Answering Queries in Description Logics: Theory and Applications to Data Management

Diego Calvanese[1], Michael Zakharyaschev[2]

[1] KRDB Research Centre
Free University of Bozen-Bolzano

[2] Birbeck College, London

ESSLLI 2010, August 14–20, 2010
Copenhagen, Denmark

Overview of the Course

1. Introduction and background
    1. Ontology-based data management
    2. Brief introduction to computational complexity
    3. Query answering in databases
    4. Querying databases and ontologies
2. Lightweight description logics
    5. Introduction to description logics
    6. DLs for conceptual data modeling: the *DL-Lite* family
    7. The $\mathcal{EL}$ family of tractable description logics
3. Query answering in the *DL-Lite* family
    8. Query answering in description logics
    9. Lower bounds for more expressive description logics
    10. Query answering by rewriting
4. The combined approach to query answering
    11. Query answering in *DL-Lite*: data completion
    12. Query rewriting in $\mathcal{EL}$
5. Linking ontologies to relational data
    13. The impedance mismatch problem
    14. Query answering in Ontology-Based Data Access systems
6. Conclusions and references

Lecture 2:

‘Lightweight’ description logics:

DL-Lite and $\mathcal{EL}$

(A quick introduction to Description Logic,

focusing on tractable *DL-Lite* and $\mathcal{EL}$ logics)

Recommended reading

DL-Lite <u>available on the web</u>

(1) A. Artale, D. Calvanese, R. Kontchakov and M. Zakharyaschev.
The DL-Lite family and relations. JAIR, 36:1–69, 2009.

(2) D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati.
DL-Lite: Tractable description logics for ontologies. Proceedings of AAAI 2005.

(3) D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati.
Tractable reasoning and efficient query answering in DLs: The DL-Lite family.
Journal of Automated Reasoning, 39:385–429, 2007.

$\mathcal{EL}$

(4) F. Baader, S. Brandt, C. Lutz. *Pushing the $\mathcal{EL}$ envelope.* IJCAI 2005.

(5) F. Baader, S. Brandt, C. Lutz. *Pushing the $\mathcal{EL}$ envelope further.* OWLED 2008.

(6) C. Lutz, R. Piro, F. Wolter. *Enriching $\mathcal{EL}$-concepts with greatest fixpoints*. ECAI 2010.

Acknowledgements: Roman Kontchakov, Carsten Lutz, Frank Wolter

Description Logic

http://en.wikipedia.org/wiki/Description_logic

DL is a (large) family of knowledge representation & reasoning formalisms

- more expressive than propositional logic
- less expressive than first-order logic
  (≈ decidable modal logics, hybrid logics)
- developed by KR community for applications in AI

Application-driven equilibrium: expressiveness vs. computational costs

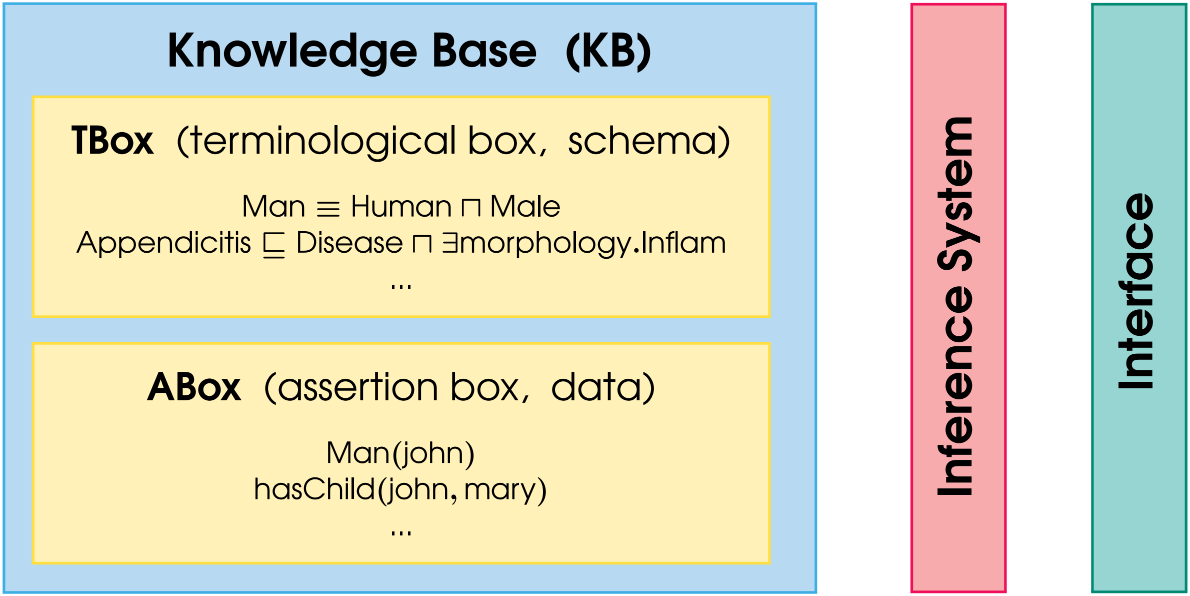
Applications:

- Ontologies (or terminologies) in medicine, bioinformatics, ...
- Semantic Web
- Ontology-based data access

Web Ontology Language (OWL) W3C standards OWL 1 (2004), OWL 2 (2009)

OWL = DL + XML

DL architecture



Description logic constructs

- **Alphabet:**
  - concept names $A_0, A_1, \ldots$ (e.g., Person, Female, ...)
  - role names $R_0, R_1, \ldots$ (e.g., hasChild, loves, ...)
  - individual names $a_0, a_1, \ldots$ (e.g., john, mary, ...)
  - concept constructs: $\top$, $\sqcap$, $\neg$, $\exists$, $\forall$, $\geq q$, ... (e.g., Person $\sqcap$ Female)
  - role constructs: R^-, $R \circ S$, ... (e.g., isChildOf)
  - axiom construct: $\sqsubseteq$ (e.g., Man $\sqsubseteq$ Person)

- **Concepts:**
  - concept names
  - $\top$, $\bot$, $\neg C$, $C \sqcap D$, $\forall R.C$, $\exists R.C$, $\geq q R.C$,
    where C, D are concepts and R a role

Examples: Person $\sqcap$ Female, Person $\sqcap$ $\neg$Female,
Person $\sqcap$ $\exists$hasChild.$\top$, Person $\sqcap$ $\forall$hasChild.Male

Description logic semantics

- (standard Tarski-style) **interpretation** is a structure $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
  - $\Delta^{\mathcal{I}}$ is the **domain** of $\mathcal{I}$ (a non-empty set)
  - $\cdot^{\mathcal{I}}$ is an **interpretation function** that maps:
    - concept name $A_i \mapsto$ subset $A_i^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ $\quad (A_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}})$
    - role name $R_i \mapsto$ binary relation $R_i^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$ $\quad (R_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})$
    - individual name $a_i \mapsto$ element $a_i^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ $\quad (a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}})$

- interpretation of **complex concepts** in $\mathcal{I}$:
  - $(\top)^{\mathcal{I}} = \Delta^{\mathcal{I}}$ and $(\bot)^{\mathcal{I}} = \emptyset$
  - $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
  - $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
  - $(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}} \, \big((x,y) \in R^{\mathcal{I}} \to y \in C^{\mathcal{I}}\big)\}$
  - $(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in C^{\mathcal{I}} \, (x,y) \in R^{\mathcal{I}}\}$
  - $(\geq q R.C)^{\mathcal{I}} = \big\{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \in C^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\} \geq q\big\}$

TBoxes

statements about **how concepts and roles are related to each other**

A TBox $\mathcal{T}$ is a finite set of **terminological axioms**:

- $C \sqsubseteq D$ — C is subsumed by D (**concept inclusion**)
- $R \sqsubseteq S$ — R is a subrole of S (**role inclusion**)

an interpretation $\mathcal{I}$ **satisfies** an axiom

- $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- $\mathcal{I} \models R \sqsubseteq S$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$

An interpretation $\mathcal{I}$ is a **model** of $\mathcal{T}$ if $\mathcal{I}$ satisfies **every axiom** of $\mathcal{T}$

ABoxes

assert knowledge about **individuals**

An ABox $\mathcal{A}$ is a finite set of **assertional axioms**

- $C(a)$ concept assertion for an individual
- $R(a,b)$ role assertion for a pair of individuals

an interpretation $\mathcal{I}$ **satisfies** an assertion
- $\mathcal{I} \models C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- $\mathcal{I} \models R(a,b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

An interpretation $\mathcal{I}$ is a **model** of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ satisfies **every axiom** of $\mathcal{T}$ and $\mathcal{A}$

OWL ontology example

- **Protégé 4.0** a free, open source ontology editor

  `http://protege.stanford.edu/`

  where you can also find a library of ontologies

  (tutorials explaining how to use Protégé are at

  `http://www.co-ode.org/resources/tutorials/`)

- built-in ontology reasoners **FaCT++**, **Pellet** or **HermiT**

`http://owl.man.ac.uk/factplusplus/` `http://pellet.owldl.com/`

`http://hermit-reasoner.com/`

Reasoning problems

Concept satisfiability: given $\mathcal{T}$ and a concept C, decide whether there is $\mathcal{I} \models \mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$

Subsumption: given $\mathcal{T}$ and concepts C, D, decide whether $\mathcal{T} \models C \sqsubseteq D$
i.e., $\forall \mathcal{I}\ (\mathcal{I} \models \mathcal{T} \ \rightarrow\ \mathcal{I} \models C \sqsubseteq D)$

Instance checking: given $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, C and an individual a from $\mathcal{A}$, decide whether $\mathcal{K} \models C(a)$

Exercise: show that these three problems are reducible to each other

Conjunctive query answering: given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a CQ $q(\vec{x})$ and a tuple $\vec{a}$ of individual names from $\mathcal{A}$, decide whether $\mathcal{K} \models q(\vec{a})$

Query answering is typically a **harder** problem than the other three

First-order translation

$$
\begin{array}{lcl}
\boldsymbol{A} & \rightsquigarrow & \boldsymbol{A(x)} \\
\neg \boldsymbol{C} & \rightsquigarrow & \neg \boldsymbol{C(x)} \\
\boldsymbol{C} \sqcap \boldsymbol{D} & \rightsquigarrow & \boldsymbol{C(x)} \wedge \boldsymbol{D(x)} \\
\forall \boldsymbol{R.C} & \rightsquigarrow & \forall \boldsymbol{y} \, \big(\boldsymbol{R(x,y)} \to \boldsymbol{C(y)}\big) \\
\exists \boldsymbol{R.C} & \rightsquigarrow & \exists \boldsymbol{y} \, \big(\boldsymbol{R(x,y)} \wedge \boldsymbol{C(y)}\big) \\
\geq \boldsymbol{qR.C} & \rightsquigarrow & \exists \boldsymbol{y_1, \dots, y_q} \bigwedge_{i<j} \big(\boldsymbol{y_i} \neq \boldsymbol{y_j} \wedge \boldsymbol{R(x, y_i)} \wedge \boldsymbol{C(y_i)}\big) \\
\boldsymbol{C} \sqsubseteq \boldsymbol{D} & \rightsquigarrow & \forall \boldsymbol{x} \, \big(\boldsymbol{C(x)} \to \boldsymbol{D(x)}\big)
\end{array}
$$

DL is embeddable into the 2-variable guarded fragment of first-order logic

(full FOL is undecidable; this guarded fragment is NExpTime-complete)

Unique name assumption (UNA)

An interpretation $\mathcal{I}$ is a **model** of a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ **under the UNA** if $\mathcal{I} \models \mathcal{K}$ and $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$, for any distinct object names a_i and a_j **occurring in** $\mathcal{A}$

OWL: a more flexible approach

- UNA is **dropped** (so no restrictions on interpretations of object names)
- User is provided with the constructs $=$ (**sameAs**) and $\neq$ (**differentFrom**) to explicitly impose constraints on individual names
- UNA is expressible: add $a_i \neq a_j$ to $\mathcal{A}$, for all distinct a_i and a_j in $\mathcal{A}$

Price of $=$ Have to check whether $a = b$ in $\mathcal{A}$ under given equality constraints

Equivalent to reachability in undirected graphs, which is

LOGSPACE-complete

(Reingold 2008)

…just peanuts for most DLs, but not for *DL-Lite* & OWL 2 QL…

The history of description logic so far

. . . – mid 1990s: efficient reasoning cannot afford full Booleans

sub-Boolean DLs with $\sqcap$ and $\forall$ are enough

$\mathcal{FL}$, $\mathcal{AL}$, . . . combined complexity $\leq$ NP

mid 1990s – 2005 'efficient' reasoning possible for **ExpTime** DLs (FaCT,...)

full Booleans and other constructs

$\mathcal{SHIQ}$, $\mathcal{SHOIN}$ ($\approx$ OWL 1), $\mathcal{SROIQ}$ ($\approx$ OWL 2) $\geq$ ExpTime

mid 2005 – . . . new challenges: answering queries & HUGE ontologies

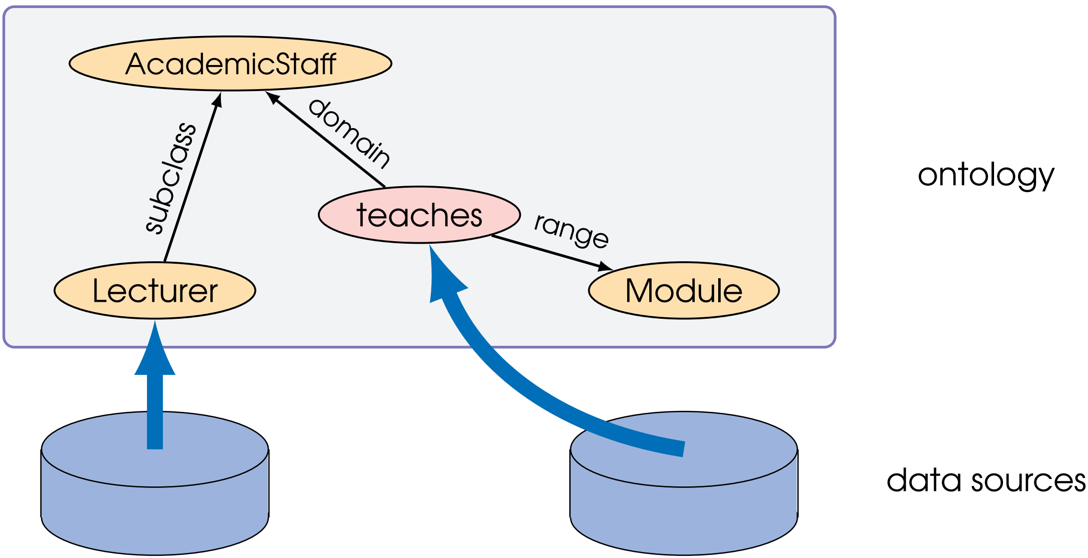
Horn DLs with $\sqcap$ and $\exists$

DL-Lite and $\mathcal{EL}$ families $\leq$ P

Which DLs are suitable for ontology-based data access?

Aim: to achieve **logical transparency** in accessing data

- hide from the user where and how data is stored
- present only a **conceptual view** of the data
- **query** the data sources through the **conceptual model** using **RDBMSs**



Designing DL for conceptual data modelling

Translating into DL:

TopManager $\sqsubseteq$ Manager

AreaManager $\sqsubseteq$ $\neg$TopManager

Manager $\sqsubseteq$ AreaManager $\sqcup$ TopManager

Employee $\sqsubseteq$ $\exists$*salary*.$\top$

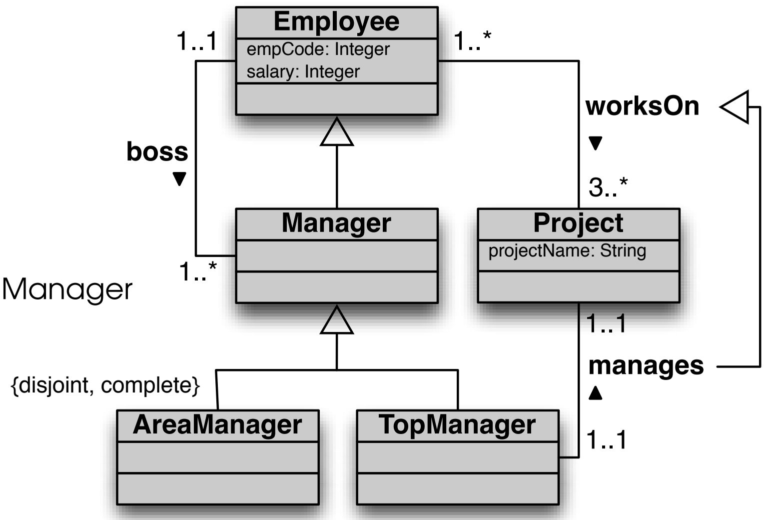
$\exists$*salary*$^-$.$\top$ $\sqsubseteq$ Integer

≥ 2 *salary*.$\top$ $\sqsubseteq$ $\bot$

Project $\sqsubseteq$ ≥ 3 *worksOn*$^-$.$\top$

manages $\sqsubseteq$ *worksOn*

CEO $\sqcap$ (≥ 5 *worksOn*.$\top$) $\sqcap$ $\exists$*manages*.$\top$ $\sqsubseteq$ $\bot$ (integrity constraint)



DL-Lite

Basic *DL-Lite* logics

under UNA

1. *DL-Lite*$^{\mathcal{N}}_{bool}$

$$R \ ::= \ P \ \mid \ P^-$$
$$B \ ::= \ \bot \ \mid \ A \ \mid \ \geq qR$$
$$C \ ::= \ B \ \mid \ \neg C \ \mid \ C_1 \sqcap C_2$$

TBox axioms $C_1 \sqsubseteq C_2$

combined complexity sat.: **NP**
data comp. instance: in **AC^0**
data comp. query: **coNP**

2. *DL-Lite*$^{\mathcal{N}}_{horn}$

TBox axioms $B_1 \sqcap \cdots \sqcap B_n \sqsubseteq B$

combined complexity: **P**
data comp. instance: in **AC^0**
data comp. query: in **AC^0**

3. *DL-Lite*$^{\mathcal{N}}_{krom}$

TBox axioms $B_1 \sqsubseteq B_2 \quad B_1 \sqsubseteq \neg B_2 \quad \neg B_1 \sqsubseteq B_2$

comb. comp.: **NLogSpace**
d.c. instance: in **AC^0**
d.c. query: **coNP**

4. *DL-Lite*$^{\mathcal{N}}_{core}$ $=$ *DL-Lite*$^{\mathcal{N}}_{horn}$ $\cap$ *DL-Lite*$^{\mathcal{N}}_{krom}$

comb. comp.: **NLogSpace**
d.c. instance: in **AC^0**
d.c. query: in **AC^0**

DL-Lite$_{bool}$, *DL-Lite*$_{horn}$, *DL-Lite*$_{krom}$, *DL-Lite*$_{core}$: only $\exists R$ available

Observations and examples

DL-Lite can only speak about the **domains** and **ranges** of binary relations, and
how many successors and predecessors a point can have
but **not** about the **types** of these successors/predecessors;
types are defined **uniformly** by domain/range constraints

Examples. Describe the models of the following KBs:

1. $\mathcal{T} = \{\top \sqsubseteq \exists R, \quad \geq 2R \sqsubseteq \bot\}$, ($R$ is total and functional)
   $\mathcal{A} = \emptyset$

2. $\mathcal{T} = \{A \sqsubseteq \neg\exists R^-, \quad A \sqsubseteq \exists R, \quad \exists R^- \sqsubseteq \exists R, \quad \geq 2^- R \sqsubseteq \bot\}$,
   $\mathcal{A} = \{A(a)\}$

- **Infinite** models are required; **no** finite model property
- Tree model property (see page 19)
- Can be simulated by first-order formulas with **one** variable (see page 20)

Bisimulations for *DL-Lite*$^{\mathcal{N}}_{bool}$

Let $\mathcal{I}$ and $\mathcal{J}$ be two interpretations.

A relation $\varrho \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ is called a **lite-bisimulation** between $\mathcal{I}$ and $\mathcal{J}$ if

(concept) for every concept name A, if $x \varrho y$ then $x \in A^{\mathcal{I}}$ iff $y \in A^{\mathcal{J}}$

(role) for every role R, if $x \varrho y$ then $x \in (=qR)^{\mathcal{I}}$ iff $y \in (=qR)^{\mathcal{J}}$

where $q \in \mathbb{N} \cup \{\infty\}$, $=qR ::= \geq qR \sqcap \neg{\geq}(q+1)R$

$(\mathcal{I}, x) \sim (\mathcal{J}, y)$ if there is a lite-bisimulation ϱ between $\mathcal{I}$ and $\mathcal{J}$ with $x \varrho y$

DL-Lite$^{\mathcal{N}}_{bool}$ concepts are **invariant under lite-bisimulations**, that is,

if $(\mathcal{I}, x) \sim (\mathcal{J}, y)$ then $x \in C^{\mathcal{I}}$ iff $y \in C^{\mathcal{J}}$, for every concept C

A first-order formula $\varphi(x)$ is equivalent to a *DL-Lite*$^{\mathcal{N}}_{bool}$ concept iff

$\varphi(x)$ is invariant under lite-bisimulations

Global lite-bisimulations for *DL-Lite*$^{\mathcal{N}}_{bool}$

A lite-bisimulation relation ϱ between $\mathcal{I}$ and $\mathcal{J}$ is **global** if

– for every $x \in \Delta^{\mathcal{I}}$ there is $y \in \Delta^{\mathcal{J}}$ with $x \varrho y$, and

– for every $y \in \Delta^{\mathcal{J}}$ there is $x \in \Delta^{\mathcal{I}}$ with $x \varrho y$

$\mathcal{I}$ is **lite-bisimilar** to $\mathcal{J}$, $\mathcal{I} \sim \mathcal{J}$, if
there is a global lite-bisimulation between $\mathcal{I}$ and $\mathcal{J}$

DL-Lite$^{\mathcal{N}}_{bool}$ TBoxes are **invariant under global lite-bisimulations**, that is,
if $\mathcal{I} \sim \mathcal{J}$ then $\mathcal{I} \models \mathcal{T}$ iff $\mathcal{J} \models \mathcal{T}$, for every *DL-Lite*$^{\mathcal{N}}_{bool}$ TBox $\mathcal{T}$

Given $\mathcal{I}$ and $x \in \Delta^{\mathcal{I}}$, let $t_{\mathcal{I}}(x) = \{C \mid x \in C^{\mathcal{I}}\}$ — the **type** of x in $\mathcal{I}$

$T_{\mathcal{I}} = \{t_{\mathcal{I}}(x) \mid x \in \Delta^{\mathcal{I}}\}$ — set of all types in $\mathcal{I}$

$\mathcal{I} \sim \mathcal{J}$ iff $T_{\mathcal{I}} = T_{\mathcal{J}}$ models are determined by their types $\rightsquigarrow$ 1-ary predicates

Tree model propety

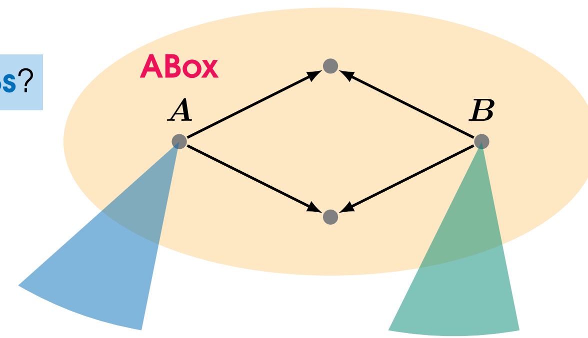
Every model of a *DL-Lite*$^{\mathcal{N}}_{bool}$ TBox is globally lite-bisimilar to a **tree-shaped model**

Examples. Construct a tree-shaped model which is globally lite-bisimilar to

$$\begin{tikzcd} t_2 \arrow[rr, "R"] & & t_3 \\ & t_1 \arrow[ul, "R"] \arrow[ur, "R"'] & \end{tikzcd}$$

where t_1, t_2, t_3 are distinct types

Tree models of *DL-Lite*$^{\mathcal{N}}_{bool}$ **KBs**?



Why is the tree-model property so important?

Embedding *DL-Lite* into 1-variable FO logic

Satisfiability of $\textit{DL-Lite}^{\mathcal{N}}_{bool}$ KBs is **NP**-complete (for combined complexity)

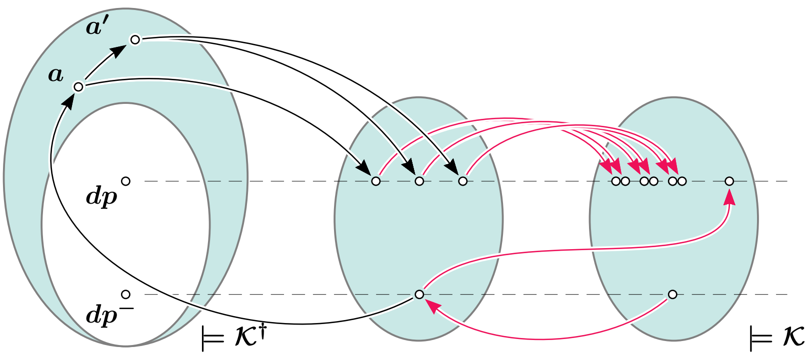
Proof $\textit{DL-Lite}^{\mathcal{N}}_{bool}$ $\mathcal{K} \rightsquigarrow \mathcal{K}^{\dagger}$ (a universal 1-variable FO formula)

$$\mathcal{T} = \{A \sqsubseteq \exists P^-, \exists P^- \sqsubseteq A, \ A \sqsubseteq \geq 2\, P, \ \top \sqsubseteq \leq 1\, P^-, \exists P \sqsubseteq A\}, \ \mathcal{A} = \{A(a), P(a, a')\}$$

$$\begin{aligned}\forall x \Big[& (A(x) \to E_1P^-(x)) \land (E_1P^-(x) \to A(x)) \land (A(x) \to E_2P(x)) \land \neg E_2P^-(x) \land (E_1P(x) \to A(x)) \\ & \land \ (E_2P(x) \to E_1P(x)) \land (E_2P^-(x) \to E_1P^-(x)) \\ & \land \ (E_1P(x) \to E_1P^-(dp^-)) \land (E_1P^-(x) \to E_1P(dp)) \Big] \ \land \ A(a) \land E_1P(a) \land E_1P^-(a')\end{aligned}$$

$(\exists P)^{\mathcal{I}} \neq \emptyset$ iff $(\exists P^-)^{\mathcal{I}} \neq \emptyset$

$\exists x\, E_1P(x) \leftrightarrow \exists x\, E_1P^-(x)$



$\mathcal{K}$ is satisfiable iff $\mathcal{K}^{\dagger}$ is.

$\mathcal{K}^{\dagger}$ computed in **LogSpace**.

$\mathcal{K}^{\dagger}$ says that

– $\exists$ appropriate dr

– $\forall$ point is of proper type

DL-Lite Horn, Krom and core (under UNA)

For *DL-Lite*$^{\mathcal{N}}_{horn}$ KBs $\mathcal{K}$, the translation $\mathcal{K}^{\dagger}$ is a conjunction of formulas of the form

(horn) $$\forall x\ (A_1(x) \land \cdots \land A_n(x) \to A(x))$$

Satisfiability of **Horn formulas** is **P-complete** (combined complexity)

For *DL-Lite*$^{\mathcal{N}}_{krom}$ KBs $\mathcal{K}$, the translation $\mathcal{K}^{\dagger}$ is a conjunction of formulas of the form

(krom) $$\forall x\ (A_1(x) \to A_2(x)),\ \ \forall x\ (A_1(x) \to \neg A_2(x)),\ \ \forall x\ (\neg A_1(x) \to A_2(x))$$

Satisfiability of **Krom formulas** is **NLogSpace-complete** (combined complexity)

For *DL-Lite*$^{\mathcal{N}}_{core}$ KBs $\mathcal{K}$, the translation $\mathcal{K}^{\dagger}$ is a conjunction of formulas of the form

(core) $$\forall x\ (A_1(x) \to A_2(x)),\ \ \forall x\ (A_1(x) \to \neg A_2(x))$$

Satisfiability of **core formulas** is **NLogSpace-complete** (combined complexity)

Canonical models for *DL-Lite*$^{\mathcal{N}}_{horn}$ and *DL-Lite*$^{\mathcal{N}}_{core}$

For a consistent *DL-Lite*$^{\mathcal{N}}_{horn}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, the **canonical model $\mathcal{I}_{\mathcal{K}}$** is constructed as follows:

1. take the ABox and add $\geq q R$ to $t(a)$ if q-many R-arrows start from a in $\mathcal{A}$
2. 'saturate' the existing types by applying the **rules** in $\mathcal{T}$
3. for every x, if $(\geq q R) \in t(x)$ but there are $< q$ R-arrows starting from x, draw the missing R-arrows to **fresh** points and add $\exists R^-$ to their types
4. go to Step 2

- If $\mathcal{I} \models \mathcal{K}$ then there is a map $h\colon \Delta^{\mathcal{I}_{\mathcal{K}}} \to \Delta^{\mathcal{I}}$ such that, for all $x, y \in \Delta^{\mathcal{I}_{\mathcal{K}}}$, basic concepts B and roles R,
  – if $x \in B^{\mathcal{I}_{\mathcal{K}}}$ then $h(x) \in B^{\mathcal{I}}$;
  – if $(x, y) \in R^{\mathcal{I}_{\mathcal{K}}}$ then $(h(x), h(y)) \in R^{\mathcal{I}}$
- $\mathcal{K} \models q(\vec{a})$ iff $\mathcal{I}_{\mathcal{K}} \models q(\vec{a})$

Exercise: construct $\mathcal{I}_{\mathcal{K}}$ for $\mathcal{K}$ on page 20

DL-Lite with role hierarchies

DL-Lite$^{\mathcal{F}}_{core}$ (only functionality) is **NLogSpace**-complete for combined complexity and in **AC^0** for data complexity

DL-Lite$^{\mathcal{HF}}_{core}$ (*DL-Lite*$^{\mathcal{F}}_{core}$ + $R_1 \sqsubseteq R_2$) is **ExpTime**-complete for combined complexity and **P**-complete for data complexity

Example: $A_1 \sqcap A_2 \sqsubseteq C$ can be simulated by the axioms:

$$
\begin{array}{rcl@{\qquad\qquad}rcl}
A_1 & \sqsubseteq & \exists R_1 & A_2 & \sqsubseteq & \exists R_2 \\
R_1 & \sqsubseteq & R_{12} & R_2 & \sqsubseteq & R_{12} \\
\geq 2\, R_{12} & \sqsubseteq & \bot & & & \\
\exists R_1^- & \sqsubseteq & \exists R_3^- & & & \\
\exists R_3 & \sqsubseteq & C & & & \\
R_3 & \sqsubseteq & R_{23} & R_2 & \sqsubseteq & R_{23} \\
\geq 2\, R_{23}^- & \sqsubseteq & \bot & & &
\end{array}
$$

DL-Lite$_{\alpha}^{(\mathcal{RN})}$: pushing the limits of *DL-Lite*

- **role inclusions + number restrictions**

if R has a proper sub-role in $\mathcal{T}$ then $\mathcal{T}$ contains
no *negative occurrences* of $\geq q\, R$ or $\geq q\, R^-$ with $q \geq 2$

- positive occurrences of qualified number restrictions $\geq q\, R.C$

if $\geq q\, R.C$ occurs in $\mathcal{T}$ then $\mathcal{T}$ contains
no *negative occurrences* of $\geq q'\, R$ or $\geq q'$ *inv*(R) with $q' \geq 2$

no TBox can contain both a functionality constraint $\geq 2\, R \sqsubseteq \bot$ and $\geq q\, R.C$, for any $q \geq 1$

- role **disjointness**, **symmetry**, **asymmetry**, **reflexivity** and **irreflexivity** constraints

all these extensions do not change the complexity
in particular, same complexity of *DL-Lite*$_{\alpha}^{(\mathcal{RN})}$ and *DL-Lite*$_{\alpha}^{\mathcal{N}}$

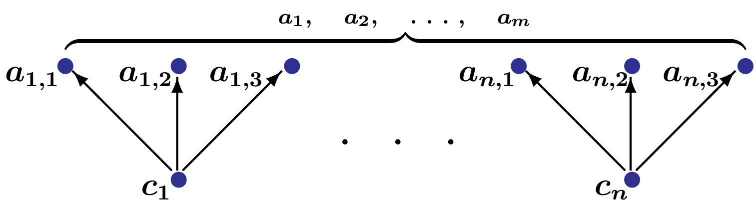
NB. transitive roles do not change the combined complexity
(NLogSpace-hard for data complexity)

DL-Lite without UNA

Without UNA, satisfiability of *DL-Lite*$_\alpha^{\mathcal{N}}$ KBs is **NP-complete** w.r.t. both **combined** and **data complexity,** for any $\alpha \in \{core, krom, horn, bool\}$

source of non-determinism: different ways of identifying ABox individuals

Lower bound: by reduction of **monotone 1-in-3 3SAT** $\bigwedge_{k=1}^{n}(a_{k,1} \lor a_{k,2} \lor a_{k,3})$



$$\mathcal{A} = \{a_{k,i} \neq a_{k,j} \mid i \neq j\} \cup \{P(c_k, a_{k,j}) \mid k \leq n, \ \ j \leq 3\} \qquad \mathcal{T} = \{\geq 4P \sqsubseteq \bot\}$$

Answer is **yes** iff there is a (true) variable a_i in the given CNF such that $\mathcal{K}_{a_i} = (\mathcal{T}, \mathcal{A} \cup \{P(c_k, a_i) \mid k \leq n\})$ is satisfiable without UNA

NB: One can get rid of $\neq$ in $\mathcal{A}$

$DL\text{-}Lite_{\alpha}^{(\mathcal{R},\mathcal{F})}$ without UNA

Deterministically glue together those ABox objects a and b for which

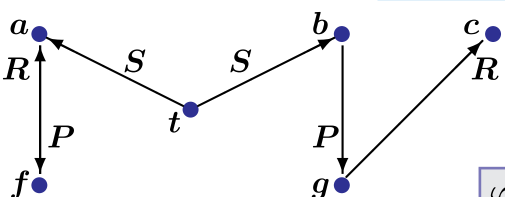
- either $\mathcal{A} \models (a = b)$
- or $\mathcal{T} \models (\geq 2R \sqsubseteq \bot)$ and $R(c,a)$, $R(c,b)$, for some ABox object c

This gives a **polynomial** reduction of no-UNA to UNA for $DL\text{-}Lite_{\alpha}^{(\mathcal{R},\mathcal{F})}$ logics,

which increases complexity by **P**

Can't do better: functionality constraints can encode inference for Horn CNFs

Example: Represent $\varphi = (a \wedge b \rightarrow c) \wedge a \wedge b$ as follows:



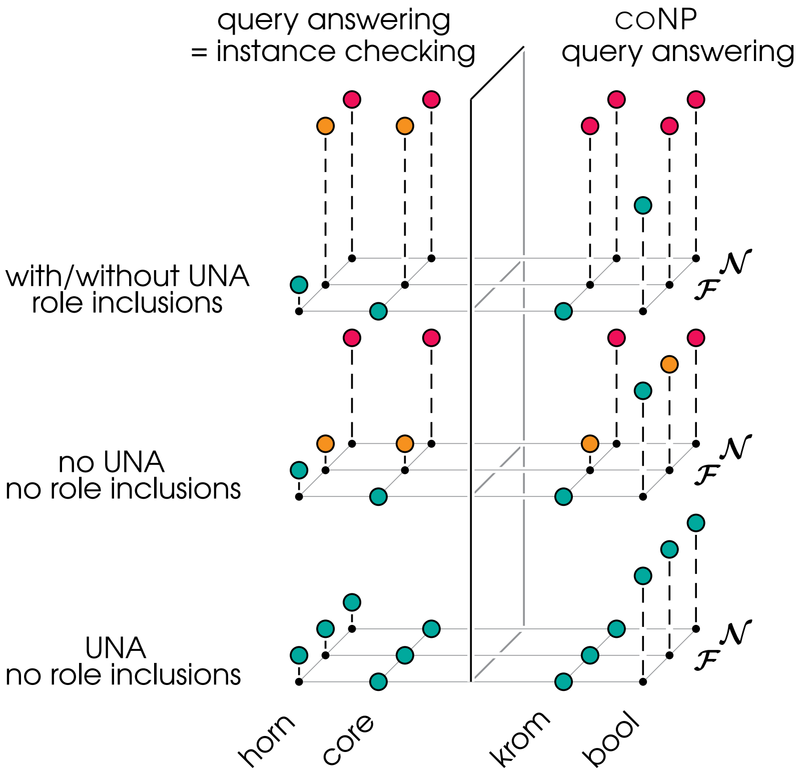
$\mathcal{A}$ includes all these P-, R- and S-arrows

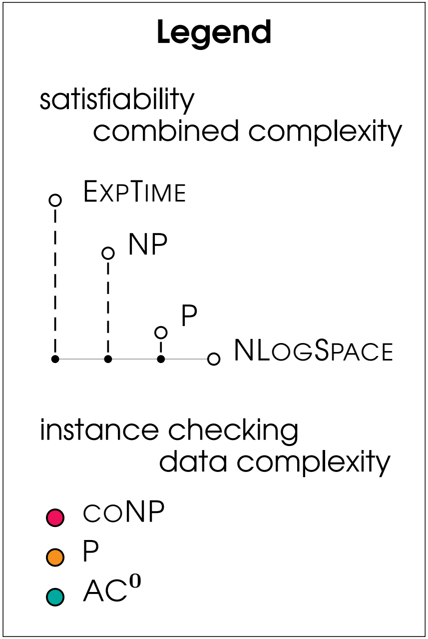
$\mathcal{T}$ says that P, R and S are functional

$\varphi \models c$ iff $(\mathcal{T}, \mathcal{A} \cup \{\neg S(t,c)\})$ is not satisfiable

Without UNA, satisfiability of $DL\text{-}Lite_{\alpha}^{(\mathcal{R},\mathcal{F})}$ KBs (with or without $=$ and $\neq$) is **P-hard** for both **combined** and **data complexity**

The *DL-Lite* family: complexity-scape





OWL 2 QL

'An **OWL 2 profile** is a trimmed down version of OWL 2 that trades some expressive power for the **efficiency of reasoning**'

'**OWL 2 QL** is aimed at applications that use very large volumes of instance data, and where query answering is the most important reasoning task.

In **OWL 2 QL**, conjunctive query answering can be implemented using conventional relational database systems.'

OWL 2 QL $=$ $\mathit{DL\text{-}Lite}^{\mathcal{H}}_{core}$ with/without UNA

with $\neq$ (but no $=$)

with (a)symmetric, (ir)reflexive and disjoint roles

(but no transitive roles)

Why not $\mathit{DL\text{-}Lite}^{\mathcal{H}}_{horn}$?

OWL 2 EL

'The **OWL 2 EL** profile is designed as a subset of OWL 2 that

- is particularly suitable for applications employing ontologies that define very large numbers of classes and/or properties,
- captures the expressive power used by many such ontologies, and
- for which ontology consistency, class expression subsumption, and instance checking can be decided in polynomial time.'

For example, OWL 2 EL provides class constructors that are sufficient to express the very large biomedical ontology SNOMED CT (≈ 400.000 axioms)

$$\begin{aligned}
\text{Pericardium} &\sqsubseteq \text{Tissue} \sqcap \exists \text{cont_in}.\text{Heart}\\
\text{Pericarditis} &\sqsubseteq \text{Inflammation} \sqcap \exists \text{has_loc}.\text{Pericardium}\\
\text{Inflammation} &\sqsubseteq \text{Disease} \sqcap \exists \text{acts_on}.\text{Tissue}\\
\text{Disease} \sqcap \exists \text{has_loc}.\exists \text{cont_in}.\text{Heart} &\sqsubseteq \text{Heartdisease} \sqcap \text{NeedsTreatment}
\end{aligned}$$

Basic $\mathcal{EL}$

$\mathcal{EL}$ concepts: $C \ ::= \ \top \ \mid \ \bot \ \mid \ A \ \mid \ \exists R.C \ \mid \ C_1 \sqcap C_2$

$\mathcal{EL}$ TBoxes: finite sets of CIs $C_1 \sqsubseteq C_2$

$\mathcal{EL}$ ABoxes: finite sets of assertions $C(a), \quad R(a,b)$

Concept satisfiability: given $\mathcal{T}$, C, decide whether there is $\mathcal{I} \models \mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$

Subsumption: given $\mathcal{T}$ and concepts C, D, decide whether $\mathcal{T} \models C \sqsubseteq D$

Instance checking: given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, C and an individual a from $\mathcal{A}$, decide whether $\mathcal{K} \models C(a)$

Reducible to each other!

Conjunctive query answering: given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a CQ $q(\vec{x})$ and a tuple $\vec{a}$ of individual names from $\mathcal{A}$, decide whether $\mathcal{K} \models q(\vec{a})$

Observations and examples

$\mathcal{EL}$ can specify some **positive** information about types of points, viz:

✓ that a point belongs to a certain concept

(but not that it **does not** belong to a concept);

✓ that there is an outgoing R-arrow which ends in a certain concept

(but not that **all** outgoing R-arrows end in the concept);

✓ that some concepts are **disjoint**

Example. Describe the models of the following KBs:

$$\mathcal{T} = \{A \sqsubseteq B_1, \quad B_1 \sqsubseteq \exists R.B_1, \quad \exists R.B_1 \sqsubseteq B_2, \quad B_1 \sqcap B_2 \sqsubseteq \exists S.B_2\},$$
$$\mathcal{A} = \{A(a)\}$$

- **Finite** models are enough (finite model property)
- Tree model property (but infinite!)
- Not 'local' as *DL-Lite*; one-variable first-order formulas are not enough

Simulations for $\mathcal{EL}$

Let $\mathcal{I}$ and $\mathcal{J}$ be two interpretations.

A relation $\varrho \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ is called a **simulation** of $\mathcal{I}$ in $\mathcal{J}$ if

(concept) for every concept name A, if $x \varrho y$ then $x \in A^{\mathcal{I}} \;\Rightarrow\; y \in A^{\mathcal{J}}$

(role) for every role name R, if $x \varrho y$ then

$$(x, x') \in R^{\mathcal{I}} \;\Rightarrow\; \exists y' \left[(y, y') \in R^{\mathcal{J}} \text{ and } x' \varrho y'\right]$$

$(\mathcal{I}, x) \preceq (\mathcal{J}, y)$ if there is a simulation ϱ of $\mathcal{I}$ in $\mathcal{J}$ with $x \varrho y$

$\mathcal{EL}$ concepts are **preserved under simulations**, that is,

if $(\mathcal{I}, x) \preceq (\mathcal{J}, y)$ then $x \in C^{\mathcal{I}} \;\Rightarrow\; y \in C^{\mathcal{J}}$, for every concept C

$\mathcal{EL}$ concepts cannot distinguish between $(\mathcal{I}, x)$ and $(\mathcal{J}, y)$ if

$(\mathcal{I}, x) \preceq (\mathcal{J}, y)$ and $(\mathcal{J}, y) \preceq (\mathcal{I}, x)$

What are the differences between *DL-Lite* and $\mathcal{EL}$?

Tree canonical models for $\mathcal{EL}$

(basically the same construction as for *DL-Lite*$^{\mathcal{N}}_{horn}$)

For a consistent $\mathcal{EL}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, the **canonical model** $\mathcal{I}_{\mathcal{K}}$ is constructed as follows

1. 'saturate' the existing types (starting with $\mathcal{A}$) by applying the **rules** in $\mathcal{T}$
2. for every x, if $\exists R.C \in t(x)$ but no R-arrow from x leads to C, draw an R-arrow to a **fresh** point and add C to its type
3. go to Step 1

- If $\mathcal{I} \models \mathcal{K}$ then there is a map $h \colon \Delta^{\mathcal{I}_{\mathcal{K}}} \to \Delta^{\mathcal{I}}$ such that, for all $x, y \in \Delta^{\mathcal{I}_{\mathcal{K}}}$, concept and role names A and R,
  – if $x \in A^{\mathcal{I}_{\mathcal{K}}}$ then $h(x) \in A^{\mathcal{I}}$;
  – if $(x, y) \in R^{\mathcal{I}_{\mathcal{K}}}$ then $(h(x), h(y)) \in R^{\mathcal{I}}$
- $\mathcal{K} \models q(\vec{a})$ iff $\mathcal{I}_{\mathcal{K}} \models q(\vec{a})$

$\mathcal{I}_{\mathcal{K}}$ can be infinite

Compact canonical models for $\mathcal{EL}$

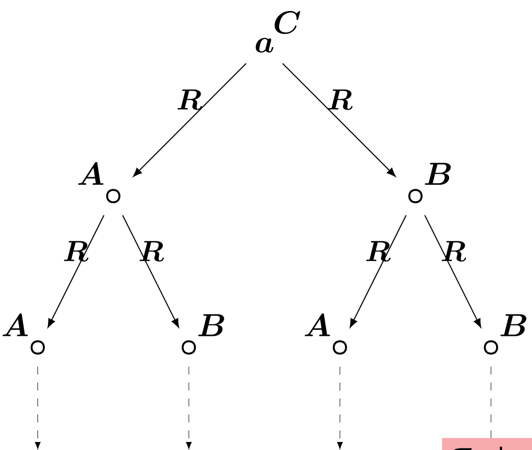
ABox $\mathcal{A}$

C a

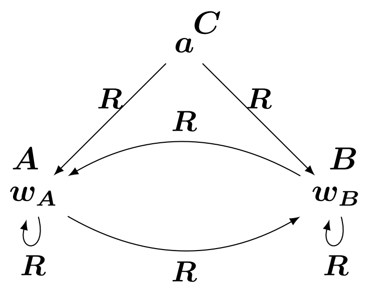
TBox $\mathcal{T}$

$\top \sqsubseteq \exists R.A, \qquad \top \sqsubseteq \exists R.B$

Canonical model $\mathcal{I}_\mathcal{K}$



Compact canonical model $\mathcal{C}_\mathcal{K}$



$\mathcal{I}_\mathcal{K}$ is obtained by unravelling $\mathcal{C}_\mathcal{K}$; $\quad (\mathcal{C}_\mathcal{K}, a) \preceq (\mathcal{I}_\mathcal{K}, a)$

Constructing $\mathcal{C}_\mathcal{K}$

Compact canonical interpretation $\mathcal{C}_\mathcal{K}$:

Con($\mathcal{K}$) = the set of all concepts in $\mathcal{K}$

$\Delta^{\mathcal{C}_\mathcal{K}} = \mathsf{Ind}(\mathcal{A}) \cup \{w_C \mid C \in \mathsf{Con}(\mathcal{K})\}$ — w_C is a **witness for C**

$A^{\mathcal{C}_\mathcal{K}} = \{a \mid \mathcal{K} \models A(a)\} \cup \{w_C \mid \mathcal{T} \models C \sqsubseteq A\}$ — (A a concept name)

$$R^{\mathcal{C}_\mathcal{K}} = \{(a,b) \mid R(a,b) \in \mathcal{A}\} \cup \{(a, w_C) \mid \mathcal{K} \models \exists R.C(a)\} \cup \{(w_C, w_D) \mid \mathcal{T} \models C \sqsubseteq \exists R.D\}$$

(R a role name)

Construct $\mathcal{C}_\mathcal{K}$ for $\mathcal{K}$ on page 31

- Can be constructed in polynomial time in the size of $\mathcal{K}$
- Inconsistency can be detected during construction

$\leadsto$ Satisfiability of $\mathcal{EL}$ KBs is PTime-complete

$\mathcal{EL}$++ and OWL 2 EL

$\mathcal{EL}$ can be extended, without losing **tractability**, with

- ✓ role implications $R_1 \circ \cdots \circ R_n \sqsubseteq R$ (e.g., $R \circ R \sqsubseteq R$ means transitivity)
- ✓ range restrictions $\top \sqsubseteq \forall R.C$
- ✓ domain restrictions $\top \sqsubseteq \forall R^-.C$
- ✓ nominals $\{a\}$, a an individual name

$\approx$ OWL 2 EL

Extensions with any of the constructs

$C \sqcup D$, $\forall R.C$, $\geq qR$, R^-, symmetric roles

result in **ExpTime-hard** reasoning

Exercise: construct an $\mathcal{ELI}$ ($\mathcal{EL}$ + inverse roles) KB $\mathcal{K}$ with $\mathcal{C}_\mathcal{K}$ of exponential size