The Data Complexity of the Syllogistic Fragments

Camilo Thorne, Diego Calvanese

KRDB Research Centre

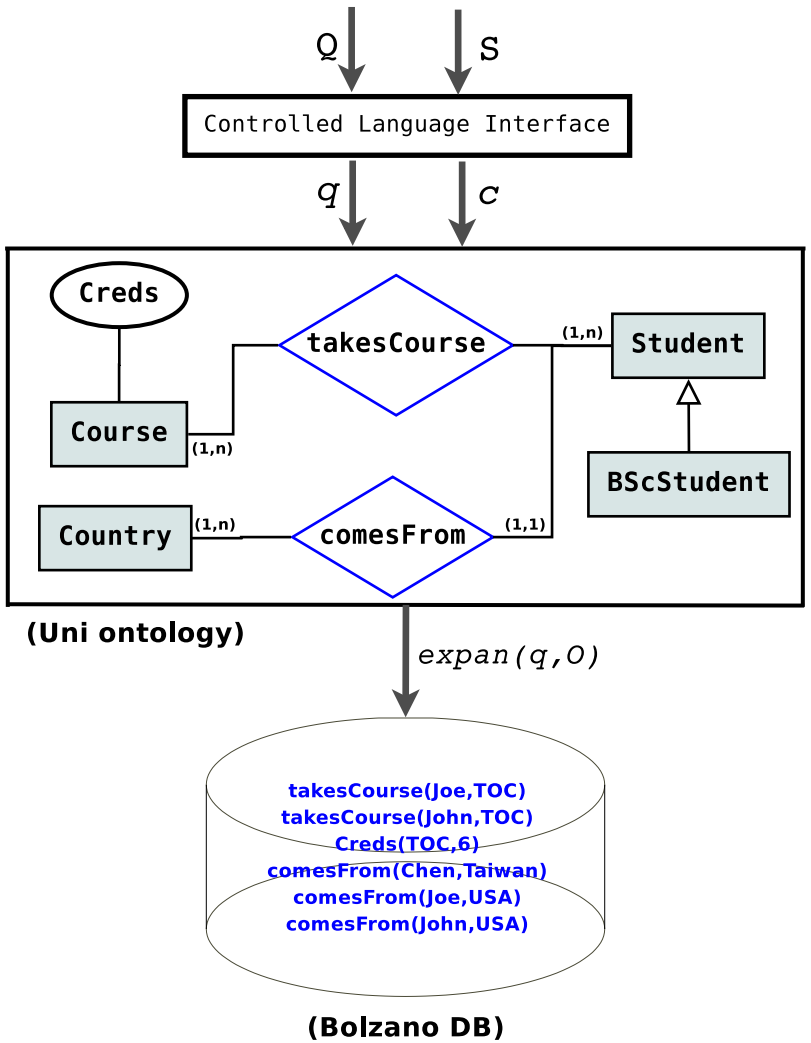
Free University of Bozen-Bolzano

2009 Amsterdam Colloquium

FOEs and CL Interfaces

- Controlled language interfaces (CLIs) to information systems have been proposed to address the usability problem [Sowa 2004, Fuchs et al. 2006]
- A fragment of English (FOE) is an ambiguity-free subset of English which
  - polynomially translate into logic formulas or meaning representations (MRs)
  - the translation can be modelled by formal semantics compositional translations
- More recently, FOEs and CLIs have been proposed for ontology-based data access systems (OBDASs) [Bernstein et al. 2005, Schwitter2006]
- Being ambiguity-free entails good processing, but less is known about other combinatorial properties

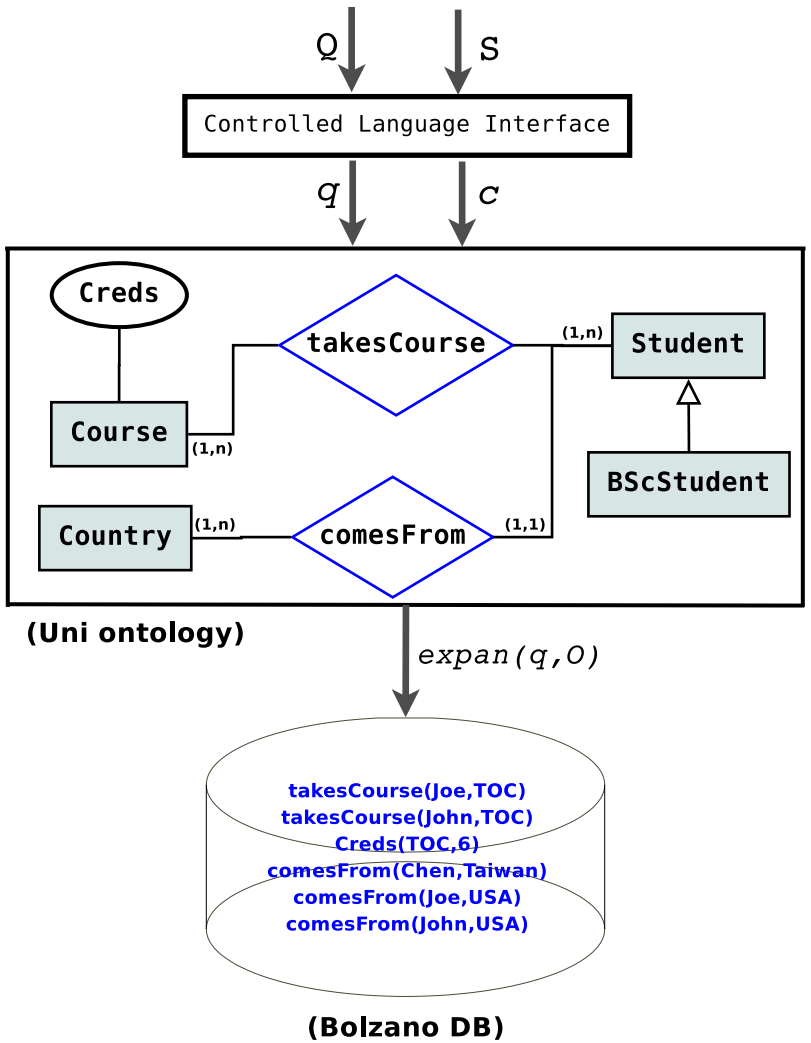
CLIs and OBDASs



An ontology based data access system a pair $(\mathcal{O}, \mathcal{D})$
where data
– is stored in databases $\mathcal{D}$
– is queried through an ontology $\mathcal{O}$

If the system is consistent:
(i) queries are expanded w.r.t. $\mathcal{O}$
(ii) expanded queries are evaluated on $\mathcal{D}$
(iii) answer tuples are returned

CLIs and OBDASs



An ontology based data access system a pair $(\mathcal{O}, \mathcal{D})$
where data
– is stored in databases $\mathcal{D}$
– is queried through an ontology $\mathcal{O}$

If the system is consistent:
(i) queries are expanded w.r.t. $\mathcal{O}$
(ii) expanded queries are evaluated on $\mathcal{D}$
(iii) answer tuples are returned

$\Rightarrow$ Add a CLI front-end?

Scalability of FOEs

- An important measure of performance/scalability of OBDASs is data complexity [Vardi 1982]
- Modulo $\tau(\cdot)$ FOE constructs have an impact on data complexity:
  - which combinations of FOE constructs give way to tractable (**PTIME** or less) data complexity?
  - which combinations of FOE constructs give rise to intractable (**coNP**-hard) data complexity?

Scalability of FOEs

- An important measure of performance/scalability of OBDASs is data complexity [Vardi 1982]
- Modulo $\tau(\cdot)$ FOE constructs have an impact on data complexity:
  - which combinations of FOE constructs give way to tractable (**PTIME** or less) data complexity?
  - which combinations of FOE constructs give rise to intractable (**coNP**-hard) data complexity?
- We can study this impact by looking at the computational properties of their MRs
- Target: semantic complexity [Pratt 2005]

The Syllogistic FOEs

- Proposed by I. Pratt and A. Third
- They capture in English common-sense syllogism (as defined, e.g., by Aristotle):

Every student attends some course	$\forall x(\mathit{Student}(x) \rightarrow \exists y(\mathit{attends}(x,y) \wedge \mathit{Course}(y)))$
Every bachelor student is a student	$\forall x(\mathit{BachelorStudent}(x) \rightarrow \mathit{Student}(x))$
Every bachelor student attends some course	$\forall x(\mathit{BachelorStudent}(x) \rightarrow \exists y(\mathit{attends}(x,y) \wedge \mathit{Course}(y)))$

- They give rise to several fragments of **Fo**

The Syllogistic FOEs

- Proposed by I. Pratt and A. Third
- They capture in English common-sense syllogism (as defined, e.g., by Aristotle):

Every student attends some course	$\forall x(\mathit{Student}(x) \rightarrow \exists y(\mathit{attends}(x,y) \wedge \mathit{Course}(y)))$
Every bachelor student is a student	$\forall x(\mathit{BachelorStudent}(x) \rightarrow \mathit{Student}(x))$
Every bachelor student attends some course	$\forall x(\mathit{BachelorStudent}(x) \rightarrow \exists y(\mathit{attends}(x,y) \wedge \mathit{Course}(y)))$

- They give rise to several fragments of **Fo**
- They can also capture (some) ontology constraints:

Every student attends some course	$\Rightarrow$	cardinality $(1, n)$
Every student comes from some country	$\Rightarrow$	cardinality $(1, n)$
Every bachelor student is a student	$\Rightarrow$	IS-A
Every course is attended by some student	$\Rightarrow$	cardinality $(1, n)$

The Syllogistic FOEs

COP	Copula, common and proper nouns, negation, universal, existential quantifiers
COP+Rel	COP plus relative pronouns
COP+TV	COP plus transitive verbs
COP+TV+DTV	COP+TV plus ditransitive verbs
COP+Rel+TV	COP+Rel plus transitive verbs
COP+Rel+TV+DTV	COP+Rel+TV plus ditransitive verbs
COP+Rel+TV+RA	COP+Rel+TV plus anaphoric pronouns (e.g., he, him, it, herself) of bounded scope
COP+Rel+TV+GA	COP+Rel+TV plus unbounded anaphoric pronouns
COP+Rel+TV+DTV+RA	COP+Rel+TV+DTV plus bounded anaphoric pronouns

The Syllogistic FOEs

COP	Copula, common and proper nouns, negation, universal, existential quantifiers
COP+Rel	COP plus relative pronouns
COP+TV	COP plus transitive verbs
COP+TV+DTV	COP+TV plus ditransitive verbs
COP+Rel+TV	COP+Rel plus transitive verbs
COP+Rel+TV+DTV	COP+Rel+TV plus ditransitive verbs
COP+Rel+TV+RA	COP+Rel+TV plus anaphoric pronouns (e.g., he, him, it, herself) of bounded scope
COP+Rel+TV+GA	COP+Rel+TV plus unbounded anaphoric pronouns
COP+Rel+TV+DTV+RA	COP+Rel+TV+DTV plus bounded anaphoric pronouns

$\Rightarrow$ To study data complexity we consider an interrogative fragment

Tree Shaped Questions [Thorne 2009]

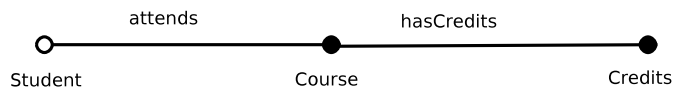
- The interrogative FOE of tree-shaped questions (TSQs) is built with
  - the determiner "some" and the pronouns "something, somebody" (existential)
  - relative pronouns and VP-coordination (conjunction)
  - interrogative pronouns such as "which, what, who," (etc.)
  - copula, transitive verbs, proper and common nouns, passives

Tree Shaped Questions [Thorne 2009]

- The interrogative FOE of tree-shaped questions (TSQs) is built with
  - the determiner "some" and the pronouns "something, somebody" (existential)
  - relative pronouns and VP-coordination (conjunction)
  - interrogative pronouns such as "which, what, who," (etc.)
  - copula, transitive verbs, proper and common nouns, passives

Which student attends some course that weights some credits?

$$\mathit{Student}(x) \wedge \exists y \mathit{takesCourse}(x, y) \wedge \mathit{Course}(y) \wedge$$
$$\exists z(\mathit{hasCredits}(y, z) \wedge \mathit{Credits}(z))$$



- MRs: $\varphi(x) ::= A(x) \mid \exists y R(x, y) \mid \exists y(R(x, y) \wedge \varphi(y)) \mid \varphi(x) \wedge \varphi'(x)$

Consistency, Query Evaluation and Data Complexity

- Semantic complexity: reasoning about FOE facts F, sentences S and questions Q reduces to their **FO** MRs $\tau(F)$, $\tau(S)$ and $\tau(Q)$
- Modulo $\tau(\cdot)$ we consider knowledge bases (KBs) $(\mathcal{S}, \mathcal{F})$
- To understand computational properties, consider $\mathcal{F}$ as input and focus on OBDASs management tasks

  (i) consistency (= **KB-SAT**):

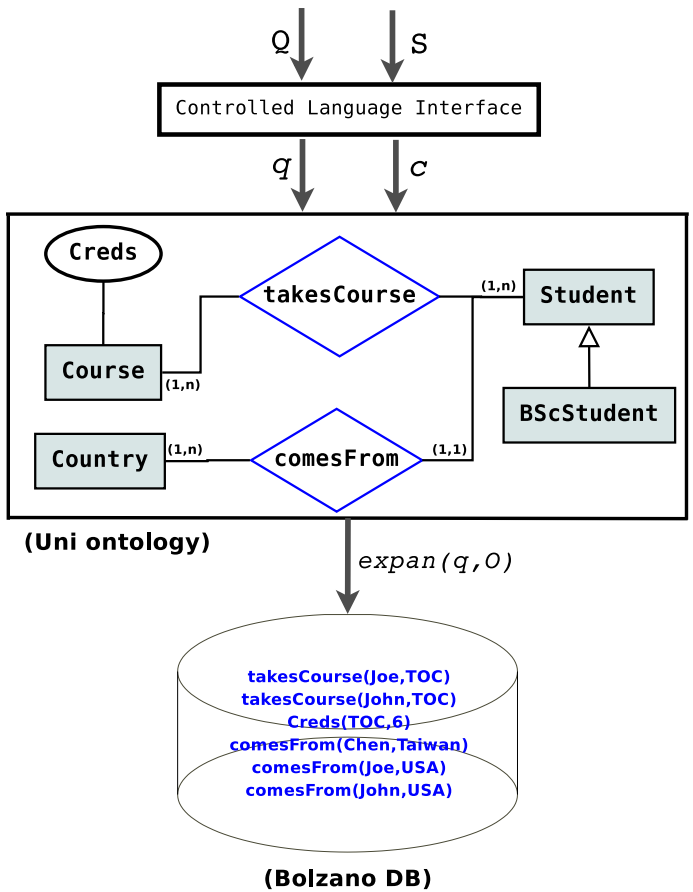
  – is $\tau(\mathcal{S}) \cup \tau(\mathcal{F})$ consistent?

  (ii) query evaluation (= **KB-QA**):

  – does $\tau(\mathcal{S}) \cup \tau(\mathcal{F})$ entail $\tau(Q)[t/x]$?

- Data complexity: we consider $\mathcal{F}$ as the only input, because data outsizes the other parameters [Vardi 1982]!

TSQs and SQL SELECT-PROJECT-JOIN queries



Which student attends some course?

$$Student(x) \wedge \exists y(takesCourse(x, y) \wedge Course(y))$$

```
SELECT Student.SName
  FROM Student,takesCourse,Course
  WHERE takesCourse.SName=Student.SName
    AND takesCourse.CName=Course.CName
```

We get as answers: {Joe, John}

(= entailment holds for these atoms!)

Resolution Saturations

- Define derivability function $\rho(\cdot)$ with rules

$$\text{res}\ \frac{\Gamma, C \vee \overline{L} \qquad \Gamma, C \vee L'}{(C \vee C')\sigma} \qquad \text{fact}\ \frac{\Gamma, C \vee L \vee L'}{(C \vee L)\sigma}$$

- The resolution calculus is a function $\mathcal{R}(\cdot)$ s.t.

$$\mathcal{R}(\Gamma) := \Gamma \cup \rho(\Gamma)$$

- The saturation $\mathcal{R}^\infty(\Gamma)$ of Γ is defined as
  - $\mathcal{R}^0(\Gamma) := \Gamma$
  - $\mathcal{R}^{i+1}(\Gamma) := \mathcal{R}(\mathcal{R}^i(\Gamma))$, for $i > 0$
  - $\mathcal{R}^\infty(\Gamma) := \bigcup\{\mathcal{R}^i(\Gamma) \mid i \geq 0\}$

Resolution Saturations

- Define derivability function $\rho(\cdot)$ with rules

$$\text{res}\ \frac{\Gamma, C \vee \overline{L} \qquad \Gamma, C \vee L'}{(C \vee C')\sigma} \qquad \text{fact}\ \frac{\Gamma, C \vee L \vee L'}{(C \vee L)\sigma}$$

- The resolution calculus is a function $\mathcal{R}(\cdot)$ s.t.

$$\mathcal{R}(\Gamma) := \Gamma \cup \rho(\Gamma)$$

- The saturation $\mathcal{R}^\infty(\Gamma)$ of Γ is defined as
  - $\mathcal{R}^0(\Gamma) := \Gamma$
  - $\mathcal{R}^{i+1}(\Gamma) := \mathcal{R}(\mathcal{R}^i(\Gamma))$, for $i > 0$
  - $\mathcal{R}^\infty(\Gamma) := \bigcup\{\mathcal{R}^i(\Gamma) \mid i \geq 0\}$

- Sound and complete for **Fo** (un)satisfiability, modulo clausification and skolemization, but not a decision procedure!

Resolution Decision Procedures [Joyner 1976]

- Saturations finitely converge when
  - the (functional) depth of terms/litterals is bounded by an $d \in \mathbb{N}$
  - the length of clauses is bounded by $l \in \mathbb{N}$
- Such bounds can be enforced over by using resolution refinements
  - splitting the clauses
  - using A-ordered resolution
  - monadizing

Resolution Decision Procedures [Joyner 1976]

- Saturations finitely converge when
  - the (functional) depth of terms/litterals is bounded by an $d \in \mathbb{N}$
  - the length of clauses is bounded by $l \in \mathbb{N}$
- Such bounds can be enforced over by using resolution refinements
  - splitting the clauses
  - using A-ordered resolution
  - monadizing
- However, the technique works only for specific classes
- Litteral/term depth can be bounded when we resolve covering litterals
- The refined resolution calculi decide the $\mathcal{S}^+$ class of clauses [Leitsch 2003]:
  - every litteral (positive or negative) contains at most one variable
  - or else, it contains all the variables of the clause

A-order and Clause Splitting

- With an acceptable order like $\prec_d$ (over covering clauses) depth bound d can be inferred

$$L \prec_d L \Leftrightarrow_{df} \begin{cases} d(L) < d(L'), \\ V(L) \subseteq V(L'), \text{ and} \\ d(x, L) < d(x, L), \text{ for all } x \in V(L) \end{cases}$$

- With the splitting rule length bound l can be inferred:

$$\text{split } \frac{\Gamma, C \vee L \vee L' \qquad \begin{array}{c} \Gamma, C \vee L \\ \vdots \\ C'\sigma \end{array} \qquad \begin{array}{c} \Gamma, C \vee L' \\ \vdots \\ C'\sigma \end{array}}{C'\sigma} \; (V(L) \cap V(L') = \emptyset)$$

A-order and Clause Splitting

- With an acceptable order like $\prec_d$ (over covering clauses) depth bound d can be inferred

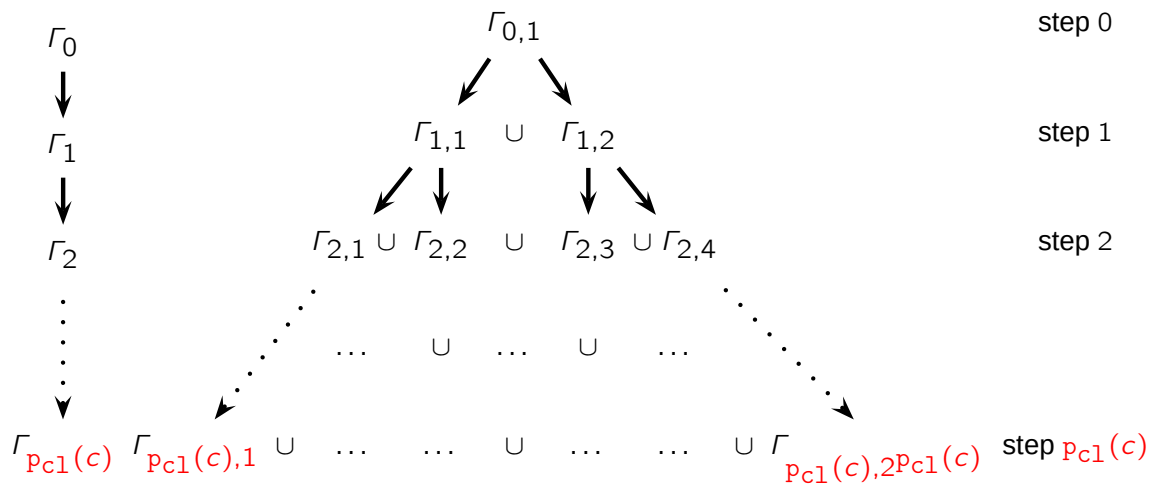
$$L \prec_d L \Leftrightarrow_{df} \begin{cases} d(L) < d(L'), \\ V(L) \subseteq V(L'), \text{ and} \\ d(x, L) < d(x, L), \text{ for all } x \in V(L) \end{cases}$$

- With the splitting rule length bound l can be inferred:

$$\text{split } \frac{\Gamma, C \vee L \vee L' \qquad \begin{matrix} \Gamma, C \vee L \\ \vdots \\ C'\sigma \end{matrix} \qquad \begin{matrix} \Gamma, C \vee L' \\ \vdots \\ C'\sigma \end{matrix}}{C'\sigma} \; (V(L) \cap V(L') = \emptyset)$$

- Refinements are sound and complete w.r.t. (un)satisfiability and decide $\mathcal{S}^+$ when combined with monadization [Joyner 1976]

Data Complexity of Saturations



- If the signature of a set Γ of clauses is finite, and a a bound depth d and a length bound l exist, $\mathcal{R}^{\infty}(\Gamma)$ finitely converges
- If Γ has c constants, convergence is reached after $\mathtt{p_{cl}}$ (without splitting) or $2^{\mathtt{p_{cl}}(c)}$ (with splitting) steps
- By making polynomially many guesses, we check for Γ's (un)satisfiability in non deterministic polynomial time on c

Consistency: membership in NP *for COP+Rel+TV+DTV*

- When clausifying COP+Rel+TV+DTV MRs we get
  - litterals with at most 3 variables
  - unary and binary function terms
  - clauses that are not horn clauses
- But: Pratt and Third show that (un)satisfiablity for a set Γ of such clauses reduces polynomially in $\#(\Gamma)$ to that of a monadic set Γ_u of clauses [Pratt&Third 2005]
- This set Γ_u is a set of $\mathcal{S}^+$ clauses
- Resolution terminates: use the consisteny checking algorithm for $\mathcal{S}^+$
- The reduction plus the consistency checking procedure sketch a non deterministic polynomial time algorithm for **KB-SAT**

Query evaluation: coNP*-hardness for COP+Rel*

A conjunction of 2+2 clauses is a conjunction $\Phi := C_1 \wedge \cdots \wedge C_k$

$$C_i := p_{i1} \vee p_{i2} \vee \neg n_{i1} \vee \neg n_{i2}$$

SAT for 2+2 clauses in **NP**-complete [Scharf, 1994]

Encode Φ into DB $\mathcal{F}_\Phi$

$$\vdots$$

$$c_i \text{ Ps } l_{i,1},\ c_i \text{ Ps } l_{i,2},\ c_i \text{ Ns } n_{i,1},\ c_i \text{ Ns } n_{i,2}$$

$$\vdots$$

$$\text{true is an } A_t$$

Consider $\mathcal{S}_\Phi$

$$\text{no } A_t \text{ is an } A_f \qquad \text{everything that is not an } A_t \text{ is an } A_f$$

Query evaluation: coNP*-hardness for COP+Rel*

A conjunction of 2+2 clauses is a conjunction $\Phi := C_1 \wedge \cdots \wedge C_k$

$$C_i := p_{i1} \vee p_{i2} \vee \neg n_{i1} \vee \neg n_{i2}$$

SAT for 2+2 clauses in **NP**-complete [Scharf, 1994]

Encode Φ into $\mathcal{D}_\Phi$

$$\vdots$$

$$P(c_i, l_{i,1}), P(c_i, l_{i,2}), N(c_i, n_{i,1}), N(c_i, n_{i,2})$$

$$\vdots$$

$$A_t(\text{true})$$

Consider $\mathcal{O}_\Phi$

$$\forall x(A_f(x) \leftrightarrow \neg A_t(x))$$

Query evaluation: coNP*-hardness for COP+Rel*

A conjunction of 2+2 clauses is a conjunction $\Phi := C_1 \wedge \cdots \wedge C_k$

$$C_i := p_{i1} \vee p_{i2} \vee \neg n_{i1} \vee \neg n_{i2}$$

SAT for 2+2 clauses in **NP**-complete [Scharf, 1994]

Ask TSQ Q_Φ

does somebody P_1s something that *Val*s some A_f and
P_2s something that *Val*s some A_f
N_1s something that *Val*s some A_t
N_2s something that *Val*s some A_t?

Query evaluation: coNP*-hardness for COP+Rel*

A conjunction of 2+2 clauses is a conjunction $\Phi := C_1 \wedge \cdots \wedge C_k$

$$C_i := p_{i1} \vee p_{i2} \vee \neg n_{i1} \vee \neg n_{i2}$$

SAT for 2+2 clauses in **NP**-complete [Scharf, 1994]

Ask query ψ_Φ

$$\begin{aligned}
&\exists c \exists l_1 \exists l_2 \exists l_3 \exists l_4 (\\
&P_1(c, l_1) \wedge \exists v_1 (\mathit{Val}(l_1, v_1) \wedge A_f(v_1)) \wedge \\
&P_2(c, l_2) \wedge \exists v_2 (\mathit{Val}(l_2, z_2) \wedge A_f(v_2)) \wedge \\
&N_1(c, l_3) \wedge \exists v_2 (\mathit{Val}(l_3, v_3) \wedge A_t(v_3)) \wedge \\
&N_2(c, l_4) \wedge \exists v_3 (\mathit{Val}(l_4, v_4) \wedge A_t(v_4)))
\end{aligned}$$

Query evaluation: coNP*-hardness for COP+Rel*

A conjunction of 2+2 clauses is a conjunction $\Phi := C_1 \wedge \cdots \wedge C_k$

$$C_i := p_{i1} \vee p_{i2} \vee \neg n_{i1} \vee \neg n_{i2}$$

SAT for 2+2 clauses in **NP**-complete [Scharf, 1994]

We claim

$$\mathcal{O}_\Phi \cup \mathcal{D}_\Phi \not\models \psi_\Phi \text{ iff } \Phi \text{ has no model}$$

Query evaluation: coNP*-hardness for COP+Rel*

A conjunction of 2+2 clauses is a conjunction $\Phi := C_1 \wedge \cdots \wedge C_k$

$$C_i := p_{i1} \vee p_{i2} \vee \neg n_{i1} \vee \neg n_{i2}$$

SAT for 2+2 clauses in **NP**-complete [Scharf, 1994]

"$\Rightarrow$" given a $\mathcal{I}$ s.t.
- $\mathcal{I} \models \mathcal{O}_\Phi$
- $\mathcal{I} \models \mathcal{D}_\Phi$
- $\mathcal{I} \not\models \psi_\Phi$

Idea:

$$\begin{array}{lll} \mathcal{I}, \gamma \not\models \exists v(\mathit{Val}(l, v) \wedge A_f(v)) & \Leftrightarrow & \mathcal{I}, \gamma \models \neg\exists v(\mathit{Val}(l, v) \wedge A_f(v)) \\ & \Leftrightarrow_{\mathcal{O}_\Phi} & \mathcal{I}, \gamma \models \neg\exists v(\mathit{Val}(l, v) \wedge \neg A_t(v)) \\ & \Leftrightarrow & \mathcal{I}, \gamma \models \forall v(\mathit{Val}(l, v) \rightarrow A_t(v)) \end{array}$$

Query evaluation: coNP*-hardness for COP+Rel*

A conjunction of 2+2 clauses is a conjunction $\Phi := C_1 \wedge \cdots \wedge C_k$

$$C_i := p_{i1} \vee p_{i2} \vee \neg n_{i1} \vee \neg n_{i2}$$

SAT for 2+2 clauses in **NP**-complete [Scharf, 1994]

"$\Rightarrow$" given a $\mathcal{I}$ s.t.
- $\mathcal{I} \models \mathcal{O}_\Phi$
- $\mathcal{I} \models \mathcal{D}_\Phi$
- $\mathcal{I} \not\models \psi_\Phi$

Idea:

$$\begin{array}{lll} \mathcal{I}, \gamma \not\models \exists v(\mathit{Val}(I, v) \wedge A_f(v)) & \Leftrightarrow & \mathcal{I}, \gamma \models \neg\exists v(\mathit{Val}(I, v) \wedge A_f(v)) \\ & \Leftrightarrow_{\mathcal{O}_\Phi} & \mathcal{I}, \gamma \models \neg\exists v(\mathit{Val}(I, v) \wedge \neg A_t(v)) \\ & \Leftrightarrow & \mathcal{I}, \gamma \models \forall v(\mathit{Val}(I, v) \rightarrow A_t(v)) \end{array}$$

we partition the domain!

Query evaluation: coNP*-hardness for COP+Rel*

A conjunction of 2+2 clauses is a conjunction $\Phi := C_1 \wedge \cdots \wedge C_k$

$$C_i := p_{i1} \vee p_{i2} \vee \neg n_{i1} \vee \neg n_{i2}$$

SAT for 2+2 clauses in **NP**-complete [Scharf, 1994]

Define truth assignment $\delta(\cdot)$ by putting

$$\delta(l) = \text{true} \Leftrightarrow (l, \text{true}) \in \mathit{Val}^{\mathcal{I}}$$

Query evaluation: coNP*-hardness for COP+Rel*

A conjunction of 2+2 clauses is a conjunction $\Phi := C_1 \wedge \cdots \wedge C_k$

$$C_i := p_{i1} \vee p_{i2} \vee \neg n_{i1} \vee \neg n_{i2}$$

SAT for 2+2 clauses in **NP**-complete [Scharf, 1994]

"$\Leftarrow$": Symmetrical argument

Query evaluation: coNP*-hardness for COP+Rel*

A conjunction of 2+2 clauses is a conjunction $\Phi := C_1 \wedge \cdots \wedge C_k$

$$C_i := p_{i1} \vee p_{i2} \vee \neg n_{i1} \vee \neg n_{i2}$$

SAT for 2+2 clauses in **NP**-complete [Scharf, 1994]

Therefore:

$$\mathcal{O}_\Phi \cup \mathcal{D}_\Phi \not\models \psi_\Phi \text{ iff } \Phi \text{ has a model}$$

Thus: conjunction, universal and existential quantification in queries and full negation in ontologies blows up data complexity!

Query evaluation: membership in coNP *for COP+Rel+TV+DTV*

Idea: Since COP+Rel+TV+DTV is closed under negation, we apply the algorithm for consistency checking

$$(\mathcal{S}, \mathcal{F}) \not\models \tau(\text{does some N VP?}) \Leftrightarrow \mathcal{S} \cup \mathcal{F} \cup \{\tau(\text{no N VP})\} \text{ has a model}$$

Indeed, in **Fo**:

$$\begin{aligned} \Gamma \not\models \exists(\varphi(x) \wedge \psi(x)) \quad &\Leftrightarrow \quad \Gamma \cup \{\neg\exists x(\varphi(x) \wedge \psi(x))\} \\ &\Leftrightarrow \quad \Gamma \cup \{\forall x(\varphi(x) \wedge \neg\psi(x))\} \end{aligned}$$

$\Rightarrow$ When we extend coverage to anaphoric pronouns, all becomes undecidable (reduction from unbounded tiling problem)!

Data Complexity (Summary)

	TSQs	Fragment
COP	in **LogSpace**	in **LogSpace**
COP+TV	in **PTime**	in **LogSpace**
COP+TV+DTV	in **coNP**	in **LogSpace**
COP+Rel	**coNP**-complete [Pratt2009]	in **NP**
COP+Rel+TV	**coNP**-complete [Pratt2009]	**NP**-complete [Pratt2005]
COP+Rel+DTV	**coNP**-complete	**NP**-complete
COP+Rel+DTV+TV	**coNP**-complete	**NP**-complete

	Atomic question	Fragment
COP+Rel+TV+GA	undecidable	undecidable [Pratt2005]
COP+Rel+DTV+TV+RA	undecidable	undecidable [Pratt2005]
COP+Rel+DTV+TV+GA	undecidable	undecidable [Pratt2005]

	TSQs+RA	Fragment
COP+Rel+TV+RA	undecidable	**NP**-complete

Conclusions

- We have proposed to study the scalability to data/data complexity of the syllogistic FOEs + TSQs w.r.t. OBDA
- To study data complexity we have proposed reasoning algorithms based on resolution decision procedures
- We have shown that **KB-QA** w.r.t. TSQs is
  - in **PTIME**: transitive verbs, copula, ditransitive verbs, nouns, every, some, no
  - **coNP**-hard: transitive verbs, copula, (arbitrary) relative pronouns, nouns, no
  - undecidable: we add on top of all the **coNP** function words, (bounded) anaphoric pronouns
- Results are similar for **KB-SAT**