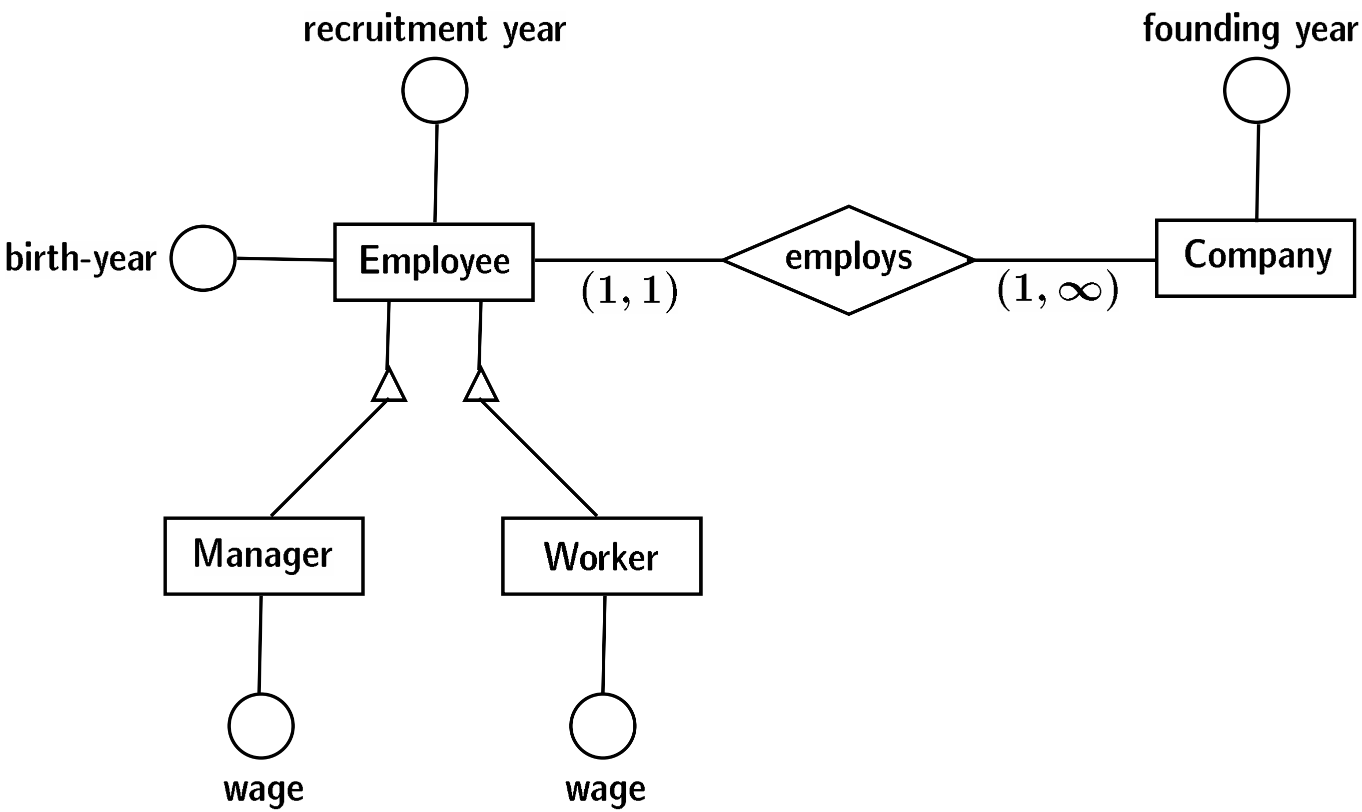
- qualifying number restricions ($\approx$ graded modalities)
- inverse roles, transitive roles, and role hierarchies

and the result $\mathbb{Q}$-$\mathcal{SHIQ}$ is still in ExpTime [L__02].

Reasoning with Conceptual Database Models

Entity Relationship (ER) diagrams are predominant formalism for conceptual database modelling

Description Logics can be used for reasoning about them (e.g. consistency checking)

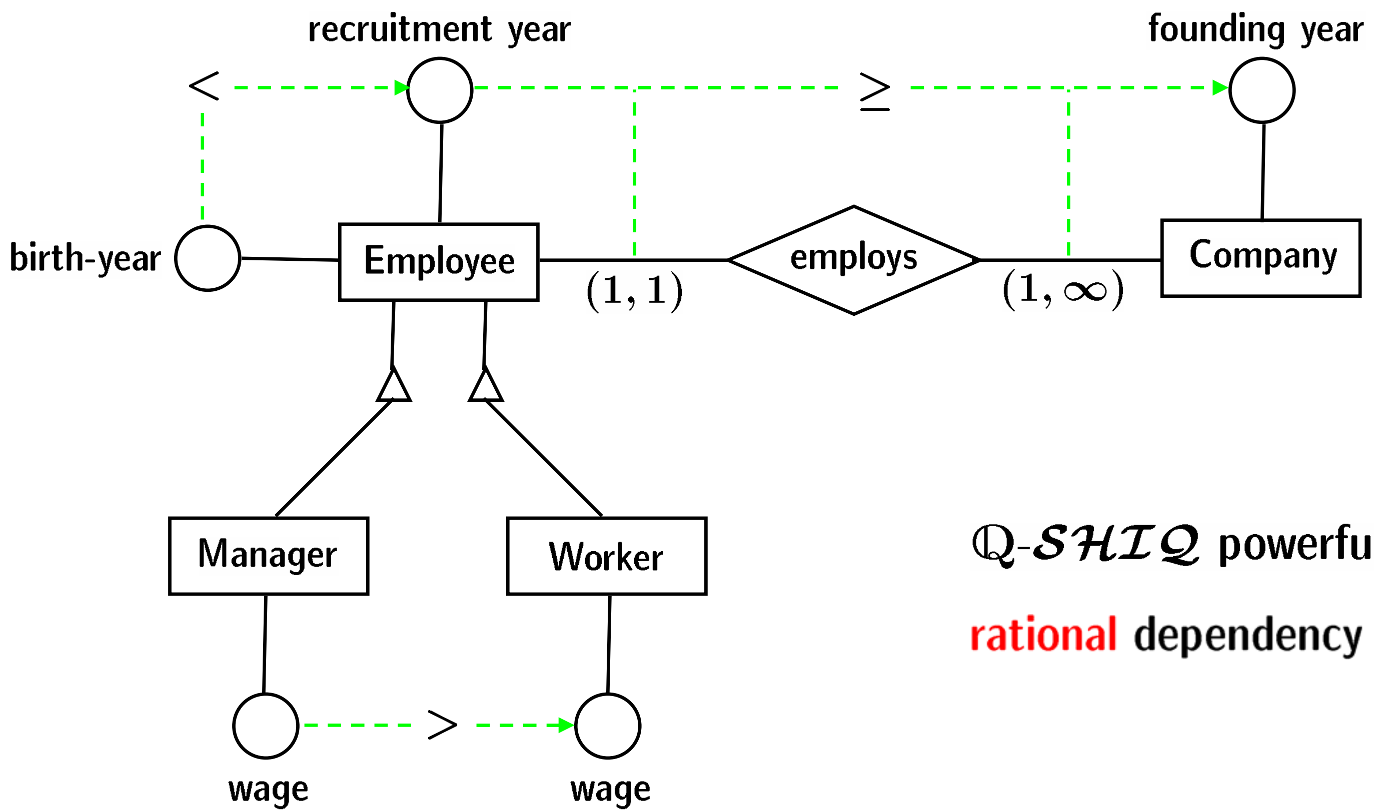


Numerical Attribute Dependencies

$$\text{Company} \sqsubseteq \forall \text{employs}.\text{Employee} \sqcap (\geq 1\ \text{employs})$$

$$\text{Employee} \sqsubseteq \forall \text{employs}^{-}.\text{Company} \sqcap (= 1\ \text{employs}^{-})$$

$$\text{Manager} \sqsubseteq \text{Employee}$$

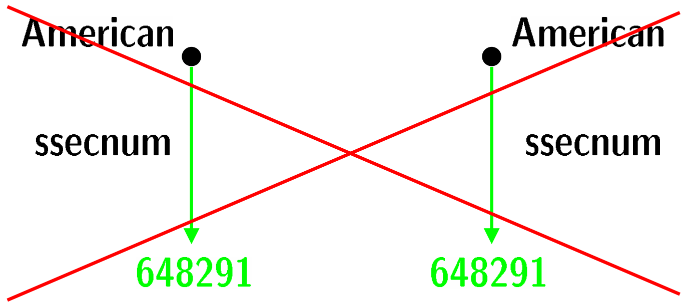


$\mathbb{Q}$-$\mathcal{SHIQ}$ powerful enough for

rational dependency constaints.

Concrete Features as Keys

"Americans are uniquely identified by their social security number"



- Add keybox, i.e., sets of expressions

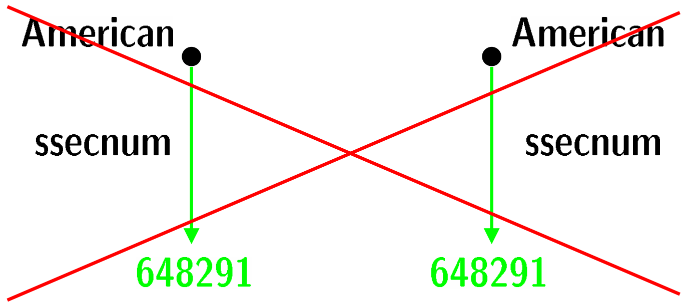
$$u_1, \ldots, u_n \text{ keyfor } C$$

- For example ssecnum keyfor American.
- Closely related to nominals: $=_0(g)$ is nominal for keybox "g keyfor $\top$"
- Very useful in database applications

Theorem [Areces et al.02] $\mathcal{ALC}(\mathcal{D})$-concept satisfiability w.r.t. keyboxes is undecidable for some concrete domains $\mathcal{D}$.

Concrete Features as Keys

"Americans are uniquely identified by their social security number"



- Add keybox, i.e., sets of expressions

$$u_1, \ldots, u_n \text{ keyfor } C$$

- For example ssecnum keyfor American.

A keybox is called Boolean if only Boolean combinations of concept names occur inside key definitions.

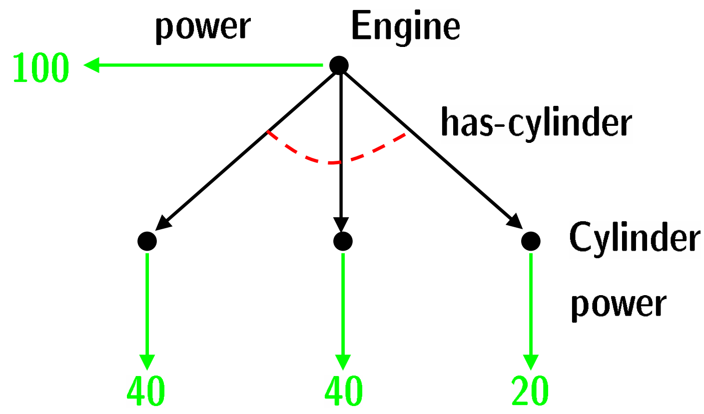
Theorem [Areces et al.02] $\mathcal{ALC}(\mathcal{D})$-concept satisfiability w.r.t. Boolean keyboxes is $\textsc{NExpTime}$-complete if $\mathcal{D}$-satisfiability is in NP.

Adding Aggregation

"An engine whose power is the sum of the powers of its individual cylinders."

Allow aggregation expressions inside the concrete domain constructor:

$$=(\text{power}, \underbrace{\text{SUM}}_{\text{aggregation function}}(\text{has-cylinder} \circ \text{power}))$$



Other aggregation functions: min, max, count, average, etc.

Theorem [BaaderSattler98]. $\mathcal{ALC}(\mathcal{D}^{agg})$-concept satisfiability is decidable for the aggregation functions min and max.

But: Most interesting cases are undecidable!

For example $\mathcal{ALC}(\mathcal{D}^{agg})$ with min, max, sum [BaaderSattler98].

Conclusion

Concrete domains

- allow to integrate "concrete qualities" into DLs
- useful in many application areas
- "typically Description Logic"

Not all concrete domains are numerical:

- temporal concrete domains based on Allen's interval algebra
- spatial concrete domains based on the RCC-8 relations

There is interesting interaction with other operators as well.

(nominals, inverse roles, etc.)