On the complexity of querying data through ontologies

Meghyn Bienvenu

CNRS, Université Paris Sud

Querying data through ontologies

Idea: exploit semantic information from ontology when querying data

Example application: querying patient data

- ontology describes medical terms and relationships between terms
  - Hodgkin's lymphoma is a type of cancer
  - hypertension and high blood pressure are synonyms
- user formulates query using vocabulary of ontology
  - find patients suffering from cancer and high blood pressure
- system performs reasoning to find all (deducible) answers

In this talk:

- quick tour of the field
  - focus on description logic (DL) ontologies
- some pointers to current research

Syntax

Vocabulary

- atomic concepts (unary relations) — *Mother, Student*
- atomic roles (binary relations) — *parentOf, partOf*
- individuals (constants) — *marie, pierre*

Complex concepts

- concept constructors: $\top$, $\neg C$, $C \sqcap D$, $\exists r.C$, $\geq n\, r.C$, ...
- examples with translation to FOL:
  - $Person \sqcap \neg Student$ — $Person(x) \wedge \neg Student(x)$
  - $\exists parentOf.Female$ — $\exists y.parentOf(x, y) \wedge Female(y)$
  - $\geq 2\, parentOf.\top$ — $\exists y, z.parentOf(x, y) \wedge parentOf(x, z) \wedge y \neq z$

Complex roles:

- role constructors: $^-$ (inverse), $\circ$ (composition), ...

Semantics

Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

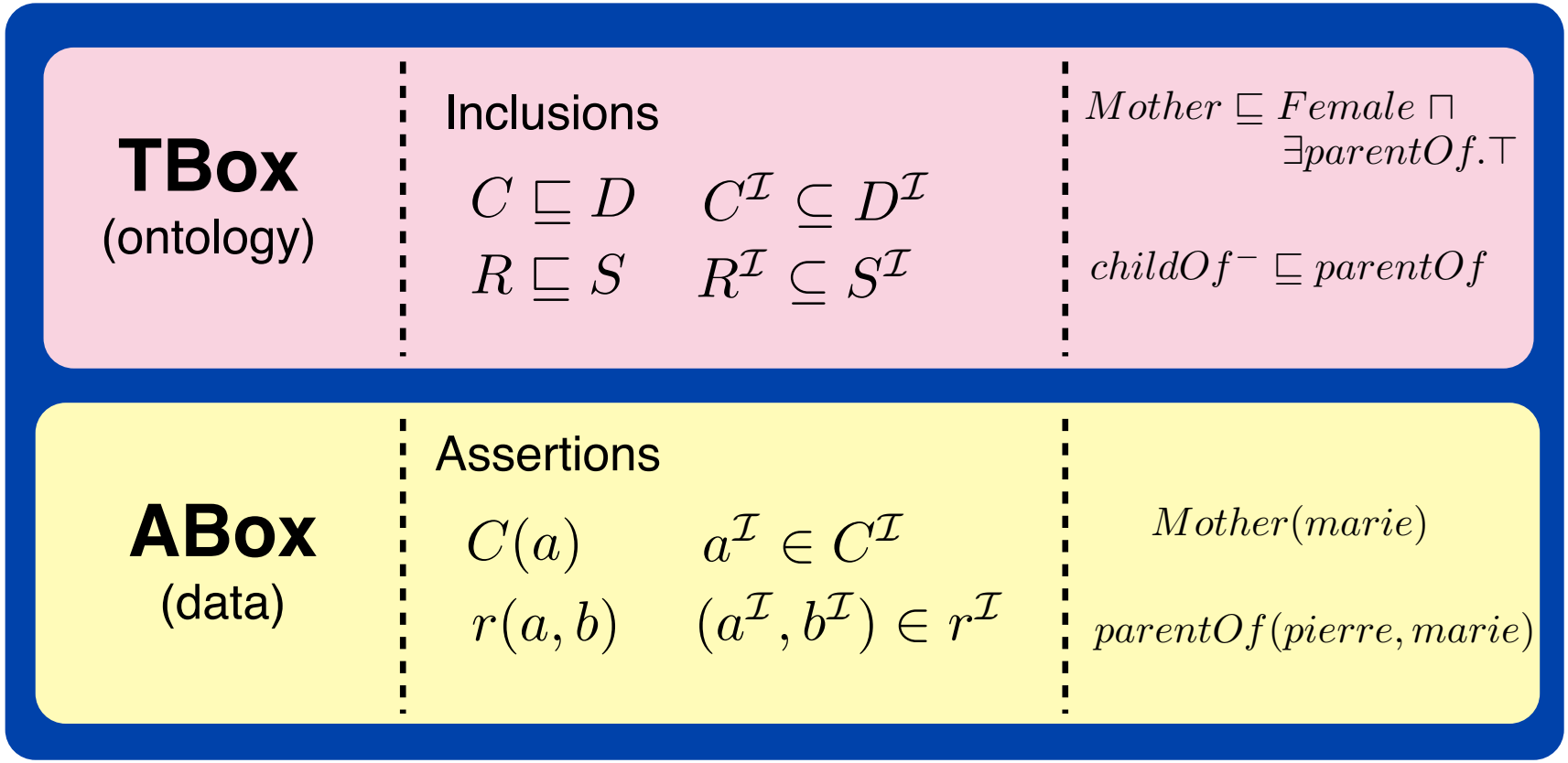
- $\Delta^{\mathcal{I}}$ is a non-empty set (universe)
- $\cdot^{\mathcal{I}}$ is a function
  - individual $a \mapsto$ an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
  - atomic concept $A \mapsto$ a unary relation $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
  - atomic role $r \mapsto$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

Extension to complex concepts and roles:

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ and $\bot^{\mathcal{I}} = \emptyset$
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ and $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$ and $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
- $(\exists r.C)^{\mathcal{I}} = \{u \mid \text{there exists } v \text{ such that } (u, v) \in r^{\mathcal{I}} \text{ and } v \in C^{\mathcal{I}}\}$
- $(\leq n\, r.C)^{\mathcal{I}} = \{u \mid \text{at most } n \text{ } v \text{ such that } (u, v) \in r^{\mathcal{I}} \text{ and } v \in C^{\mathcal{I}}\}$
- $(r^{-})^{\mathcal{I}} = \{(u, v) \mid (v, u) \in r^{\mathcal{I}}\}$

Knowledge bases

DL knowledge base = TBox + ABox



$\mathcal{I}$ is a model of $\mathcal{K}$ if $\mathcal{I}$ satisfies all assertions and axioms in $\mathcal{K}$

Reasoning tasks

Classical reasoning tasks:

subsumption does $\mathcal{T} \models C \sqsubseteq D$?

classification find all A,B such that $\mathcal{T} \models A \sqsubseteq B$

satisfiability is $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ satisfiable ?

instance checking does $(\mathcal{T}, \mathcal{A}) \models C(b)$?

Relationships among these tasks:

- $\mathcal{T} \models C \sqsubseteq D$ iff $(\mathcal{T}, \{C(a)\}) \models D(a)$
- $\mathcal{K}$ satisfiable iff $\mathcal{K} \not\models B(a)$ (where B fresh concept, not in $\mathcal{K}$)

Variants: subsumption without TBox, satisfiability of a concept, ...

Short history of DLs

1985-1995 Negative results (undecidability, NP-hardness)

Tractable fragments ($\mathcal{FL}_0, \mathcal{AL}$) based upon $\sqcap$ and $\forall R.C$

Complexity: subsumption in PTIME (but no TBox!)

Algorithms: normalization + structural comparison

Short history of DLs

1985-1995 Negative results (undecidability, NP-hardness)

Tractable fragments ($\mathcal{FL}_0, \mathcal{AL}$) based upon $\sqcap$ and $\forall R.C$

Complexity: subsumption in PTIME (but no TBox!)

Algorithms: normalization + structural comparison

1995-2005 Expressive logics like $\mathcal{SHOIQ}$ which offers:

$\neg$, $\sqcup$, $\exists R.C$, $\geq n.C$, $\leq n.C$, r^-, $r \sqsubseteq s$, $(\text{trans}\, r), \{a\}, \ldots$

Complexity: subsumption $\geq$ EXPTIME (with TBox)

Algorithms: highly optimized tableaux reasoners

Despite very high complexity, good performance !

Short history of DLs

1985-1995 Negative results (undecidability, NP-hardness)
Tractable fragments ($\mathcal{FL}_0, \mathcal{AL}$) based upon $\sqcap$ and $\forall R.C$
Complexity: subsumption in PTIME (but no TBox!)
Algorithms: normalization + structural comparison

1995-2005 Expressive logics like $\mathcal{SHOIQ}$ which offers:
$\neg$, $\sqcup$, $\exists R.C$, $\geq n.C$, $\leq n.C$, r^-, $r \sqsubseteq s$, $(\text{trans}\, r)$, $\{a\}$, ...
Complexity: subsumption $\geq$ EXPTIME (with TBox)
Algorithms: highly optimized tableaux reasoners
Despite very high complexity, good performance !

2005-present Lightweight description logics, motivated by applications
DL-Lite family (OWL 2 QL) and $\mathcal{EL}$ family (OWL 2 EL)
Algorithms: query rewriting and/or saturation

Motivations for lightweight DLs

1. **Applications requiring more expressive queries**
   - conjunctive queries (like in databases)

     $q(x, z) = \mathit{Female}(x) \wedge \mathit{childOf}(x, y) \wedge \mathit{Female}(y) \wedge \mathit{childOf}(y, z) \wedge \mathit{Female}(z)$
   - difficulty: not reducible to classical reasoning tasks

2. **Applications involving large ontologies and lots of data**
   - scalability is crucial!

Conjunctive queries

Conjunctive queries are an important subclass of first-order logic queries.

They correspond to select-project-join queries in relational DBs.

Formally: a conjunctive query (CQ) has the form

$$q(x_1, \ldots, x_k) \quad = \quad \exists x_{k+1}, \ldots, x_m \ \alpha_1 \wedge \ldots \wedge \alpha_r$$

where $\alpha_1, \ldots, \alpha_r$ are atomic formulae over the variables $x_1, \ldots, x_m$.

Semantics: a tuple $(a_1, \ldots, a_k)$ of constants is a (certain) answer to $q(x_1, \ldots, x_k)$ w.r.t. $\mathcal{K}$ iff $\mathcal{I} \models q[a_1, \ldots, a_k]$ for every model $\mathcal{I}$ of $\mathcal{K}$.

Complexity landscape: expressive DLs

	instance queries	**conjunctive queries**	
	combined	*combined*	*data*
$\mathcal{ALC}(\mathcal{H})$	EXP-complete	EXP-complete	coNP-complete
$\mathcal{ALCI}$, $\mathcal{SH}$, $\mathcal{SHIQ}$	EXP-complete	2EXP-complete	coNP-complete
$\mathcal{SHOIQ}$	NEXP-complete	open	open
$\mathcal{SROIQ}$ (OWL 2)	2NEXP-complete	open	open

$\mathcal{ALC}: \sqcap, \sqcup, \neg, \exists r.C, \forall r.C \quad \mathcal{I}: r^- \quad \mathcal{H}: r \sqsubseteq s \quad \mathcal{O}: \{a\} \quad \mathcal{S}: (\text{trans } r) \quad \mathcal{R}: r \circ t \sqsubseteq s$

- combined complexity: in terms of the TBox, ABox, and query
- data complexity: only in terms of the size of the ABox
  - appropriate when $|\mathcal{A}| >> |\mathcal{T}|$

Disjunction yields coNP-hardness

To illustrate the difficulty of answering CQs,
we show coNP-hardness in data complexity for DLs with disjunction.

For our reduction, we use the coNP-complete problem 2+2UNSAT:

Instance propositional formula $\varphi = c_1 \wedge \ldots \wedge c_n$, where each $c_i = v_{i_1} \vee v_{i_2} \vee \neg v_{i_3} \vee \neg v_{i_4}$ (first two literals positive, last two negative), possibly using truth constants true, false

Problem decide if the formula φ is satisfiable, return "yes" if not satisfiable, "no" if satisfiable

Disjunction yields coNP-hardness

Fixed TBox and query:

- $\mathcal{T} = \{V \sqsubseteq T \sqcup F\}$
- $q = \exists c, v_1, v_2, v_3, v_4 \; P_1(c, v_1) \wedge P_2(c, v_2) \wedge N_1(c, v_3) \wedge N_2(c, v_4)$
  $\wedge F(v_1) \wedge F(v_2) \wedge T(v_3) \wedge T(v_4)$

Given a 2+2CNF $\varphi = c_1 \wedge \ldots \wedge c_n$ over $x_1, \ldots, x_m$, true, false,
we use the following ABox $\mathcal{A}_\varphi$

- for each clause $c_i = v_{i_1} \vee v_{i_2} \vee \neg v_{i_3} \vee \neg v_{i_4}$:
  $P_1(c_i, v_{i_1}), P_2(c_i, v_{i_2}), N_1(c, v_{i_3}), N_2(c, v_{i_4})$
- for each variable x_j: $V(x_j)$
- $T(\text{true}), F(\text{false})$

Can show: $\mathcal{T}, \mathcal{A}_\varphi \models q_\varphi$ if and only if φ is unsatisfiable

Short history of DLs

1985-1995 Negative results (undecidability, NP-hardness)
Tractable fragments (e.g. $\mathcal{AL}$) based upon $\sqcap$ and $\forall R.C$
Complexity: subsumption in PTIME (without TBox)
Algorithms: normalization + structural comparison

1995-2005 Expressive logics like $\mathcal{SHIQ}$ which offers:
$\neg$, $\sqcup$, $\exists R.C$, $\geq n.C$, $\leq n.C$, r^-, $r \sqsubseteq s$, $(\text{trans}\, r)$, ...
Complexity: subsumption $\geq$ EXPTIME (with TBox)
Algorithms: highly optimized tableaux reasoners
Despite very high complexity, good performance !

2005-present Lightweight description logics
DL-Lite family (OWL 2 QL) and $\mathcal{EL}$ family (OWL 2 EL)
Algorithms: query rewriting and/or saturation

The *DL-Lite* family

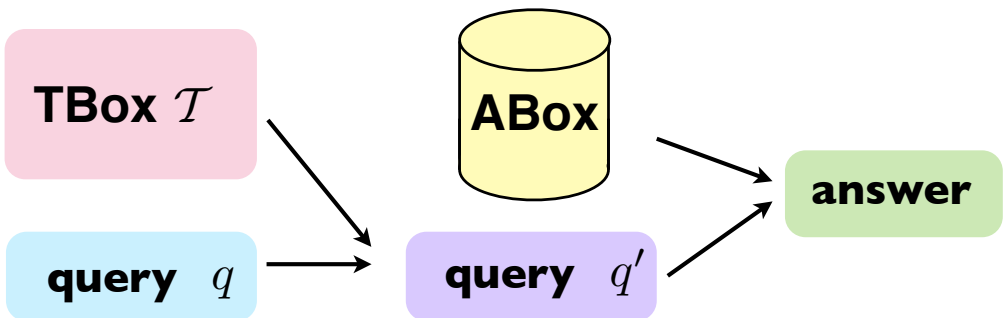
Objective:

useful ontology language allowing efficient conjunctive query answering

Idea: exploit the efficiency of relational DB systems

General approach: query rewriting

- ABox is stored as a traditional database
- the input query is rewritten to integrate the relevant information from the TBox
- the new query is evaluated over the database



Syntax of *DL-Lite*

We present the dialect *DL-Lite*$_{\mathcal{R}}$ (which underlies OWL2 QL).

Assertions: $A(c)$, $r(c, d)$

Inclusions: $B_1 \sqsubseteq B_2$, $B_1 \sqsubseteq \neg B_2$, $S_1 \sqsubseteq S_2$, $S_1 \sqsubseteq \neg S_2$ où

$$B := \top \mid A \mid \exists S \qquad\qquad S := r \mid r^{-}$$

where A is an atomic concept and r an atomic role

Other *DL-Lite* dialects allow:

- functional roles (funct S)
- cardinality restrictions ($\geq q\, S$, $\leq q\, S$)
- Horn inclusions ($B_1 \sqcap \ldots \sqcap B_n \sqsubseteq (\neg) B_{n+1}$)
- roles which are symmetric, asymmetric, reflexive, or anti-reflexive

First-order rewritability

In DL-Lite, satisfiability and CQ answering are both first-order rewritable:

- given a TBox $\mathcal{T}$, we can compute a first-order query $\varphi_{\mathcal{T}}$ such that for every ABox $\mathcal{A}$, we have:

$$(\mathcal{T}, \mathcal{A}) \models \bot \quad \text{iff} \quad \mathcal{I}_{\mathcal{A}} \models \varphi_{\mathcal{T}}$$

- given a TBox $\mathcal{T}$ and a CQ q, we can compute q' such that for every ABox $\mathcal{A}$ and tuple of constants $\vec{a}$, we have:

$$\mathcal{T}, \mathcal{A} \models q[\vec{a}] \quad \text{iff} \quad \mathcal{I}_{\mathcal{A}} \models q'[\vec{a}]$$

where $\mathcal{I}_{\mathcal{A}}$ denotes the interpretation based upon $\mathcal{A}$.

Result: both tasks are in AC^0 ($\subsetneq LOGSPACE \subseteq P$) for data complexity.

- same low data complexity as querying relational databases

Query rewriting by example

$\mathcal{T}$

$\mathsf{TeachingStaff} \sqsubseteq \exists \mathsf{teaches}$ $\quad \mathsf{Lecturer} \sqsubseteq \mathsf{TeachingStaff}$

$\exists \mathsf{teaches} \sqsubseteq \mathsf{TeachingStaff}$ $\quad \mathsf{Lecturer} \sqsubseteq \neg \mathsf{Professor}$

$\mathsf{Professor} \sqsubseteq \mathsf{TeachingStaff}$ $\quad \exists \mathsf{teaches}^{-} \sqsubseteq \mathsf{Course}$

$\mathcal{A}$

$\mathsf{Professor}(\mathsf{Sara})$ $\quad \mathsf{teaches}(\mathsf{Paul}, \mathsf{CS100})$ $\quad \mathsf{Lecturer}(\mathsf{Alex})$

q

$\mathsf{TeachingStaff}(x)$

Query rewriting by example

$\mathcal{T}$

$\mathsf{TeachingStaff} \sqsubseteq \exists \mathsf{teaches}$ $\quad \mathsf{Lecturer} \sqsubseteq \mathsf{TeachingStaff}$

$\exists \mathsf{teaches} \sqsubseteq \mathsf{TeachingStaff}$ $\quad \mathsf{Lecturer} \sqsubseteq \neg \mathsf{Professor}$

$\mathsf{Professor} \sqsubseteq \mathsf{TeachingStaff}$ $\quad \exists \mathsf{teaches}^{-} \sqsubseteq \mathsf{Course}$

$\mathcal{A}$ $\quad \mathsf{Professor}(\mathsf{Sara})$ $\quad \mathsf{teaches}(\mathsf{Paul}, \mathsf{CS100})$ $\quad \mathsf{Lecturer}(\mathsf{Alex})$

q $\quad \mathsf{TeachingStaff}(x)$

q' $\quad \mathsf{TeachingStaff}(\mathsf{x}) \vee \mathsf{Professor}(\mathsf{x}) \vee \mathsf{Lecturer}(\mathsf{x}) \vee \exists \mathsf{y}.\mathsf{teaches}(\mathsf{x}, \mathsf{y})$

Query rewriting by example

$\mathcal{T}$

$\mathsf{TeachingStaff} \sqsubseteq \exists \mathsf{teaches}$ $\quad \mathsf{Lecturer} \sqsubseteq \mathsf{TeachingStaff}$

$\exists \mathsf{teaches} \sqsubseteq \mathsf{TeachingStaff}$ $\quad \mathsf{Lecturer} \sqsubseteq \neg \mathsf{Professor}$

$\mathsf{Professor} \sqsubseteq \mathsf{TeachingStaff}$ $\quad \exists \mathsf{teaches}^{-} \sqsubseteq \mathsf{Course}$

$\mathcal{A}$ $\quad \mathsf{Professor}(\mathsf{Sara})$ $\quad \mathsf{teaches}(\mathsf{Paul}, \mathsf{CS100})$ $\quad \mathsf{Lecturer}(\mathsf{Alex})$

q $\quad \mathsf{TeachingStaff}(x)$

q' $\quad \mathsf{TeachingStaff}(\mathsf{x}) \vee \mathsf{Professor}(\mathsf{x}) \vee \mathsf{Lecturer}(\mathsf{x}) \vee \exists \mathsf{y}.\mathsf{teaches}(\mathsf{x}, \mathsf{y})$

Answers: Sara, Paul, Alex

Query rewriting in practice

Nowadays, several different query rewriting algorithms exist:

- QuOnto, Requiem, Presto, Rapid, Nyaya, ...

All offer excellent theoretical guarantees (data complexity in AC^0)...
but suffer from one major problem:

rewritten queries can be huge! $(\mathcal{O}(|\mathcal{T}| \cdot |q|)^{|q|})$

Database systems handle poorly (if at all) such enormous queries.

Question: can this blowup be avoided?

Complexity of query rewriting

For plain DL-Lite (no role inclusions):

- polytime procedure for query rewriting [Kikot et al., DL'11]

For DL-Lite$_{\mathcal{R}}$ (underlying OWL 2 QL):

- no polytime procedure for FO query rewriting (unless P=NP) [Kikot et al., DL'11]
- polynomial NR datalog rewriting possible (under some assumptions), but resulting program complex [Gottlob & Schwentick, DL'11, KR'12]
- analysis of when polynomial FO rewritings are possible [Kikot et al., KR'12]

The $\mathcal{EL}$ family

The logic $\mathcal{EL}$, and its extensions, are designed for applications requiring very large ontologies.

This family of DLs is well-suited for biomedical applications.

Examples of large biomedical ontologies:

- GO (Gene Ontology), around 20,000 concepts
- NCI (cancer ontology), around 30,000 concepts
- SNOMED (medical ontology), over 300,000 concepts (!)

$$\mathsf{Pericarditis} \sqsubseteq \mathsf{Inflammation} \sqcap \exists \mathsf{loc}.\mathsf{Pericardium}$$
$$\mathsf{Pericardium} \sqsubseteq \mathsf{Tissue} \sqcap \exists \mathsf{partOf}.\mathsf{Heart} \qquad \mathsf{Inflammation} \sqsubseteq \mathsf{Disease}$$
$$\mathsf{Disease} \sqcap \exists \mathsf{loc}.\exists \mathsf{partOf}.\mathsf{Heart} \sqsubseteq \mathsf{HeartDisease}$$

Syntax of $\mathcal{EL}$

The basic logic $\mathcal{EL}$ allows complex concepts of the following form:

$$C := \top \mid C_1 \sqcap C_2 \mid \exists R.C$$

Inclusions $C_1 \sqsubseteq C_2$ and assertions $A(c)$, $R(c, d)$

Possible extensions:

- $\bot$ (to express disjoint classes)
- domain restrictions $\mathrm{dom}(R) \sqsubseteq C$
- range restrictions $\mathrm{range}(R) \sqsubseteq C$
- role inclusions $R_1 \circ \ldots \circ R_n \sqsubseteq R_{n+1}$ (transitivity: $R \circ R \sqsubseteq R$)

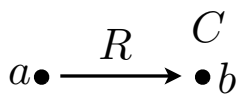
OWL 2 EL includes all these extensions.

Forward chaining and canonical models

$\mathcal{K} =$

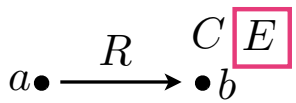
$R(a,b) \quad C(b)$

$C \sqsubseteq E \quad D \sqsubseteq \exists S.(B \sqcap D)$
$D \sqsubseteq \exists R.(A \sqcap D) \quad \exists R.E \sqsubseteq D$



Forward chaining and canonical models

$\mathcal{K} =$ $R(a,b)$ $C(b)$ | $\boxed{C \sqsubseteq E}$ $D \sqsubseteq \exists S.(B \sqcap D)$ $D \sqsubseteq \exists R.(A \sqcap D)$ $\exists R.E \sqsubseteq D$

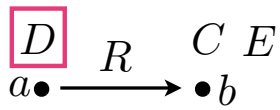


Forward chaining and canonical models

$\mathcal{K} =$ $R(a,b) \quad C(b)$

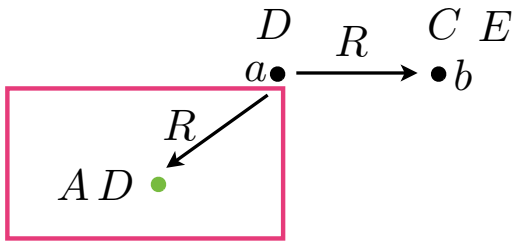
$C \sqsubseteq E \quad D \sqsubseteq \exists S.(B \sqcap D)$

$D \sqsubseteq \exists R.(A \sqcap D) \quad \boxed{\exists R.E \sqsubseteq D}$



Forward chaining and canonical models

$\mathcal{K} =$ $R(a,b)$ $C(b)$ | $C \sqsubseteq E$ $D \sqsubseteq \exists S.(B \sqcap D)$ $D \sqsubseteq \exists R.(A \sqcap D)$ $\exists R.E \sqsubseteq D$

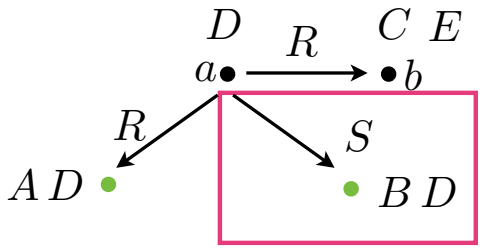


Forward chaining and canonical models

$\mathcal{K} =$ $R(a,b) \quad C(b)$

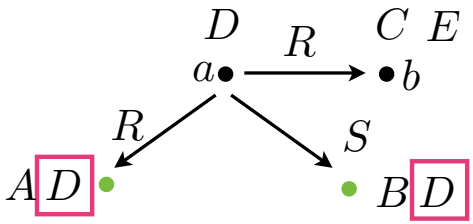
$C \sqsubseteq E \quad \boxed{D \sqsubseteq \exists S.(B \sqcap D)}$

$D \sqsubseteq \exists R.(A \sqcap D) \quad \exists R.E \sqsubseteq D$



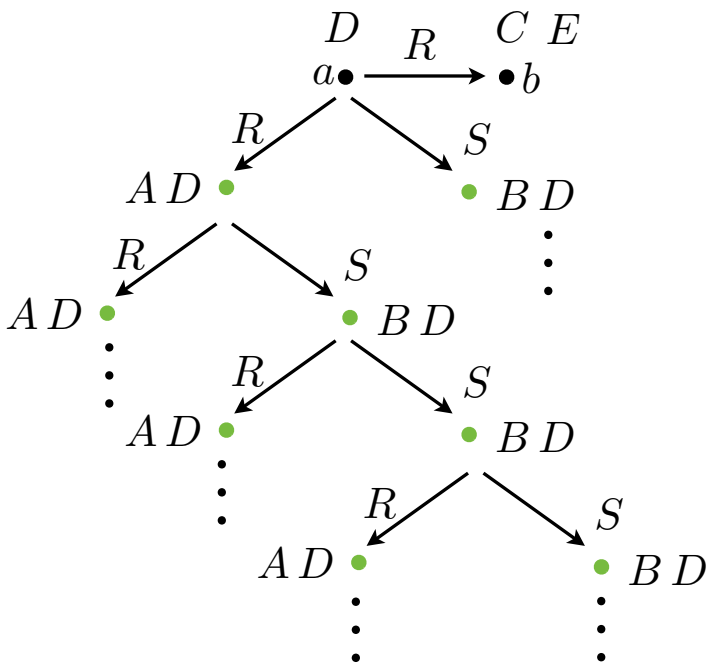
Forward chaining and canonical models

$\mathcal{K} =$ $R(a,b)$ $C(b)$ | $C \sqsubseteq E$ $D \sqsubseteq \exists S.(B \sqcap D)$ $D \sqsubseteq \exists R.(A \sqcap D)$ $\exists R.E \sqsubseteq D$



Forward chaining and canonical models

$\mathcal{K} =$ $R(a,b)$ $C(b)$ | $C \sqsubseteq E$ $D \sqsubseteq \exists S.(B \sqcap D)$ $D \sqsubseteq \exists R.(A \sqcap D)$ $\exists R.E \sqsubseteq D$



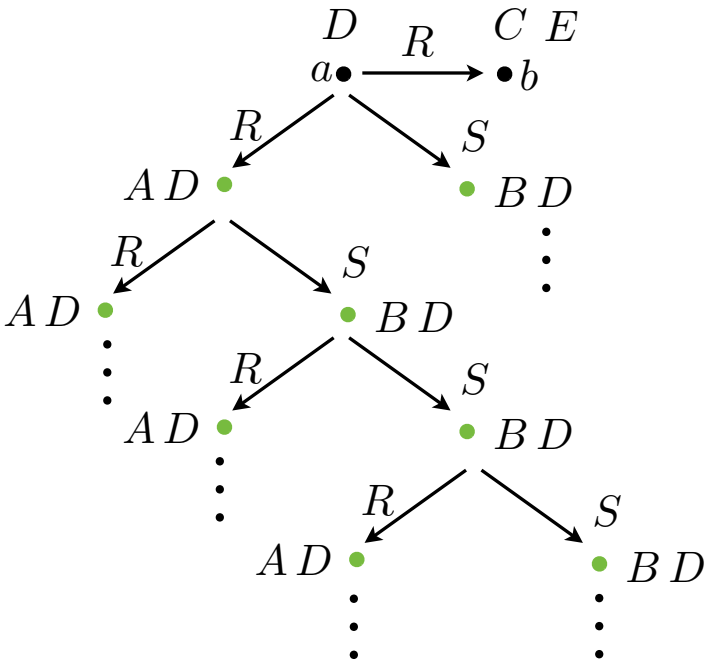
Forward chaining and canonical models

$\mathcal{K}$ =

$R(a,b)$ $C(b)$

$C \sqsubseteq E$ $D \sqsubseteq \exists S.(B \sqcap D)$

$D \sqsubseteq \exists R.(A \sqcap D)$ $\exists R.E \sqsubseteq D$



Exhaustive application of inclusions

Result: **canonical model** $\mathcal{I}_\mathcal{K}$

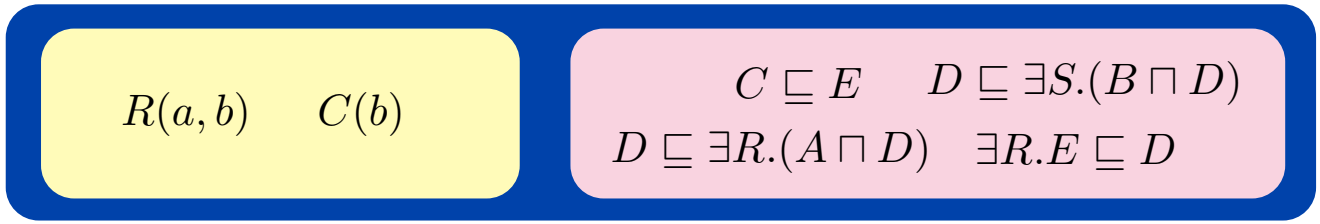
- always gives the right answer to queries

$$\mathcal{I}_\mathcal{K} \models q(\vec{a}) \text{ iff } \mathcal{K} \models q(\vec{a})$$

- may be infinite

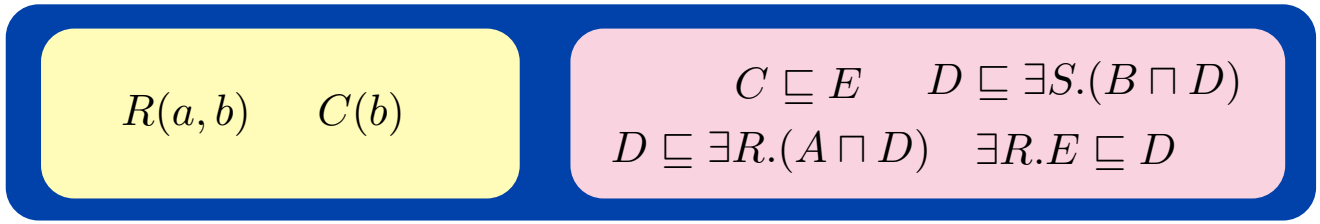
- forest structure (ABox + attached trees)

Compact representation of the canonical model

$\mathcal{K} =$ 

Idea: use the repetitions in $\mathcal{I}_{\mathcal{K}}$
to find a finite representation

Compact representation of the canonical model

$\mathcal{K} =$ 

normalized TBox:

- only atomic concepts behind $\exists$
- conjunction only on the left-hand-side

$$D \sqsubseteq \exists R.(A \sqcap D) \quad \rightsquigarrow \quad D \sqsubseteq \exists R.F \quad F \sqsubseteq A \quad F \sqsubseteq D \quad A \sqcap D \sqsubseteq F$$

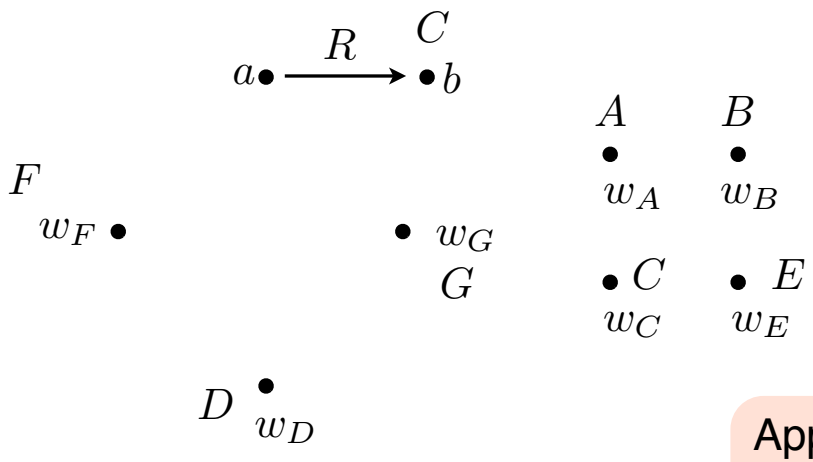
$$D \sqsubseteq \exists S.(B \sqcap D) \quad \rightsquigarrow \quad D \sqsubseteq \exists S.G \quad G \sqsubseteq B \quad G \sqsubseteq D \quad B \sqcap D \sqsubseteq G$$

Compact representation of the canonical model

$\mathcal{K} =$

$R(a,b) \quad C(b)$

$C \sqsubseteq E \quad D \sqsubseteq \exists S.G \quad G \sqsubseteq B \quad G \sqsubseteq D$
$B \sqcap D \sqsubseteq G \quad D \sqsubseteq \exists R.F \quad A \sqcap D \sqsubseteq F$
$F \sqsubseteq A \quad F \sqsubseteq D \quad \exists R.E \sqsubseteq D$



At the start:

- ABox assertions
- an individual w_A with $A(w_A)$ for each atomic concept A

Application of an inclusion on x:

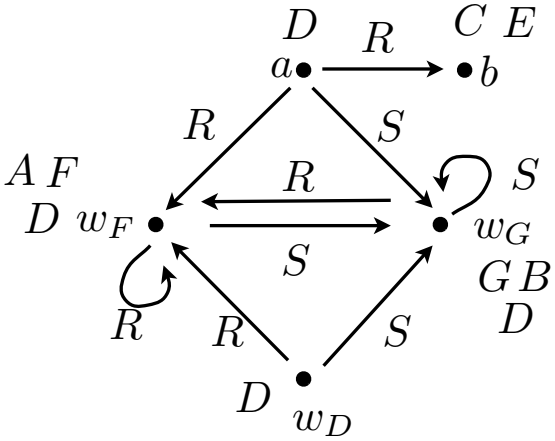
- if $C(x)$ and $C \sqsubseteq A$: add $A(x)$
- if $C(x)$ and $C \sqsubseteq \exists R.A$: add $R(x, w_A)$
- if $C(x), D(x)$ and $C \sqcap D \sqsubseteq A$: add $A(x)$

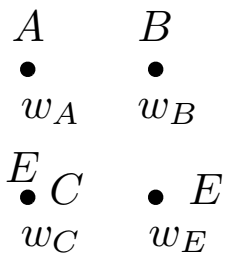
Compact representation of the canonical model

$\mathcal{K} =$

$R(a,b) \quad C(b)$

$C \sqsubseteq E \quad D \sqsubseteq \exists S.G \quad G \sqsubseteq B \quad G \sqsubseteq D$
$B \sqcap D \sqsubseteq G \quad D \sqsubseteq \exists R.F \quad A \sqcap D \sqsubseteq F$
$F \sqsubseteq A \quad F \sqsubseteq D \quad \exists R.E \sqsubseteq D$





Result: $\mathcal{C}_\mathcal{K}$

✔ Subsumption

$\mathcal{K} \models A_1 \sqsubseteq A_2$ ssi $\mathcal{C}_\mathcal{K} \models A_2(w_{A_1})$

✔ Instance queries

$\mathcal{K} \models A_1(c)$ ssi $\mathcal{C}_\mathcal{K} \models A_1(c)$

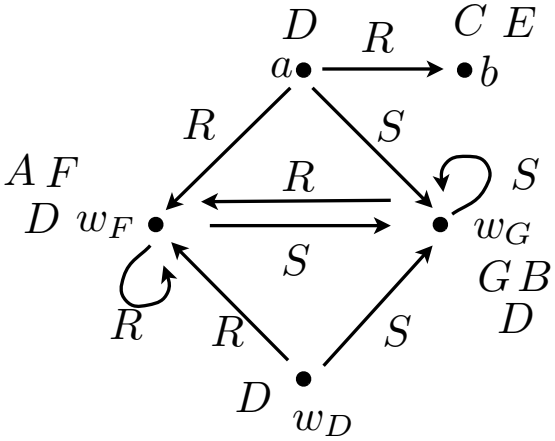
Terminates in polynomial time

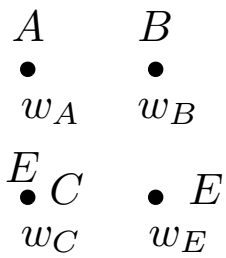
Compact representation of the canonical model

$\mathcal{K} =$

$R(a,b) \quad C(b)$

$C \sqsubseteq E \quad D \sqsubseteq \exists S.G \quad G \sqsubseteq B \quad G \sqsubseteq D$
$B \sqcap D \sqsubseteq G \quad D \sqsubseteq \exists R.F \quad A \sqcap D \sqsubseteq F$
$F \sqsubseteq A \quad F \sqsubseteq D \quad \exists R.E \sqsubseteq D$





Result: $\mathcal{C}_\mathcal{K}$

✔ Subsumption

$\mathcal{K} \models A_1 \sqsubseteq A_2$ ssi $\mathcal{C}_\mathcal{K} \models A_2(w_{A_1})$

✔ Instance queries

$\mathcal{K} \models A_1(c)$ ssi $\mathcal{C}_\mathcal{K} \models A_1(c)$

Terminates in polynomial time

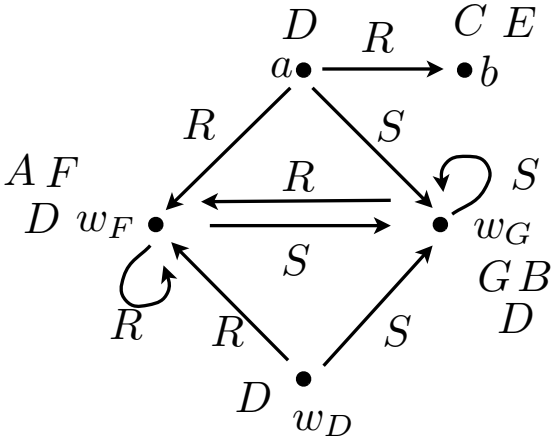
can classify SNOMED in a few seconds !

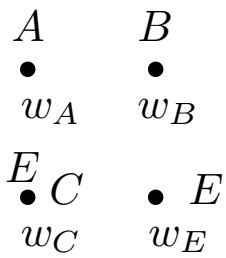
Compact representation of the canonical model

$\mathcal{K} =$

$R(a,b) \quad C(b)$

$C \sqsubseteq E \quad D \sqsubseteq \exists S.G \quad G \sqsubseteq B \quad G \sqsubseteq D$
$B \sqcap D \sqsubseteq G \quad D \sqsubseteq \exists R.F \quad A \sqcap D \sqsubseteq F$
$F \sqsubseteq A \quad F \sqsubseteq D \quad \exists R.E \sqsubseteq D$





Result: $\mathcal{C}_\mathcal{K}$

✔ Subsumption

$\mathcal{K} \models A_1 \sqsubseteq A_2$ ssi $\mathcal{C}_\mathcal{K} \models A_2(w_{A_1})$

✔ Instance queries

$\mathcal{K} \models A_1(c)$ ssi $\mathcal{C}_\mathcal{K} \models A_1(c)$

Terminates in polynomial time

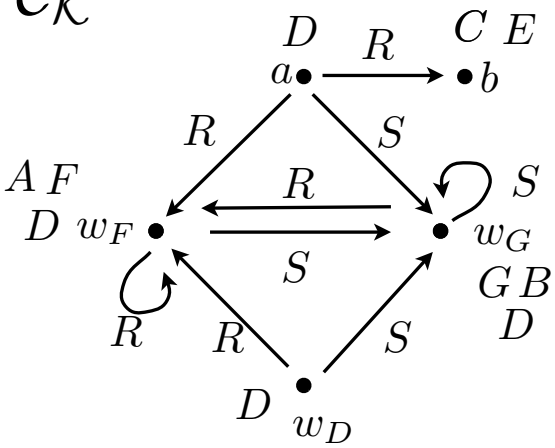
What about conjunctive queries ?

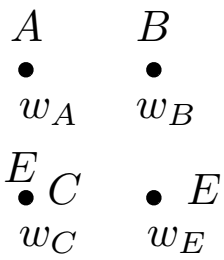
Answering conjunctive queries

$\mathcal{K} =$ $R(a,b) \quad C(b)$

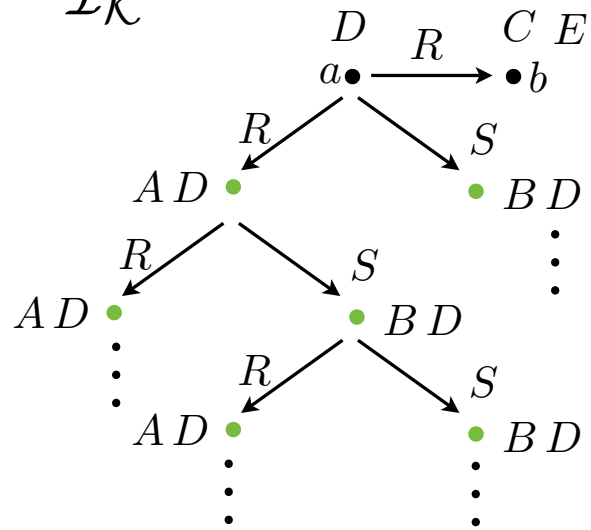
$C \sqsubseteq E \quad D \sqsubseteq \exists S.G \quad G \sqsubseteq B \quad G \sqsubseteq D$
$B \sqcap D \sqsubseteq G \quad D \sqsubseteq \exists R.F \quad A \sqcap D \sqsubseteq F$
$F \sqsubseteq A \quad F \sqsubseteq D \quad \exists R.E \sqsubseteq D$

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





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



$\neq$

answer: yes

$q = \exists x\, R(x,x)$

answer: no

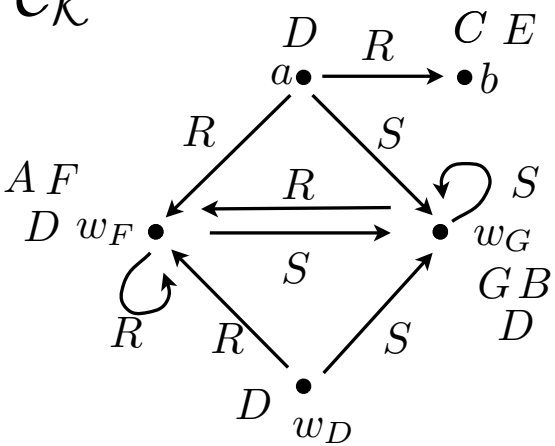
Answering conjunctive queries

$\mathcal{K} =$

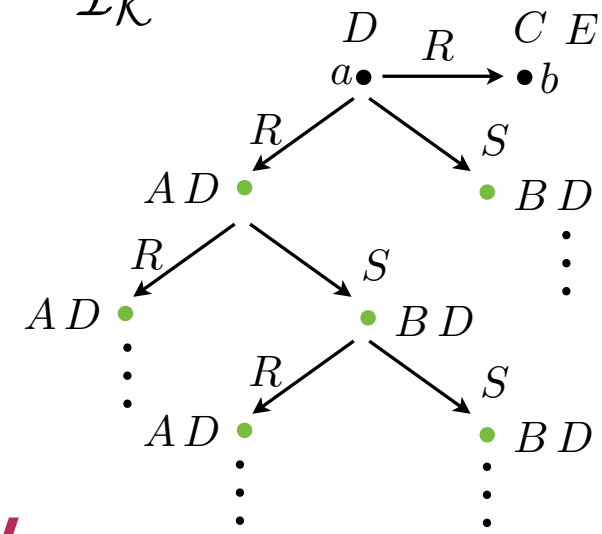
$R(a,b) \quad C(b)$

$C \sqsubseteq E \quad D \sqsubseteq \exists S.G \quad G \sqsubseteq B \quad G \sqsubseteq D$
$B \sqcap D \sqsubseteq G \quad D \sqsubseteq \exists R.F \quad A \sqcap D \sqsubseteq F$
$F \sqsubseteq A \quad F \sqsubseteq D \quad \exists R.E \sqsubseteq D$

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



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



answers: $(a,b), (a,w_F), (w_F,w_F), (w_G,w_F), (w_D,w_F)$

$\neq$

$q(x,y) = D(x) \wedge R(x,y)$

answers: (a,b)

Answering conjunctive queries

Problem: false positives - query matches in $\mathcal{C}_\mathcal{K}$ that do not exist in $\mathcal{I}_\mathcal{K}$

Solution: modify q to prevent such matches

For our examples:

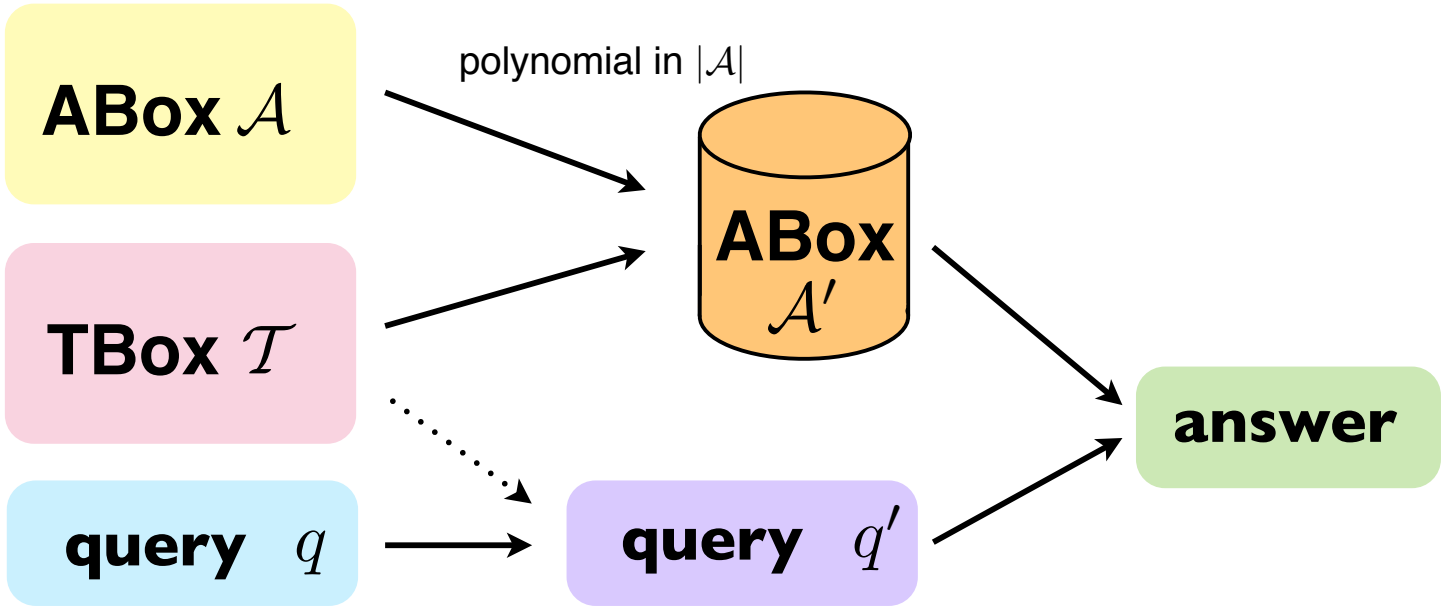
$$\exists x\, R(x,x) \quad \rightsquigarrow \quad \exists x\, R(x,x) \wedge \bigwedge_A (x \neq w_A)$$

$$D(x) \wedge R(x,y) \quad \rightsquigarrow \quad D(x) \wedge R(x,y) \wedge \bigwedge_A (x \neq w_A \wedge y \neq w_A)$$

Remark: rewriting of q is independent of both $\mathcal{T}$ and $\mathcal{A}$

Combined rewriting

The approach we have just seen is called "combined rewriting".



This approach guarantees polynomial data complexity.

Advantage: more widely applicable than "pure" rewriting

Disadvantage: uses more space (if $|\mathcal{A}|$ is big...), modifies the data

Note: combined rewriting also interesting for *DL-Lite*

First-order rewritability in $\mathcal{EL}$

Combined approach requires ability to modify the data

- not always possible / desirable ! (e.g. information integration)

Question: can we identify queries which are FO-rewritable ?

Some first results in this direction [Bienvenu et al., DL'12] for IQs:

- always possible if TBox is acyclic
- for general TBoxes: the problem of deciding FO-rewritability is PSPACE-hard, in EXPTIME
- EXPTIME-hard if ABoxes have restricted signature

Non-uniform complexity analysis: consider specific TBox, query (see [Lutz and Wolter, KR'12] for more on this).

Expressive "lightweight" DLs

Interestingly, much more expressive DLs have polynomial data complexity.

Horn-$\mathcal{SHIQ}$: extends both DL-Lite and $\mathcal{EL}$

- classical reasoning is EXPTIME-complete in combined complexity (like for full $\mathcal{SHIQ}$)
- conjunctive query answering is P-complete in data complexity (like for $\mathcal{EL}$)

New querying algorithm for Horn-$\mathcal{SHIQ}$ [Eiter et. al, AAAI'12] based upon datalog:

- can be seen as rewriting the query using the TBox, then evaluating it over completed ABox

Recap of complexity landscape

	instance queries *combined*	**conjunctive queries** *combined*	*data*
Plain database		NP-complete	in AC_0
DL-Lite	in P	NP-complete	in AC_0
$\mathcal{EL}$	P-complete	NP-complete	P-complete
$\mathcal{ELI}$, Horn-$\mathcal{SH}(\mathcal{O})\mathcal{IQ}$	EXP-complete	EXP-complete	P-complete
Horn-$\mathcal{SR}(\mathcal{O})\mathcal{IQ}$	2EXP-complete	2EXP-complete	P-complete
$\mathcal{ALC}(\mathcal{H})$	EXP-complete	EXP-complete	coNP-complete
$\mathcal{ALCI}$ / $\mathcal{SH}$ / $\mathcal{SHIQ}$	EXP-complete	2EXP-complete	coNP-complete
$\mathcal{SHOIQ}$	NEXP-complete	open	open
$\mathcal{SROIQ}$ (OWL 2)	2NEXP-complete	open	open

Conclusion

Research in DLs has undergone big changes in recent years:

- new application: using ontologies to access data
- conjunctive query answering now a central reasoning task
- focus on new families of tractable DLs (DL-Lite, $\mathcal{EL}$)

Nowadays, complexity landscape quite well understood

- two measures: combined complexity and data complexity
- landscape for CQs more nuanced than for traditional reasoning tasks

Two main techniques used for lightweight DLs:

- query rewriting
- saturation (aka forward-chaining, chase)

Current work and future directions

Remains a lot of do in order to make query answering really practicable:

- more refined complexity analysis (beyond data complexity)
  - complexity of query rewriting [Kikot et al., DL'11, KR'12], non-uniform complexity [Lutz & Wolter, KR'12]
- database-style optimizations
  - semantic indexing - Quest [Rodriguez-Muro & Calvanese, ISWC'11, KR'12], query minimization [Bienvenu et al., KR'12]
- benchmarks for testing algorithms sorely lacking !
- what about more expressive query languages?
  - regular path queries [Bienvenu et al., DL'12], CQs extended with negation or inequalities (cf. [Rosati '07])
- querying inconsistent data, cf. [Rosati, IJCAI'11] [Bienvenu, AAAI'12]

Bibliography

References: Intro to DLs

Textbook

The Description Logic Handbook: Theory, Implementation and Applications. Edited by Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider. Cambridge University Press (2003).

However, most of what was covered in the talk isn't in this book.

References: DL-Lite family

- Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati. *Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family*. Journal of Automated Reasoning 39(3): 385-429 (2007).
- Alessandro Artale, Diego Calvanese, Roman Kontchakov, Michael Zakharyaschev. *The DL-Lite Family and Relations*. Journal of Artificial Intelligence Research 36: 1-69 (2009).
- Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, Michael Zakharyaschev. *The Combined Approach to Query Answering in DL-Lite*. Proceedings of KR (2010).

References: DL-Lite family

- QuOnto A. Acciarri, D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, M. Palmieri, and R. Rosati: *QuOnto: Querying ontologies*. Proceedings of AAAI (2005).
- Requiem H. Perez-Urbina, B. Motik, and I. Horrocks: *Tractable query answering and rewriting under description logic constraints*. Journal of Applied Logic (2010).
- Presto R. Rosati and A. Almatelli: *Improving query answering over DL-Lite ontologies*. Proceedings of KR (2010).
- Rapid Alexandros Chortaras, Despoina Trivela, Giorgos B. Stamou: *Optimized Query Rewriting for OWL 2 QL*. Proceedings of CADE (2011).
- Nyaya G. Gottlob, G. Orsi, and A. Pieris: *Ontological queries: Rewriting and optimization*. Proceedings of ICDE (2011). [Note: for Datalog +/-]

References: DL-Lite family

- S. Kikot, R. Kontchakov, and M. Zakharyaschev: *On (In)Tractability of OBDA with OWL 2 QL*. Proceedings of DL (2011).
- Stanislav Kikot, Roman Kontchakov and Michael Zakharyaschev: *Conjunctive Query Answering with OWL 2 QL*. To appear in KR 2012.
- Georg Gottlob and Thomas Schwentick: *Rewriting Ontological Queries into Small Nonrecursive Datalog Programs*. To appear in KR 2012.
- Mariano Rodriguez-Muro and Diego Calvanese: *High Performance Query Answering over DL-Lite Ontologies*. To appear in KR 2012.

References: $\mathcal{EL}$ family

- Franz Baader, Sebastian Brandt, Carsten Lutz. *Pushing the $\mathcal{EL}$ Envelope*. Proceedings of IJCAI (2005).
- Franz Baader, Sebastian Brandt, Carsten Lutz. *Pushing the $\mathcal{EL}$ Envelope Further*. Proceedings of OWLED (2008).
- Carsten Lutz, David Toman, Frank Wolter. *Conjunctive Query Answering in the Description Logic $\mathcal{EL}$ Using a Relational Database System*. Proceedings of IJCAI (2009).
- Jing Mei, Shengping Liu, Guo Tong Xie, Aditya Kalyanpur, Achille Fokoue, Yuan Ni, Hanyu Li, Yue Pan: *A Practical Approach for Scalable Conjunctive Query Answering on Acyclic $\mathcal{EL}^+$ Knowledge Base*. Proceedings of ISWC (2009).

References: Horn-$\mathcal{SHIQ}$

- Thomas Eiter, Georg Gottlob, Magdalena Ortiz, Mantas Simkus: *Query Answering in the Description Logic Horn-$\mathcal{SHIQ}$*. Proceedings of JELIA (2008).
- Yevgeny Kazakov: *Consequence-Driven Reasoning for Horn-$\mathcal{SHIQ}$ Ontologies*. Proceedings of IJCAI (2009).
- Thomas Eiter, Magdalena Ortiz, Mantas Simkus, TrungKien Tran, Guohui Xiao: *Query Rewriting for Horn-$\mathcal{SHIQ}$ plus Rules*. To appear in AAAI 2012.

References: Expressive DLs

- Magdalena Ortiz, Diego Calvanese, Thomas Eiter: *Characterizing Data Complexity for Conjunctive Query Answering in Expressive Description Logics*. Proceedings of AAAI (2006).
- Birte Glimm, Carsten Lutz, Ian Horrocks, Ulrike Sattler: *Conjunctive Query Answering for the Description Logic $\mathcal{SHIQ}$*. Journal of Artificial Intelligence Research (2008).
- Yevgeny Kazakov: *$\mathcal{RIQ}$ and $\mathcal{SROIQ}$ Are Harder than $\mathcal{SHOIQ}$*. Proceedings of KR (2008).
- Birte Glimm, Sebastian Rudolph: *Status $\mathcal{QIO}$: Conjunctive Query Entailment Is Decidable*. Proceedings of KR (2010).

Miscellaneous references

Non-uniform complexity of query answering:

- Carsten Lutz and Frank Wolter: *Non-Uniform Data Complexity of Query Answering in Description Logics*. To appear in KR 2012.
- Meghyn Bienvenu, Carsten Lutz, and Frank Wolter: *FO-Rewritability in EL: Preliminary Results*. To appear in DL 2012.

Semantic indexing:

- M. Rodriguez-Muro and D. Calvanese: *Semantic index: Scalable query answering without forward chaining or exponential rewritings*. Proceedings of ISWC (2011). Note: also see their KR'12 paper.

Query containment and minimization in DLs:

- Meghyn Bienvenu, Carsten Lutz, and Frank Wolter: *Query Containment in Description Logics Reconsidered*. To appear in KR 2012.

Miscellaneous references

Path queries:

- Meghyn Bienvenu, Magdalena Ortiz, and Mantas Simkus: *Answering expressive path queries over lightweight DL knowledge bases*. To appear in DL 2012.

Negative results for richer query languages:

- Riccardo Rosati: *The Limits of Querying Ontologies*. Proceedings of ICDT (2007).

Inconsistency-tolerant query answering:

- Meghyn Bienvenu: *On the Complexity of Consistent Query Answering in the Presence of Simple Ontologies*. AAAI 2012.
- Riccardo Rosati: *On the Complexity of Dealing with Inconsistency in Description Logic Ontologies*. Proceedings of IJCAI (2011).
- Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, Domenico Fabio Savo: *Inconsistency-Tolerant Semantics for Description Logics*. Proceedings of RR (2010).