The Data Complexity of the Syllogistic Fragments of English

Camilo Thorne and Diego Calvanese

KRDB Research Centre
Free University of Bozen-Bolzano
4 Via della Mostra, 39100, Italy
{cthorne,calvanese}@inf.unibz.it

Abstract. The syllogistic fragments of English (syllogistic FOEs) express syllogistic reasoning. We want to know how suitable they would be as front-end languages for ontology-based data access systems (OBDASs), front-ends that have been proposed to rely on controlled fragments of natural language. In particular, we want to know how well syllogistic FOE-based data management tasks for OBDASs scale to data. This, we argue, can be achieved by studying the semantic complexity of the syllogistic FOEs and by considering those computational properties that depend on the size of the data alone.

Keywords: Syllogistic fragments of English, tree-shaped questions, ontology-based data access, semantic and data complexity

1 Introduction

A *fragment of English* (FOE) is any (grammatical) subset of English. Montague, back in the 1970's [9] showed how to define a compositional, formal semantics for a FOE by means of *compositional translations* $\tau(\cdot)$ that recursively assign to each English syntactic constituent a HO *meaning representation* (MR), where HO can be conceived of as the extension of FO with the λ-abstraction, λ-application, β-normalization and, eventually, the types of the simply-typed λ-calculus [9]. Since HO (FO) possesses a formal semantics, embodied by an *interpretation function* $\cdot^{\mathcal{I}}$, we can, modulo $\tau(\cdot)$, apply $\cdot^{\mathcal{I}}$ to FOEs. Such formal semantic analysis gives rise to the notion of *semantic complexity*, proposed by Pratt in [11], viz., the computational properties of their MRs (which define fragments of FO) and, a fortiori, the FO reasoning decision problems *expressible* by such FOEs.

An important family of FOEs are the syllogistic FOEs studied by Pratt and Third in [11]. These FOEs capture common-sense syllogistic reasoning, which was (with Aristotle) the starting point of all research in formal logic. The syllogistic FOEs capture also wide classes of common-sense constraints and, as a result, overlap in expressiveness with well-known knowledge representation formalisms such as conceptual modelling (e.g., ER-diagrams) and ontology (e.g., OWL) languages.

Recently [3, 6, 8] FOEs (in particulat, controlled FOEs, viz., fragments devoid of structural or semantic ambiguity) have been proposed as front-end (natural) languages for OBDASs. An OBDAS [13, 4] is a pair $(\mathcal{O}, \mathcal{D})$, where $\mathcal{O}$ is an ontology (a set of,

ultimately, FO axioms) and $\mathcal{D}$ is a database (DB), meant to specify partially the knowledge we have of a given domain (DBs are FO structures). Scalability in OBDASs can be understood through the *data complexity* of data management tasks, i.e., though their (computational) complexity measured w.r.t. the size of $\mathcal{D}$ alone, which is crucial insofar as real-world DBs may contain giga or terabytes of data, if not more [4, 15]. Modulo $\tau(\cdot)$, the semantic complexity of font-end fragments for OBDASs can impact the performance (the scalability to data) of the back-end data management tasks and routines.

In this paper we study the suitability of the syllogistic FOEs as front-end languages for OBDASs by considering their scalability to data. To understand such scalability we study the data complexity of syllogistic FOE-based data management tasks for OBDASs. We focus on the two main OBDAS management tasks, namely, declaring and accessing information, which can be each represented, accordingly, by a FO decision problem: *(i)* knowledge base satisfiability and *(ii)* query evaluation. To infer such data complexity bounds we adopt as main strategy *resolution-based saturation decision procedures* for fragments of FO as outlined by Joyner in [7].

2 The Fragments of English and Tree-Shaped Questions

The syllogistic FOEs are defined incrementally. The idea is to start with a FOE, called COP, that covers: *(i)* copula ("is"), *(ii)* verb-phrase negation ("is not"), *(ii)* the determiners "some", "every" and "no", together with common and proper nouns. The fragment and the translation $\tau(\cdot)$ are defined at the same time, by means of a semantically annotated context-free grammar. Standard HO MRs are used. Thereafter, by extending coverage to a new English construct, viz., transitive verbs (e.g., "likes"), ditransitive verbs (e.g., "gives"), relatives (e.g., "that") and anaphors (e.g., "him"), the other members of the family are defined. See Table 1. For the detailed definition of the fragments, we send the reader to [11]. See Table 2 for their MRs.

The information that we can express/store in such fragments can be queried/accessed by questions. A relevant interrogative FOE is that of *tree shaped questions* (TSQs), which express some of the most common queries to relational databases (which intersect with `SELECT-PROJECT-JOIN` SQL queries [1]), while remaining quite natural for speakers. They are built through query words (e.g., "who"), relatives, transitive verbs, copula, common nouns, the determiner "some", the pronoun "somebody", passives (e.g., "is loved by") and conjunction ("and"). See Table 1. For their formal definition we send the reader to [14]. See Table 2 for their MRs.

We intend to understand the computational properties of the syllogistic FOEs *in the size of the data*. We consider sets $\mathcal{S}$ of quantified and $\mathcal{F}$ of ground sentences. The pair $(\mathcal{S}, \mathcal{F})$ is a KR *knowledge base* (KB). Notice that, modulo $\tau(\cdot)$, $\mathcal{S}$ maps into ("expresses") and ontology $\mathcal{O}$ and $\mathcal{F}$ into a DB $\mathcal{D}$, and thus a KB $(\mathcal{S}, \mathcal{F})$ into an OBDAS $(\mathcal{O}, \mathcal{D})$. We study two decision problems. On the one hand, KB satisfiability (KB-SAT):

- **Given:** $(\mathcal{S}, \mathcal{F})$.
- **Check:** is $\tau(\mathcal{S}) \cup \tau(\mathcal{F})$ satisfiable?

And, on the other hand, query answering (KB-QA):

- **Given:** $(\mathcal{S}, \mathcal{F})$, a question Q and (possibly) a constant c.

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
TSQs	Copula, common and proper nouns, existential quantifiers, transitive verbs, noun and verb phrase coordination, relative pronouns, passives, query words

Table 1. Coverage of the FOEs and of TSQs.

– **Check:** does $\tau(\mathcal{S}) \cup \tau(\mathcal{F}) \models \tau(Q)\{x \mapsto c\}$?

where $\tau(Q)$ is a formula of (possibly) free variable x. By analogy to [15], we define the *data complexity* of KB-SAT and KB-QA as their computational complexity when $\mathcal{F}$ is the only input to the problem. The *size* $\#(\mathcal{F})$ of $\mathcal{F}$ is defined as the number of distinct proper names (or individual constants in $\tau(\mathcal{F})$) occurring in $\mathcal{F}$.

3 Data Complexity of the FOEs.

Resolution decision procedures. A *term* t is *(i)* a variable x or a constant c or *(ii)* an expression $f(t_1, \ldots, t_n)$ where f is a function symbol and $t_1, \ldots, t_n$ terms. In the latter case, we speak about *function terms*. A *litteral* L is a FO atom $P(t_1, \ldots, t_n)$. By a *clause* we understand a disjunction $L_1 \vee \cdots \vee L_n \vee \overline{N}_{n+1} \vee \cdots \vee \overline{N}_{n+m}$ of positive and negative litterals. The *empty* clause or *falsum* is denoted $\bot$. By $V(t)$, $V(L)$ and $V(C)$ we denote the sets of variables of, resp., term t, litteral L and clause C. A term, litteral, clause or set of clauses is said to be *ground* if it contains no free variables. A *substitution* σ is a function from variables to terms. It is called a *renaming* when it is a function from variables to variables. Substitutions can be extended to terms and litterals in the standard way. A *unifier* is a substitution σ s.t., given two terms t and t', $t\sigma = t'\sigma$. A *most general unifier* is a unifier σ s.t. for every other unifier σ' there exists a renaming σ'' with $\sigma' = \sigma\sigma''$.

The *depth* of a term is defined by *(i)* $d(x) := d(c) := 0$ and *(ii)* $d(f(t_1, \ldots, t_n)) := \max\{d(t_i) \mid i \in [1, n]\} + 1$. The *depth* $d(L)$ of a litteral L or $d(\Gamma)$ of set of clauses Γ is the maximal depth of their terms. The *relative depth* of a variable x in a term is defined by *(i)* $d(x, y) := d(x, c) := 0$ and *(ii)* $d(x, f(t_1, \ldots, t_n)) := \max\{d(x, t_i) \mid i \in [1, n]\} + 1$. The *relative depth* $d(x, L)$ of a variable x in a litteral L is its maximal relative depth among L's terms.

COP	$\varphi_l(x) \longrightarrow A(x)$ $\varphi_r(x) \longrightarrow \pm\varphi_l(x)$	$\forall x(\varphi_l(x) \Rightarrow \pm\varphi_r(x))$ $\exists x(\varphi_l(x) \wedge \varphi_r(x))$	No student failed. A student failed.
COP+TV	$\varphi_l(x) \longrightarrow A(x)$ $\varphi_r(x) \longrightarrow \pm\varphi_l(x) \mid \forall y(A(x) \Rightarrow \pm\psi(x,y))$ $\mid \exists y(A(x) \wedge \psi(x,y))$	$\forall x(\varphi_l(x) \Rightarrow \pm\varphi_r(x))$ $\exists x(\varphi_l(x) \wedge \varphi_r(x))$	No student failed. Some student follows every course.
COP+TV+DTV	$\varphi_l(x) \longrightarrow A(x)$ $\varphi_{tv}(x) \longrightarrow \pm\varphi_l(x) \mid \forall y(A(x) \Rightarrow \pm\psi(x,y))$ $\mid \exists y(A(x) \wedge \psi(x,y))$ $\varphi_{dtv}(x,y) \longrightarrow \forall z(A(x) \Rightarrow \pm\chi(x,y,z))$ $\mid \exists z(A(x) \wedge \chi(x,y,z))$ $\varphi_r(x) \longrightarrow \varphi_{tv}(x) \mid \forall y(A(x) \Rightarrow \pm\varphi_{dtv}(x,y))$ $\mid \exists y(A(x) \wedge \varphi_{dtv}(x,y))$	$\forall x(\varphi_l(x) \Rightarrow \pm\varphi_r(x))$ $\exists x(\varphi_l(x) \wedge \varphi_r(x))$	Every student gives no credit to some student. A student borrowed a book from some library.
COP+Rel	$\varphi_l(x) \longrightarrow A(x) \mid \pm\varphi_l(x) \wedge \pm\varphi_l(x)$ $\varphi_r(x) \longrightarrow \varphi_l(x)$	$\forall x(\pm\varphi_l(x) \Rightarrow \pm\varphi_r(x))$ $\exists x(\pm\varphi_l(x) \wedge \pm\varphi_r(x))$	Every student who is not dum is smart.
COP+TV+Rel	$\varphi_l(x) \longrightarrow A(x)$ $\varphi_r(x) \longrightarrow \pm\varphi_l(x) \mid \forall y(A(x) \Rightarrow \pm\psi(x,y))$ $\mid \exists y(A(x) \wedge \psi(x,y))$	$\forall x(\varphi_l(x) \Rightarrow \pm\varphi_r(x))$ $\exists x(\varphi_l(x) \wedge \varphi_r(x))$	No student failed. Some student studies every course.
COP+Rel+TV+DTV	$\varphi_l(x) \longrightarrow A(x) \mid \pm\varphi_r \wedge \pm\varphi_r$ $\varphi_{tv}(x) \longrightarrow \pm\varphi_l(x) \mid \forall y(A(x) \Rightarrow \pm\psi(x,y))$ $\mid \exists y(A(x) \wedge \psi(x,y))$ $\varphi_{dtv}(x,y) \longrightarrow \forall z(A(x) \Rightarrow \pm\chi(x,y,z))$ $\mid \exists z(A(x) \wedge \chi(x,y,z))$ $\varphi_r(x) \longrightarrow \varphi_{tv}(x) \mid \forall y(A(x) \Rightarrow \pm\varphi_{dtv}(x,y))$ $\mid \exists y(A(x) \wedge \varphi_{dtv}(x,y))$	$\forall x(\varphi_l(x) \Rightarrow \pm\varphi_r(x))$ $\exists x(\varphi_l(x) \wedge \varphi_r(x))$	Every helpful student gives some aid to some student. Some diligent student borrowed every book from every library.
TSQs	$\varphi(x) \longrightarrow A(x) \mid \exists y R(x,y) \mid \varphi_1(x) \wedge \varphi_2(x)$ $\mid \exists y(R(x,y) \wedge \varphi(y))$	$\varphi(x)$	Which student who attends some course is diligent?

Table 2. The MRs generated by the FOEs and TSQs. Note that $\psi(x,y)$ (resp. $\chi(x,y,z)$) stands for some binary (resp. ternary) atom, while $\pm$ means that a formula may or may not be negated. Complete FOE utterances comply with the pattern **Det N VP**, and complete (wh-)TSQs with the pattern **Intpro N VP** or $\mathbf{Intpro\ S}_g$, where $\mathbf{S}_g$ denotes a (subordinate) clause.

We consider the so-called *saturation-based* version (or format) of the resolution calculus in which we iteratively (monotonically w.r.t. $\subseteq$) generate the set of all possible clauses derived from Γ using the rules

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

where σ is a most general unifier (of L and L' in this case), until either *(i)* $\bot$ is derived or *(ii)* all possible clauses are generated (fixpoint computation). Formally, consider a function $\rho(\cdot)$ over sets of clauses, defined in terms of *res* amd *fact*. A *resolution calculus* is a function $\mathcal{R}(\cdot)$ s.t. $\mathcal{R}(\Gamma) := \Gamma \cup \rho(\Gamma)$. A *derivation* δ from Γ is defined by putting *(i)* $\mathcal{R}^0(\Gamma) := \Gamma$ and $\mathcal{R}^{i+1}(\Gamma) := \mathcal{R}(\mathcal{R}^i(\Gamma))$, for $i > 0$. Thereafter the *saturation* of Γ is defined as $\Gamma^\infty := \bigcup\{\mathcal{R}^i(\Gamma) \mid i \geq 0\}$. The positive integer i is called the *depth* or *rank* of δ. The set(s) of clauses derived at each rank $i \geq 0$ of δ is (are) called the *state(s)* of δ. The *size* of δ is defined as its total number of states. Resolution is sound and complete w.r.t. (un)satisfiability: Γ is unsatisfiable iff $\bot \in \Gamma^\infty$. Moreover, if Γ is satisfiable, we can build out of Γ^∞ a Herbrand model of Γ [5].

Resolution saturations are not in general computable (they may not converge finitely). However, Joyner in [7] showed that finite convergence can be achieved provided that two conditions are met: *(i)* that the depth of litterals does not grow beyong a certain bound $d \geq 0$ and *(ii)* that the length of clauses (the number of disjunctions) does not grow beyond a bound $l \geq 0$. Several *refinements* can be used to ensure the existence of such bounds and a fortiori finite convergence for several fragments of FO.

To control depth, *acceptable orderings* (A-orderings), that is, well-founded and substitution-invariant partial orders on clause litterals and sets thereof, can be used (which force resolution on litterals that are maximal w.r.t. the ordering). The best known is the $\prec_d$ ordering defined by

$$L \prec_d L' \text{ iff } d(L) < d(L'), V(L) \subseteq V(L') \text{ and, for all } x \in V(L), d(x, L) < d(x, L),$$

a refinement sound and complete w.r.t. satisfiability. To control length the splitting rule

$$\textit{split}\ \frac{\begin{array}{ccc} \Gamma, C \vee L & & \Gamma, C \vee L' \\ \vdots & & \vdots \\ \Gamma, C \vee L \vee L' & C'\sigma & C'\sigma \end{array}}{C'\sigma}\ (V(L) \cap V(L') = \emptyset)$$

can be used (it is sound and complete w.r.t. satisfiability). These refinements are guaranteed to work the way we want them to in case they are applied to *covering* clauses. A litteral L is said to be covering whenever *(i)* $d(L) = 0$ or *(ii)* for every functional term t in L, $V(t) = V(L)$. If all the litterals of a clause C are covering, so is C. This property is not, however, closed under resolution or its refinements: applying them to covering clauses may result in non-covering clauses. To prevent this from happening, a further refinement is required: *monadization* [7]. Intuitively, what this does is to reduce the (un)satisfiability of non-covering clauses, satisfying some structural properties, into that of a set of covering clauses. The applicability of the refinements thus depends on

the FO fragments such clauses are drawn from, but, whenever *all* are applicable, saturations finitely converge [5].

The different systems arising from the different combinations of rules, orderings and refinements are summarized by Table 3. Note that saturations exhibit the shape of a tree (of branching factor 2) or of a sequence, depending on whether the calculi make use or not of the splitting rule.

In particular, the $\mathcal{R}_{2,5}$ calculus of Table 3 decides the $\mathcal{S}^+$ class of clauses [5]. The class $\mathcal{S}^+$ is the class where every clause C satisfies: *(i)* $V(C) = V(t)$, for every functional term t in C, and *(ii)* either L has at most one variable or $V(L) = V(C)$, for every litteral L in C.

Data Complexity of KB-QA and KB-SAT. In this section we study the data complexity of KB-SAT and KB-QA by applying resolution decision procedures to the syllogistic FOEs. We apply data complexity arguments to sets $\Sigma \cup \Delta$ of non-ground and ground clauses. This makes sense, because, modulo $\tau(\cdot)$ and clausification, FOE constraints $\mathcal{S}$ map to sets Σ of non-ground clauses, FOE facts $\mathcal{F}$ map to sets Δ of ground clauses, and, in general, KBs $(\mathcal{S}, \mathcal{F})$ to sets $\Sigma \cup \Delta$ of clauses.

We do as follows. For the tractable FOEs we rely on the "separation" property of resolution saturations [5] (resolution of ground clauses can be delayed to the end). For the intractable, on the "monadic reducibility" property shown by Pratt and Third in [11] that enforces a reduction to $\mathcal{S}^+$ clauses for the fragments involved; this we combine with a data complexity of the $\mathcal{S}^+$ class (and saturations).

- **Separation:** $\bot \in (\Sigma \cup \Delta)^\infty$ iff there exists a set $\Sigma' \subseteq \Sigma^\infty$ s.t. *(i)* $d(\Sigma') \leq d(\Delta)$, *(ii)* $\bot \in (\Sigma' \cup \Delta)^\infty$ and *(iii)* Σ' is finite.
- **Monadic reducibility:** every set Γ of COP+TV+DTV+Rel clausified MRs (or any fragment thereof). can be polynomially (in the size of Γ) transformed into a set Γ_u of unary clauses s.t. Γ is satisfiable iff Γ_u is satisfiable.

Lemma 1. *Let* $(\mathbf{C}, \mathbf{F}, \mathbf{R})$ *be a finite FO signature, where* **C** *is a (finite) set of constants,* **F** *a (finite) set of function symbols and* **R** *a (finite) set of predicate symbols. Consider a clause set* Γ *over such signature and suppose that there exist both a term depth bound* $d \geq 0$ *and a clause length bound* $k \geq 0$*. Then*

1. *the number of clauses derivable by the saturation is (worst-case)*
   (a) exponential in the number of constants in **C** *if we use the splitting rule or*
   (b) polynomial in the number of constants in **C** *otherwise, and*
2. *the depth of the saturation is (worst-case) polynomial in in the number of constants in* **C**.

Proof. Assume that a depth bound d and a length bound l exist. Let c be the number of constant symbols in $\mathbf{C}$, v the number of variables in $\mathbf{V}$, f the number of function symbols in $\mathbf{F}$, p the number of predicate symbols in $\mathbf{R}$, ar_f the maximum arity of the function symbols, and ar_p the maximum arity of the predicate symbols. We can define

		split	*mon*	*split* *mon*
	$\mathcal{R}_{1,1}$	$\mathcal{R}_{1,2}$	$\mathcal{R}_{1,4}$	$\mathcal{R}_{1,5}$
$\prec_d$	$\mathcal{R}_{2,1}$	$\mathcal{R}_{2,2}$	$\mathcal{R}_{2,4}$	$\mathcal{R}_{2,5}$

Table 3. Resolution calculi.

the number te_i of terms of depth $i \geq 0$ inductively by setting *(i)* $te_0 := v + c$, *(ii)* $te_{i+1} := f \cdot te_n^{ar_f}$. Thus, the number *te* of terms of depth $\leq d$ is

$$te \leq \sum_{i=0}^{d} te_i = f^0 \cdot (v+c)^{ar_f^0} + ... + f^d \cdot (v+c)^{ar_f^d} := \mathtt{p_{te}}(c) \tag{1}$$

which defines a polynomial $\mathtt{p_{te}}(c)$. This in its turn yields as upper bound to the number *li* of positive and negative literals

$$li \leq 2 \cdot p \cdot te^{ar_p} = 2 \cdot p \cdot \mathtt{p_{te}}(c)^{ar_p} := \mathtt{p_{li}}(c) \tag{2}$$

thus defining a polynomial $\mathtt{p_{li}}(c)$. Finally, from *li* we derive an upper bound to the number *cl* of clauses of length $\leq l$

$$cl \leq li^l = \mathtt{p_{li}}(c)^l := \mathtt{p_{cl}}(c) \tag{3}$$

which again defines a polynomial $\mathtt{p_{cl}}(c)$. The splitting rule splits saturations into two, yielding a (saturation) tree of worst-case size $\leq 2^{\mathtt{p_{cl}}(c)}$, largest (derived) state of size $\leq \mathtt{p_{cl}}(c)$ and that will converge after $\leq \mathtt{p_{cl}}(c)$ iterations. □

Theorem 1. KB-SAT *is in* **NP** *in data complexity for* $\mathcal{S}^+$.

Proof. Let $\Sigma \cup \Delta$ be a set of $\mathcal{S}^+$ clauses. Consider now a $\mathcal{R}_{2,5}$-saturation. Calculus $\mathcal{R}_{2,5}$ decides $\mathcal{S}^+$ and saturations finitely converge. Assume w.l.o.g. that Σ contains no constants and that Δ is of depth $d(\Delta) = 0$ and has c distinct constants (where $c \geq 0$). By Lemma 1 we know that the saturation will be tree-shaped, of rank $\leq \mathtt{p}(c)$, of size $\leq 2^{\mathtt{p}(c)}$ and of maximal state of size $\leq \mathtt{p}(c)$.

Outline a non-deterministic algorithm for KB-SAT as follows. Start with $\Sigma \cup \Delta$. For each rank $i \in [0, \mathtt{p}(c)]$ of the saturation, guess/choose a state $j \in [0, 2^i]$. Notice that the algorithm will make polynomially many choices on c. Finally, check, in time polynomial in c whether $\bot$ is in the resulting state, and, if no, compute, in time polynomial in c, a Herbrand model of $\Sigma \cup \Delta$. □

Theorem 2 (KB-SAT). *The data complexity for* KB-SAT *is*

1. *in* **LSpace** *for COP, COP+TV and COP+TV+DTV,*
2. *in* **NP** *for COP+Rel, and*
3. **NP***-complete for COP+Rel+TV, COP+Rel+TV and COP+Rel+TV+DTV.*

	TSQs	**Fragment**
COP	in **LSpace** [Th 3]	in **LSpace** [Th 2]
COP+TV	in **PTime** [Th 3]	in **LSpace** [Th 2]
COP+TV+DTV	in **coNP**	in **LSpace** [Th 3]
COP+Rel	**coNP**-complete [10]	in **NP** [Th 2]
COP+Rel+TV	**coNP**-complete [10]	**NP**-complete [10]
COP+Rel+DTV	**coNP**-complete [Th 3]	**NP**-complete [Th 3]
COP+Rel+DTV+TV	**coNP**-complete [Th 3]	**NP**-complete [Th 3]

	Atomic question	**Fragment**
COP+Rel+TV+GA	undecidable [Th 4]	undecidable [11]
COP+Rel+DTV+TV+RA	undecidable [Th 4]	undecidable [11]
COP+Rel+DTV+TV+GA	undecidable [Th 4]	undecidable [11]

	TSQs+RA	**Fragment**
COP+Rel+TV+RA	undecidable [Th 4]	**NP**-complete [Th 3]

Table 4. Data complexity of KB-QA and KB-SAT (a.k.a. fragment complexity) for the syllogistic FOEs and TSQs.

Proof. (Sketch.) For the fragments COP, COP+TV and COP+TV+DTV we reason as follows. Let $(\mathcal{S}, \mathcal{F})$ be a KB and consider its MRs $\tau(\mathcal{S})$ and $\tau(\mathcal{F})$ (which can be computed in space logarithmic in $\#(\mathcal{F})$). Computing their skolemization and clausification does not affect data complexity, since it is the identity for $\tau(\mathcal{F})$. By inspecting the resulting clauses we can observe that they are covering: using A-ordered resolution prevents clauses from growing beyond a certain depth bound d. Furthermore, it can be proven that applying *res* and *fact*, does not increase clause length beyond a certain bound l, nor does it result in non-covering clauses. Therefore, the A-ordered resolution calculi without splitting from Table 3 decide the satisfiability of $\tau(\mathcal{S}) \cup \tau(\mathcal{F})$. In addition, we know by the "separation" property that we can "separate" data from facts provided $\tau(\mathcal{S})$ is satisfiable.

Sketch a decision algorithm for KB-SAT as follows. Check whether $\tau(\mathcal{S})$ is satisfiable, i.e., whether $\bot \in \tau(\mathcal{S})^\infty$, computation that does not depend on $\#(\mathcal{F})$ (or $\#(\tau(\mathcal{F}))$). If the answer is negative, return "no". If the answer is positive: *(i)* Compute the finite model $\mathcal{D}$ of $\tau(\mathcal{F})$ (i.e., the Herbrand model defined from $\tau(\mathcal{F})$). *(ii)* Compute the FO formula $\varphi_{\mathcal{S}} := \bigwedge\{C \mid C \text{ clause of } \tau(\mathcal{S})^\infty\}$. Then,

$$\tau(\mathcal{S}) \cup \tau(\mathcal{F}) \text{ is satisfiable iff } \mathcal{D} \models \varphi_{\mathcal{S}},$$

which outlines a reduction to relational database query answering, known to be in **LSpace** (actually, in AC^0) [1]. Membership in **LSpace** follows.

Membership in **NP** for COP+Rel, COP+Rel+TV and COP+Rel+TV+DTV is derived as follows. Consider a KB $(\mathcal{S}, \mathcal{F})$. Consider now the resulting MRs, $\tau(\mathcal{S})$ and $\tau(\mathcal{F})$. Clausifying such MRs can be done in time constant in $\#(\tau(\mathcal{F}))$. By Pratt and Third's "monadic reducibility" property, we know that we can reduce, in time poly-

nomial in $\#(\tau(\mathcal{F}))$ their satisfiability to that of a set $\tau(\mathcal{S})_u \cup \tau(\mathcal{F})_u$ of monadic clauses. By inspection, we can, moreover, observe that such classes belong to the $\mathcal{S}^+$ class. We can now apply Lemma 1, whence it follows that KB-SAT is in **NP**. For COP+Rel+TV+RA we observe that the "monadic reducibility" property still holds for *restricted anaphoric pronouns* [11], wherein we impose pronouns like "him" to co-refer with their closest antecedent noun phrase within, moreover, a single utterance (and not beyond).

Finally, **NP**-hardness for COP+Rel+TV and COP+Rel+TV+DTV can be inferred by a reduction from the **NP**-complete satisfiability problem for 2+2 clauses [12]. A 2+2 clause is a clause $L_1 \vee L_2 \vee \overline{L}_3 \vee \overline{L}_4$ containing two positive litterals and two negative litterals. □

Theorem 3 (KB-QA). *If we consider TSQs, then the data complexity of* KB-QA *is*

1. *in* **LSpace** *for COP,*
2. *in* **PTime** *for COP+TV,*
3. *in* **coNP** *for COP+TV+DTV, and*
4. **coNP***-complete for COP+Rel, COP+Rel+TV and COP+Rel+TV+DTV.*

Proof. (Sketch.) KB-QA for COP is in **LSpace** in data complexity, because it can be shown that its MRs are contained by the description logic *DL-Lite*, for which such result holds [2]. Similarly, it can be shown that COP+TV KB-QA reduces to Datalog KB-QA. Furthemore, given a COP+TV KB $(\mathcal{S}, \mathcal{F})$ and a TSQ Q, such reduction proceeds in space logarithmic in $\#(\mathcal{F})$. It thus preserves data complexity. Since Datalog KB-QA is in **PTime**, the result follows.

The **coNP** upper bound for COP+Rel and COP+Rel+TV follows from the **coNP**-completeness for data complexity of KB-QA for the two-variable fragment of FO [10]. Regarding COP+TV+DTV and COP+Rel+TV+DTV, we observe that: *(i)* TSQs can be expressed quite easily by COP+Rel+TV+DTV, by extending this FOE with grammar rules accounting for wh- and y/n-questions. *(ii)* COP+Rel+TV+DTV is closed under negation. We can thus reduce KB-QA (again, by a reduction space logarithmic in the size of the data) to COKB-QA (i.e., the complement of KB-SAT) and apply Theorem 2.

Finally, **coNP**-hardness derives from the fact that we can again reduce the satisfiability of 2+2 clauses to COP+Rel COKB-QA (i.e., the complement of KB-QA). This lower bound then propagates to COP+Rel+TV and COP+Rel+TV. □

Theorem 4. KB-QA *is undecidable*

1. *for COP+Rel+TV+RA with TSQs+RA, and*
2. *for COP+Rel+TV+GA and COP+Rel+TV+DTV+RA with atomic questions.*

Proof. (Sketch.) We can define a reduction from the unbounded tiling problem, known to be undecidable, to KB-QA for COP+Rel+TV+RA with *indeterminate pronouns* (e.g., "Anybody who does not love somebody, hates him.") and TSQs+RA, i.e., TSQs where anaphoric pronouns have been added to the fragment (e.g., "Does some man like somebody who hates him?").

For COP+Rel+TV+GA and COP+Rel+TV+DTV+RA the result follows by reduction from unsatisfiability and by the fact that, as it was shown in [11], SAT is undecidable for these fragments. The reduction requires atomic y/n-questions (e.g. "Is Socrates a philosopher?"). □

4 Conclusions

We have studied the data complexity of Pratt's syllogistic FOEs w.r.t. KB-SAT (viz., KB satisfiability) and KB-QA (viz., answering TSQs over KBs). In so doing, we have assessed their scalability as front-end languages for OBDASs, in particular w.r.t. data and constraint declaration and querying, which the aforementioned decision problems formalize. Our results show that the data complexity of the non-recursive fragments, COP, COP+TV and COP+TV+DTV, are grosso modo, tractable (the upper bound for KB-QA for COP+TV+DTV is not tight and could be improved), and that data complexity is grosso modo, intractable, when relatives are added (the upper bound for KB-SAT for COP+Rel is not tight either). Adding anaphoric pronouns either to the syllogistic FOEs alone or in combination with TSQs results, in general, in undecidability.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Welsey, 1995.
2. R. Bernardi, D. Calvanese, and C. Thorne. Expressing *DL-Lite* ontologies with controlled English. In *Proceedings of the 20th International Workshop on Description Logics (DL 2007)*, 2007.
3. D. Braines, J. Bao, P. R. Smart, and N. R. Shadbolt. A controlled natural language interface for semantic media wiki using the Rabbit language. In *Proceedings of the 2009 Controlled Natural Language Workshop (CNL 2009)*, 2009.
4. D. Calvanese, G. de Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Efficiently managing data intensive ontologies. In *Proceedings of the 2nd Italian Semantic Web Workshop: Semantic Web Applications and Perspectives (SWAP 2005)*, 2005.
5. C. G. Fermüller, A. Leitsch, U. Hustadt, and T. Tammet. *Resolution Decision Procedures*, volume 2 of *Handbook of Automated Reasoning*, chapter 2, pages 1791–1849. Elsevier - The MIT Press, 2001.
6. N. E. Fuchs and K. Kaljurand. Mapping Attempto Controlled English to OWL-DL. In *Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006)*, 2006.
7. W. H. J. Jr. Resolution strategies as decision procedures. *Journal of the ACM*, 23(3):398–417, 1976.
8. E. Kaufmann and A. Bernstein. How useful are natural language interfaces to the semantic web for casual end-users? In *Proceedings of the 6th International Web Conference and the 2nd Asian Web Conference (ISWC/ASWC 2007)*, pages 281–294, 2007.
9. R. Montague. Universal grammar. *Theoria*, 36(3):373–398, 1970.
10. I. Pratt. Data complexity of the two-variable fragment with counting quantifiers. *Information and Computation*, 207(8):867–888, 2008.
11. I. Pratt and A. Third. More fragments of language. *Notre Dame Journal of Formal Logic*, 47(2):151–177, 2006.
12. A. Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *Journal of Intelligent Information Systems*, 2(3):265–278, 1993.
13. S. Staab and R. Studer, editors. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, 2004.
14. C. Thorne and D. Calvanese. Tree shaped aggregate queries over ontologies. In *Proceedings of the International Conference on Flexible Query Answering Systems (FQAS 2009)*, 2009.
15. M. Vardi. The complexity of relational query languages. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, 1982.