Expressing DL-Lite QA with Controlled English

Camilo Thorne

KRDB Research Centre

Bolzano. March 4, 2008.

KRDB Lunch Talk (1)

Motivation

- ▷ Natural language (NL) interfaces to databases and to structured knowledge sources (e.g., KBs) have been proposed to ease access to data.
- ▷ Among these, **controlled language** (CL) interfaces have been proposed.
- ▷ CLs are subsets of NLs (e.g. English) tailored to data access and management.
- ▷ Their main aim is to avoid NL **ambiguity**.

The Problem

- ▷ CL declarations are mapped into an internal representation over which reasoning is performed. After this, a CL answer may be generated.
- ▷ CLs are, e.g., Attempto and Peng, which translate into FOL, via DRT.
- ▷ None of these systems addresses the **computational complexity** of this task.
- ▷ How can ve evaluate this computational complexity?

Ontology Based Data Acess

- ▷ **The background:** Efficient ontology based data access is formally modelled by by the entailment problem:

$$\langle \mathcal{T}, \mathcal{D} \rangle \models q\sigma$$

  where $\langle \mathcal{T}, \mathcal{D} \rangle$ is a DL-Lite knowledge base and $q\sigma$ the grounding of a conjunctive query. This problem is a.k.a. the **query answering** problem (QA) for DL-Lite KBs [Calvanese *et al.*, 2005].

- ▷ **Our aim:** To evaluate the computational complexity of accessing data with CL, we should **express** QA in CL.

Expressing QA in CL

- To express QA in CL we have to go through a number of steps:
  - **(i)** We have to **express** the DL $\mathcal{L}$.
  - **(ii)** We have to **express** the query language $\mathcal{Q}$.
- **How?:** Defining grammars:
  - A grammar G generating a **controlled declarative fragment** L(G) that maps onto $\mathcal{L}$ *modulo* a compositional translation $\phi(.)$.
  - A grammar G' generating a **controlled interrogative fragment** L(G') that maps onto $\mathcal{Q}$ *modulo* a compositional translation $\phi(.)$.
- It should hold that $\phi(\text{L(G)}) \subseteq \mathcal{L}$ and $\phi(\text{L(G')}) \subseteq \mathcal{Q}$. Our CL will be L(G') $\cup$ L(G').

Compositional Translations

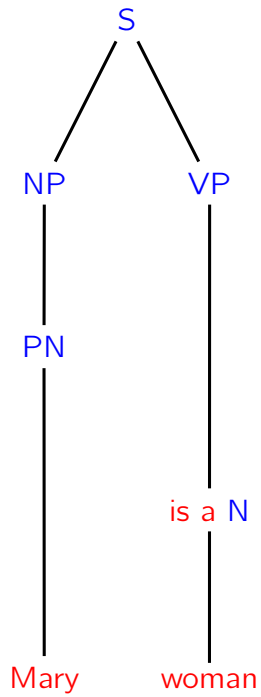
- A **compositional translation** $\phi(.)$ is a function that homeomorphically maps NL utterances onto logical formulas called **meaning representations** (MRs).
- MRs typically belong to either FOL or λ-FOL, the extension of FOL with lambda abstraction, function application and beta reduction (from λ calculus).
- The function $\phi(.)$ is then composed with an interpretation funtion $[\![.]\!]$ for these MRs, thus providing a **formal semantics** for NL, i.e. a function $[\![.]\!]^* := \phi(.) \circ [\![.]\!]$.
- The evaluation of $\phi(.)$ mirrors NL syntactic composition.
- $\phi(.)$ can be hardwired into the CL's grammar.

An Example (1)

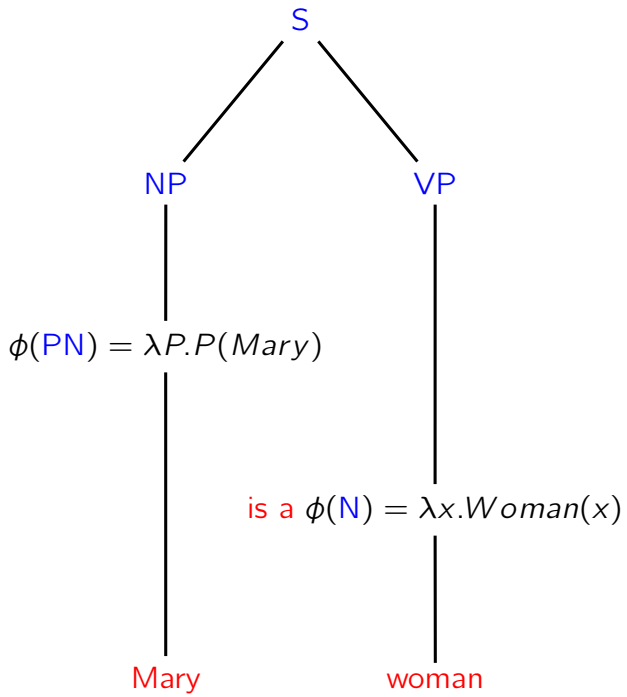
Syntax Rules	MR ($= \phi(.)$)
S $\rightarrow$ NP VP	$\phi(\text{NP})(\phi(\text{VP})) \rhd_\beta \phi(\text{S})$
VP $\rightarrow$ is a N	$\phi(\text{VP}) = \phi(\text{N})$
NP $\rightarrow$ PN	$\phi(\text{NP}) = \phi(\text{PN})$
NP $\rightarrow$ Det N	$\phi(\text{Det})(\phi(\text{N})) \rhd_\beta \phi(\text{NP})$

Lexicon	MR ($= \phi(.)$)
N $\rightarrow$ woman	$\phi(\text{N}) = \lambda x.Woman(x)$
N $\rightarrow$ man	$\phi(\text{N}) = \lambda x.Man(x)$
IV $\rightarrow$ leaves	$\phi(\text{IV}) = \lambda x.Leaves(x)$
PN $\rightarrow$ Mary	$\phi(\text{PN}) = \lambda P.P(Mary)$
Det $\rightarrow$ every	$\phi(\text{Det}) = \lambda P.\lambda Q.\forall x(P(x) \rightarrow Q(x))$

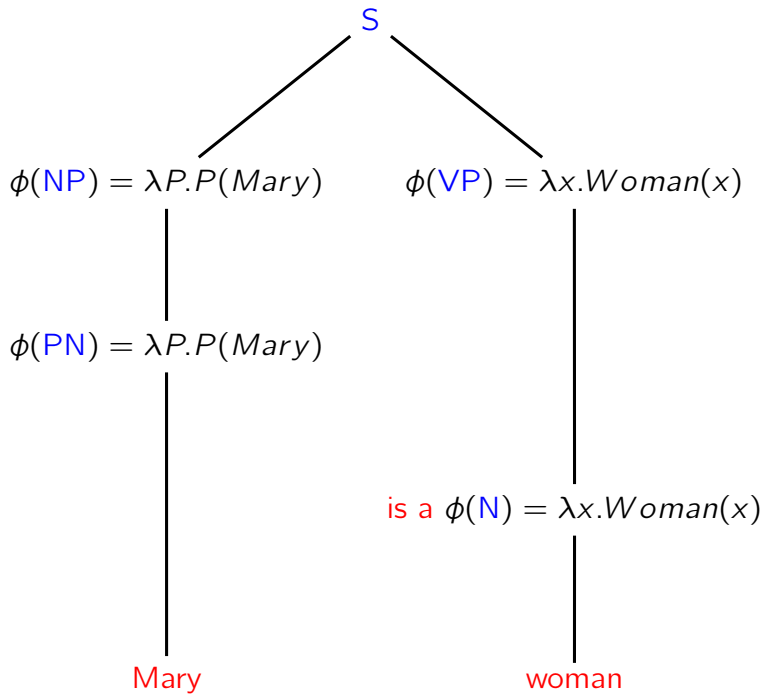
An Example (2)



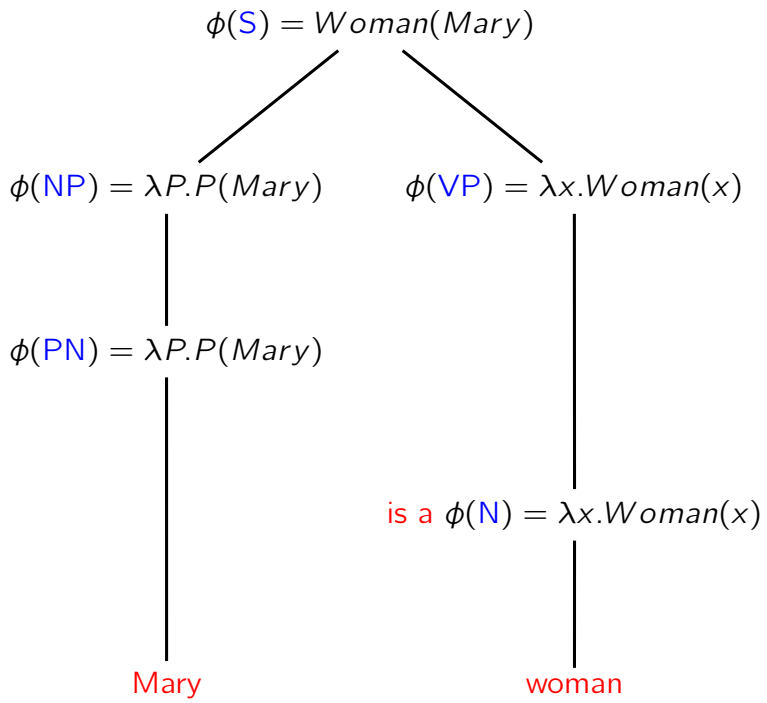
An Example (3)


S
NP
VP
ϕ(PN) = $\lambda P.P(Mary)$
is a ϕ(N) = $\lambda x.Woman(x)$
Mary
woman

An Example (4)


S
ϕ(NP) = $\lambda P.P(Mary)$
ϕ(VP) = $\lambda x.Woman(x)$
ϕ(PN) = $\lambda P.P(Mary)$
is a ϕ(N) = $\lambda x.Woman(x)$
Mary
woman

An Example (5)


$\phi(\text{S}) = Woman(Mary)$
$\phi(\text{NP}) = \lambda P.P(Mary)$
$\phi(\text{VP}) = \lambda x.Woman(x)$
$\phi(\text{PN}) = \lambda P.P(Mary)$
is a $\phi(\text{N}) = \lambda x.Woman(x)$
Mary
woman

Expressing DLs

- ▷ As a first step we need to express DL TBoxes and ABoxes.
- ▷ We have chosen to choose as DL DL-Lite$_{R,\sqcap}$.
- ▷ DL-Lite$_{R,\sqcap}$ offers good a computational complexity for QA (w.r.t. conjunctive queries).
- ▷ The fragment that expresses DL-Lite$_{R,\sqcap}$ KBs is called Lite English.

DL-Lite$_{R,\sqcap}$

- DL-Lite$_{R,\sqcap}$ **knowledge base** KB $= \langle \mathcal{T}, \mathcal{D} \rangle$ is a set of TBox ($\mathcal{T}$) and ABox ($\mathcal{D}$) **assertions**:

$$A(c), R(c_1, c_2) \qquad \text{(ABox assertions)}$$
$$C_l \sqsubseteq C_r \qquad \text{(TBox concept inclusion assertions)}$$
$$R_1 \sqsubseteq R_2 \qquad \text{(TBox role inclusion assertions)}$$

- DL-Lite$_{R,\sqcap}$ left (C_l) and right (C_r) hand side **concepts** are then defined as follows:

$$C_l ::= A \mid \exists R \mid \exists R^- \mid C_l \sqcap C_l.$$
$$C_r ::= C_l \mid \neg A \mid \neg \exists R \mid \neg \exists R^- \mid \exists R.C_r \mid \exists R^-.C_r \mid C_r \sqcap C_r.$$

- It is equipped with standard FOL semantics. TBox reasoning is in **P** [Calvanese *et al.*, 2005].

Lite English (1)

▷ **Lite English** [Bernardi, Calvanese and Thorne, 2007] is a fragment of English that compositionally translates into DL-Lite$_{R,\sqcap}$ KB assertions. Utterances respect the pattern:

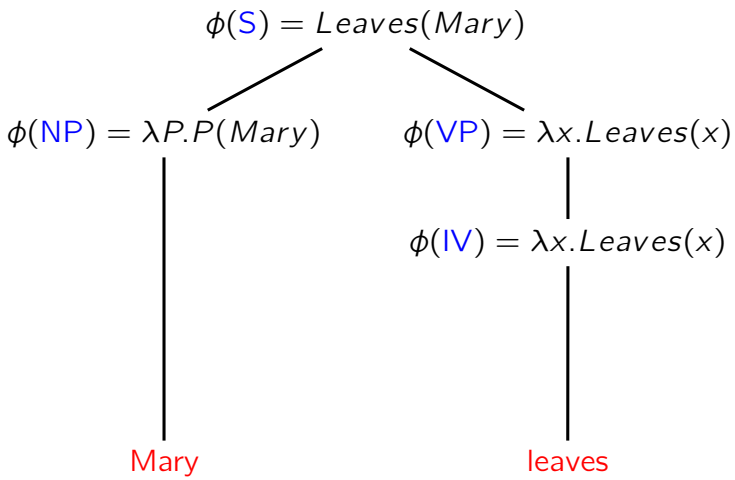
Det N VP

▷ **Dets** are universal quantifiers (conveying subsumption), the Ns a C_l concept and the VPs a C_r concept, for instance:

Category	DL-Lite$_{R,\sqcap}$	Lite English	MR
Det		every	$\lambda P.\lambda Q.\forall x(P(x) \rightarrow Q(x))$
N	C_l	man that sleeps	$\lambda x.Man(x) \wedge Sleeps(x)$
VP	C_r	loves some woman who works	$\lambda y.\exists y(Loves(x,y) \wedge Woman(y)) \wedge Works(y))$

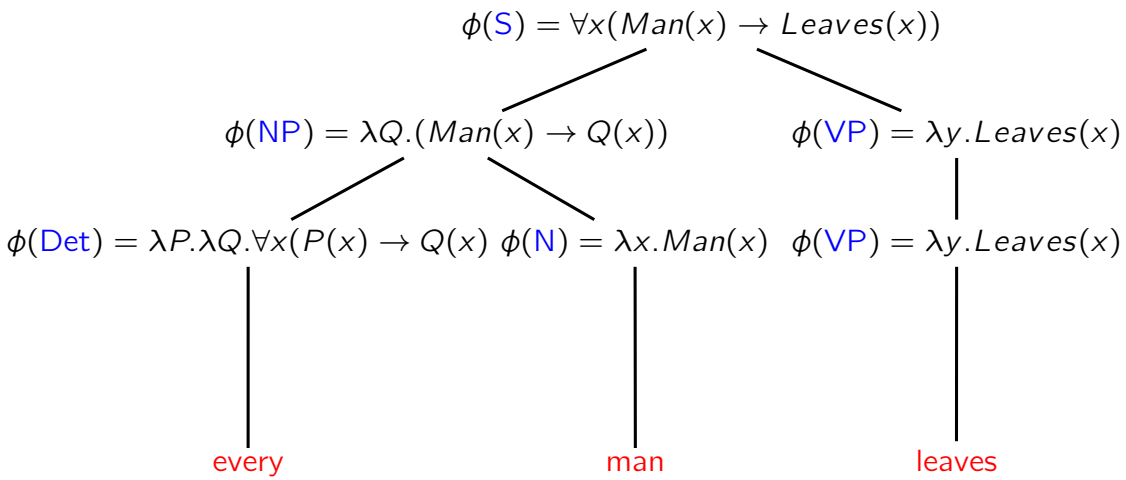
Lite English (2)

- Lite English can express DL-Lite$_{R,\sqcap}$ KBs, and thus inherits of all the properties of this DL.
- We can express ABox assertions like $Leaves(Mary)$:



Lite English (3)

▷ We can also express TBox assertions like $Man \sqsubseteq Leaves$:



▷ The fact that DL-Lite$_{R,\sqcap}$ has boolean conjunction both in left and right hand side concepts allows for restricted **relative** pronouns. **Negation** is restricted to VPs. **Quantifiers** occur in a fixed order.

Relative Expressive Power of Lite English

- We want to understand why Lite English behaves so well.
- For this, we have to compare it with other fragments of English that behave differently w.r.t. computational complexity.
- [Pratt and Third, 2005] define a family of fragments of English and study them w.r.t. the expressive power of their MRs and the computational complexity of satisfiability and entailment among these MRs.

Pratt & Third - COP

- Given the following lexicon:

Copula	=	is a		
Negation	=	is not a		
Determiners/Quantifiers	=	some	every	no
Nouns	=	man, mortal	...	
Proper Names	=	Socrates	...	

- We can build sentences of the following form:

Every man is a mortal

Socrates is a man

Socrates is a mortal.

- We have thus captured the entailment:

$$\{\forall x(Man(x) \rightarrow Mortal(x)), Man(Socrates)\} \models Mortal(Socrates)$$

Pratt & Third - Extending COP

- New fragments are built by extending COP with:

  TV = Transitive verbs, e.g. "reads" (binary relations).
  DTV = Distransitive verbs, e.g., "gives" (ternary relations).
  REL = Relative pronouns, e.g., "who", "that", "which", etc. (conjunction).

- which yield new FOL sets of MRs:

Sentence	MR	Fragment
Every man who is not dead is alive	$\forall x(Man(x) \wedge \neg Dead(x) \rightarrow Alive(x))$	COP+REL
Every soldier defends a country	$\forall x((Soldier(x) \rightarrow \exists y(Country(y) \wedge \text{Defends(x, y)}))$.	COP+TV
Every salesman sells some merchandise to some customer	$\forall x(Salesman(x) \rightarrow \exists y(Customer(y) \wedge \exists z(Merchandise(z) \wedge \text{Sells(x, y, z)}))$.	COP+DTV

Pratt & Third 2005 - Complexity

Fragment	Decision class for satisfiability
COP	**P**
COP+TV+DTV	**P**
COP+REL	**NP-Complete**
COP+REL+TV	**EXPTIME-Complete**
COP+REL+TV+DTV	**NEXPTIME-Complete**

▷ The addition of **relatives** is computationally expensive. Relatives in NL convey boolean **conjunction**. Hence, when combined with boolean **negation** we get a fragment that is propositionally complete, therefore, intractable.

DL-Lite$_{R,\sqcap}$ is as Expressive as COP

▷ **THEOREM** DL-Lite$_{R,\sqcap}$ is as expressive as COP.

(i) COP expresses containment among sets. DL-Lite$_{R,\sqcap}$ can express this via inclusion assertions of the form $A \sqsubseteq B$.

(ii) COP expresses disjointness among sets. DL-Lite$_{R,\sqcap}$ can express this via disjointness assertions of the form $A \sqsubseteq \neg B$.

(iii) COP expresses that an individual belongs to a set or its complement. DL-Lite$_{R,\sqcap}$ can express this via disjointness assertions and ABox assertions.

(iv) COP expresses that the intersection of two sets is not empty. We can express this in DL-Lite$_{R,\sqcap}$ by the ABox assertions $P(c)$ and $Q(c)$ and by dropping the unique name assumption (UNA).

▷ But the converse is false.

DL-Lite$_{R,\sqcap}$ Overlaps with COP+TV+DTV

▷ **THEOREM** DL-Lite$_{R,\sqcap}$ overlaps in expressive power with COP+TV+DTV, but neither contains the other.

(i) COP+TV+DTV and DL-Lite$_{R,\sqcap}$ contain COP, whence the overlapping.

(ii) ($\Rightarrow$) DL-Lite$_{R,\sqcap}$ role-typing assertions $\exists R \sqsubseteq A$, yield in FOL, when prenexed, skolemized and clausified:

$$\neg R(x, y) \vee A(x).$$

These clauses lie, however, beyond COP+TV+DTV MRs [Pratt and Third, 2005].

DL-Lite$_{R,\sqcap}$ Overlaps with COP+TV+DTV

(ii) ($\Leftarrow$) The COP+TV sentence $\psi = \exists x \forall y (P(x) \wedge Q(x) \rightarrow R(x,y))$ that says "there is a least element" is not closed under **union of chains**. We can build a non well-founded model $\mathcal{M}_\omega$ from the chain $\mathcal{M}_1 \preccurlyeq \mathcal{M}_2 \preccurlyeq \ldots \preccurlyeq \mathcal{M}_i \preccurlyeq \ldots$, for $i \in \mathbb{N}^+$, of models of ψ:

$\mathcal{M}_1$:	$0 \longrightarrow +n$	$(\mathbb{N})$	$\mathcal{M}_1 \models \psi$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathcal{M}_i$:	$-(i-1) \longrightarrow +n$	$(\mathbb{N} \cup \{-1, ..., -(i-1)\})$	$\mathcal{M}_i \models \psi$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathcal{M}_\omega = \bigcup_{i \in \mathbb{N}} \mathcal{M}_i$:	$-n \longleftrightarrow +n$	$(\mathbb{Z})$	$\mathcal{M}_\omega \not\models \psi$

But DL-Lite$_{R,\sqcap}$ is closed under this property, since all of its assertions belong (in FOL) to the $\forall\exists^*$ prefix class.

Relational Queries

- ▷ In order to complete the picture, we must now express queries over DL-Lite$_{R,\sqcap}$ KBs.
- ▷ QA is tractable (**LOGSPACE**) w.r.t. data complexity when we consider as query language conjunctive queries (CQs).
- ▷ We must therefore express CQs in CL.

Conjunctive Queries

- A **simple conjunctive query** (CQ) is an expression of the form:

$$q(\vec{x}) \leftarrow \exists \vec{y} \Phi(\vec{x}, \vec{y}, \vec{c})$$

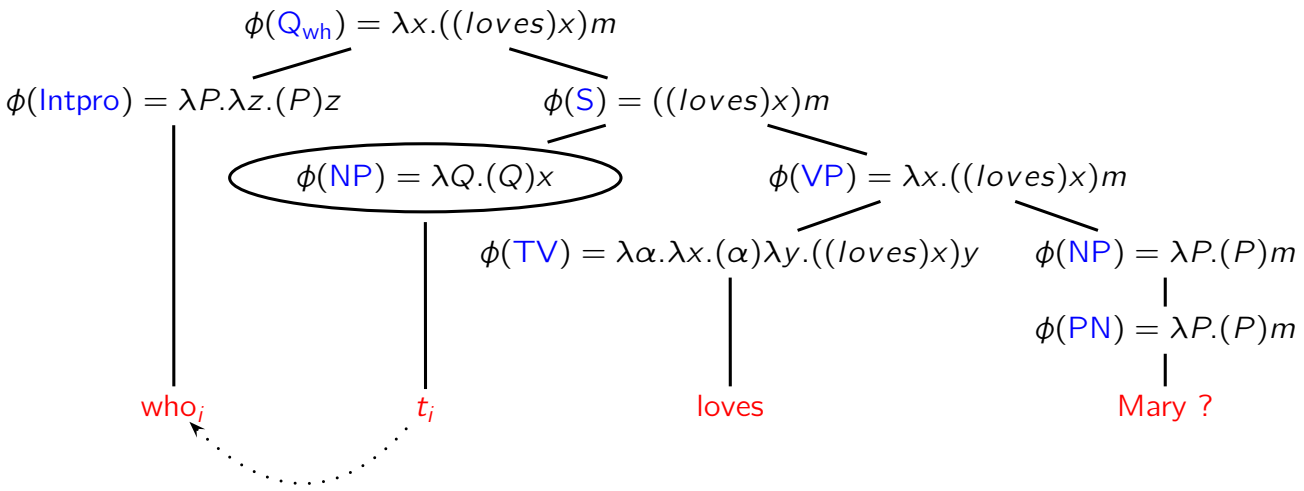
- $\Phi(\vec{x}, \vec{y}, \vec{c})$ is a conjunction of **relational atoms**, called **body**. $\vec{x}$ is a possibly empty finite sequence of **distinguished variables**. If empty, the query is called **boolean**. $\vec{y}$ is a finite sequence of existentially quantified variables.

Expressing CQs

- ▷ Lite Englishes interrogative fragment allows as content words: adjectives, common and proper nouns, intransitive and transitive verbs.
- ▷ As function words we only allow words that express boolean conjunction and existential quantification.

Category	Words	MR ($= \phi(.)$)
Det	some	$\lambda P.\lambda Q.\exists x(P)x \wedge (Q)x$
Pro	somebody, something	$\lambda P.\exists x(P)x$
Coord	and	$\lambda P.\lambda Q.\exists x(P)x \wedge (Q)x$
Relpro	who, that	$\lambda P.\lambda x.(P)x:$
Intpro	which,what,	$\lambda P.\lambda Q.\lambda x[(P)x \wedge (Q)x]$
Intpro	who	$\lambda P.\lambda x.(P)x$

Expressing CQs - Example



CQs Expressible by Lite English

- ▷ Lite English expresses some subclasses of CQs.
- ▷ It can express linear CQs and tree-shaped CQs that are unary, whether boolean or not.
- ▷ We are currently working in covering a wider class, i.e., of n-ary CQs over relations of arity $n \leq 2$.
- ▷ Join variables in the queries correspond to **dependencies** among syntactic components.

Containment and Equivalance

- We must also deal with the problem of optimizing CQ evaluation over DL-Lite KBs.
- This involves dealing with containment and equivalence.
- We say that a CQ q **contains** a CQ q' w.r.t. a (fixed) TBox $\mathcal{T}$ whenever, for every ABox $\mathcal{A}$ it holds that:

$$q(\langle \mathcal{T}, \mathcal{D} \rangle) \subseteq q'(\langle \mathcal{T}, \mathcal{D} \rangle).$$

- We say that two CQs q and q' are **equivalent** w.r.t. a (fixed) TBox $\mathcal{T}$ whenever they contain each other.

Aggregate Queries (1)

- Questions about aggregates are very frequent in specific settings like, e.g. geographical DBs, like GEOBASE:

  What is the largest river in Washington state?

  What is the population of Dallas?

  Which state has the lowest population density?

- Out of its 880 questions we found that:

	CQs	Aggregations	Negation
Wh-questions	34.54%	65.35%	0.11%

- We want to handle also **numerical data**.

Aggregate Queries (2)

▷ The class of **conjunctive aggregate queries** (CQs + agg) that extends that of conjunctive queries is constituted by queries of the form:

$$q(\vec{x}, \vec{\alpha}(\vec{y})) \leftarrow \exists \vec{z}(\Phi(\vec{x}, \vec{y}, \vec{z}, \vec{c}) \wedge \Psi(\vec{x}, \vec{y}, \vec{z}, \vec{c}))$$

$$\text{SELECT } A_1, ..., A_k, \alpha(B_1), ..., \alpha(B_k)$$
$$\text{FROM } R_1, ..., R_n$$
$$\text{WHERE } C_1 \wedge ... \wedge C_m$$
$$\text{GROUP BY } A_1, ..., A_k$$

▷ The **core** of the aggregate query is the conjunctive query that underlies the aggregate query.

Aggregate Queries (3)

- We have looked at functions **max**, **min**, **sum**, **prod** and **count**.
- We have considered several syntactic restrictions:
  - only **relational** bodies (**Rel**),
  - presence of numerical **constants** (**Cons**),
  - presence of numerical **comparisons** ($\leq$).
- The starting point was the work by [Cohen et al., 2007], on aggregate queries. whose results we extend. We exploit the notion of **core**.

Semantics, Containment and Equivalence

- The **semantics** of a CQ + agg exploits the certain answer semantics of its core:

$$q(\langle \mathcal{T}, \mathcal{D} \rangle) := \{(\sigma(\vec{x}).\mathbf{agg}(\vec{y})) | \langle \mathcal{T}, \mathcal{D} \rangle \models \beta(q)\sigma\}.$$

- As before, $\langle \mathcal{T}, \mathcal{D} \rangle$ is a DL-Lite KB.
- We consider also QA over **totally ordered domains**, e.g., $\mathbb{Z}$ or $\mathbb{Q}$.
- Equivalence and containment are defined as for CQs.
- We denote QA the associated decision problem.

Reduction to Core Equivalence

▷ **THEOREM** Query equivalence (under set semantics) for CQs + agg is reducible to core equivalence whenever numerical constants and comparisons do not occur simultaneously:

	Rel	**Rel + Cons**	**Rel + $\leq$**	**Rel + $\leq$ + Cons**
count(1)	Yes	Yes	Yes	No
sum	Yes	Yes	Yes	No
prod	Yes	Yes	Yes	No

(1) [Cohen et al.,2007].

Complexity of Query Equivalence

▷ **THEOREM** The complexity of query equivalence for CQs + agg is the same as for CQs if numerical constants and comparisons are not both present:

	Rel + Cons	**Rel + $\leq$**
max/**min**(1)	**NP-Complete**	Π_P^2-**Complete**
count(1)	**NP-Complete**	Π_P^2-**Complete**
sum	**NP-Complete**	Π_P^2-**Complete**
prod	**NP-Complete**	Π_P^2-**Complete**

(1) [Cohen et al., 2007].

QA for DL-Lite TBoxes

- ▷ **THEOREM** QA for CQs + **Rel** w.r.t. DL-Lite$_{R,\sqcap}$ TBoxes is **LOGSPACE** in data complexity.
- ▷ **THEOREM** QA for CQs + **Rel** + $\leq$ w.r.t. DL-Lite$_{R,\sqcap}$ TBoxes and over linearly ordered domains is **Co-NP-hard** in data complexity. Inclusion assertions (IAs) and linear finite linear orders yield a **Co-NP** upper bound:

Ordering	**Rel** + $\leq$ + IAs
TO (TOLG,TOL,TOG)	**Co-NP-complete**
TDis (TDisLG,TDisL,TDisG)	**Co-NP-complete**
TPO (TPOLG,TPOL,TPOG)	**Co-NP-complete**

Further Work

- ▷ Express aggregate conjunctive queries (CQs + agg) with the SUM, COUNT, MIN and MAX aggregate operators of SQL.
- ▷ Look at containment and equivalence w.r.t. a TBox and at QA (w.r.t. DL-Lite KBs) for this class of queries.
- ▷ Extend the coverage of Lite English by paraphrases (satisfaction-preserving rewritings). This should be possible, since both CQs and the DL-Lite family of logics are decidable.

THANKS FOR YOUR PATIENCE!