Managing Structured Data with Controlled English and Description Logics

CAMILO THORNE
KRDB Research Centre
Free University of Bozen-Bolzano
3, Piazza Domenicani, 39100 (Italy)
cthorne@inf.unibz.it

ABSTRACT. Retrieving data from a database is a well-defined and unambigous task, namely that of evaluating a formal query of a limited expressiveness over an instance of the database schema. This query characterizes exactly and unequivocally the data to be retrieved. The same holds for specifying and storing relational data. But database interfaces are often obscure for non-experts and even more so interfaces supporting ontology reasoning services. We believe that this problem can be solved by advocating the use of controlled languages. That is, by defining a fragment of English, Lite English, that compositionally translates into a description logic, $\mathsf{DL\text{-}Lite}_R$, well-suited for data-management tasks, with very tight expressivity bounds and for which efficent (**LOGSPACE**) query answering (QA) algorithms exist.

1 Introduction

The tasks of structuring, modelling, declaring, updating and querying data are all but trivial, let alone intuitively appealing to a casual end-user. The interfaces of relational database management systems (RDBMSs) like dBASE or Oracle are based on query languages such as SQL or Datalog which require skills way beyond those of non-experts. This task becomes even harder when we consider adding a reasoning layer over RDBMSs, by using ontologies and advocating an *ontology-driven data access* strategy (cf. [4]).

This may be avoided by shifting to natural language interfaces as propounded by, e.g., Androustopoulos in [1]. But query and ontology languages are formal languages (combining declarative and imperative features) that allow no place for ambiguity (cf. [12]), whereas natural language is full of ambiguities. In particular, retrieving data of a relational datasource w.r.t. ontology involves satisfying some crucial requirements or constraints, namely:

(i) A formal query must characterize exactly the data to be retrieved from a database – the set of tuples (or records) that satisfy it (cf. [12]).

Proceedings of the Twelfth ESSLLI Student Session
Ville Nurmi and Dmitry Sustretov (editors)

(ii) The expressive power of query languages must be well-known. The problem of query answering (QA) for relational databases (DBs) falls under **LOGSPACE** w.r.t. *data complexity*, i.e., on the number of records of the database (cf. [12]).

(iii) When we take into account ontologies, QA becomes the *entailment* $\mathcal{O}, \mathcal{D} \models \psi$, where $\mathcal{O}$ is an ontology, a logical theory about the domain, $\mathcal{D}$ a DB and ψ a formal query. Hence, the expressive power of the ontology language should also be subject to strict expressivity bounds: QA should stay in **LOGSPACE**.

To address the problem of managing structured data w.r.t. an ontology with NL a compromise between the expressive power and features of query and ontology languages and the intuitive appeal of NL has to be reached. We believe that this compromise involves the use of *controlled languages*, partly following the ideas suggested by Sowa in [14]. Controlled languages are fragments of NL tailored to deal with data management tasks, which means that they have been stripped of every ambiguity. Their utterances translate into some logical expression that encodes semantics at the sentence level (cf. [14]) and which can then be postprocessed into, say, SQL.

The remainder of this paper is structured as follows. In section 2 we will argue that controlled languages should be combined with the so-called *fragments of language* approach of I. Pratt and A. Third, which deal explicitly with the issue of expressive power. Section 3 will be devoted to DL-Lite_R, the logic we consider the best suited to carry on with data management tasks and that we have taken as the basis of our controlled language. Section 4 will then introduce the controlled language, Lite English, we are working in, together with some expressive power results. Last, but not least, Section 5 will outline our conclusions so far and the ongoing work.

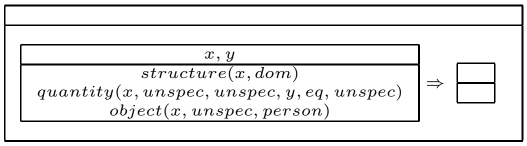
2 Fragments of English and Controlled Languages

Controlled languages (CLs) are fragments of natural language, say, of English, with a limited lexicon and set of grammar rules. Their main aim is to perform data management tasks in NL: specifying, declaring and querying data stuctured and stored in knowledge bases or databases, like, e.g., Attempto Controlled English (ACE) in [11]. They must be able to express three things: (a) the specification or conceptualization (a.k.a. *ontology*) of the domain, (b) the data and (c) the user queries (cf. [14]). Constraints on components are used to force utterances to have a unique parse tree and to translate into a unique semantic representation. This has the effect of eliminating ambiguity.

Example 2.1. Let us have a look at how ACE works. In ACE gammar rules have the form:

```
WhQuestion --> NP[+Q,+NOM] VP/-
```

The features are to be read as follows: a Wh-question must be rewritten into a focus (+NOM) NP containing a wh-word (+Q) and a VP with no gaps (/-). As semantic representation, ACE uses discourse representation structures (DRSs). For example, this would be the DRS associated to an indefinite NP (cf. [11])



But then, the issue of the expressive power of these semantic representations and how they are computed becomes critical, if we are to satisfy the requirements **(i)** – **(iii)** on data management and access we underlined in the introduction. Why? Because these semantic representations may belong to *any* logic, some of which may lie beyond the reach of any formal query or ontology language for which QA w.r.t. an ontology is tractable. For instance, DRSs are as expressive as FOL, which would imply QE to be undecidable.

CLs have, therefore, to be complemented with *fragments of natural language* (FLs), an approach focused in measuring the contribution of each syntactic construct of NL to expressive power and computational complexity. This is possible because following formal semantics, their utterances compositionally translate *modulo* a compositional translation ϕ into a semantic representation called *meaning representation* (MR). This MR is, typically, a FOL formula. This approach stems from Ian Pratt and Alan Third's work on English FLs (cf. [13]). Pratt and Third build incrementally a family of English fragments by starting from a fragment, COP, that covers very basic constructs, like copula, nouns, negation and quantification. Its coverage is then extended to other NL constructs:

Fragment	**Coverage**	**Decision class for SAT**
COP	Copula, common and proper nouns, negation, universal and existential quantifiers.	**P**
COP+TV+DTV	COP + transitive verbs, distransitive verbs.	**P**
COP+REL	COP + relative pronouns.	**NP-Complete**
COP+REL+TV	COP + transitive verbs, relative pronouns.	**EXPTIME-Complete**
COP+REL+TV+DTV	COP+TV+DTV + relative pronouns.	**EXPTIME-Complete**
COP+REL+TV+RA	COP+REL+TV + restricted (intrasentential) anaphora.	**NEXPTIME-Complete**
COP+REL+TV+GA	COP+REL+TV + generalized anaphora.	undecidable

The *expressive power* of a FL is then defined as that of the fragment of FOL into which it compositionally translates. This is because the set of MRs of each fragment constitutes a fragment of FOL. The key idea is that each NL construct adds a new logic construct to the underlying FOL fragment, modifying both its expressive power and its computational properties. In particular, each addition affects the complexity of the satisfiability (SAT) and entailment problems of the logic fragments, until they become undecidable. Note further that as they are closed under boolean negation, entailment reduces to SAT.

Example 2.2. Some examples may help at this point. As the reader can see, each utterance gives way to a FOL meaning representation exhibiting different logical constructs, following the semantics of function words and content words in each fragment (cf. [13]):

Sentence	MR (FOL)	Fragment
No man is a woman	$\forall x(Man(x) \rightarrow \neg Woman(x))$	COP
Every man *who* is not dead is alive	$\forall x(Man(x) \wedge \neg Dead(x) \rightarrow Alive(x))$	COP+ REL
Every scholar *reads* some book	$\forall x((Scholar(x) \rightarrow \exists y(Book(y) \wedge Reads(x,y)))$.	COP+ TV
Every salesman *sells* some merchandise to some customer	$\forall x(Salesman(x) \rightarrow \exists y(Customer(y) \wedge \exists z(Merchandise(z) \wedge Sells(x,y,z)))$.	COP+ TV+DTV

The complexity of analysis I. Pratt and A. Third above show that only the first two fragments, COP and COP+TV+DTV are tractable. But, why is this the case? Because of the logic constructs these NL constructs can express or capture. A careful glance at the fragments tells us the following:

- Quantifiers may occur in any order.
- Negation expresses boolean negation.
- Relative pronouns express boolean conjunction.
- Transitive verbs express binary relations. Ditransitive verbs, ternary relations. Nouns, unary relations (i.e., sets).

The introduction of *relative clauses* produces an exponential blowup, and therefore intractability. Why? Because COP+REL can express boolean conjunction and negation. That is, a complete set of boolean operators, whence MRs become as expressive as the propositional calculus which is **NP-Complete**. On the other hand, introducing binary and ternary relations does not affect computational complexity.

Recall that it is essential (requirement **(iii)**) for entailment to be tractable. This means that, if we want to retain this property, we have to stick, as far as possible, to the English constructs that COP and COP+TV+DTV cover. As we will see, this is actually the case when we define a FL or a CL that compositionally translates into DL-Lite$_R$.

3 DL-Lite and QA

$\mathsf{DL\text{-}Lite}_R$ (cf. [7]) is a description logic specifically tailored to meet data management requirements and tasks and in particular, to query relational data sources w.r.t. an ontology (cf. [5]). $\mathsf{DL\text{-}Lite}_R$ allows us to encode ontologies and databases as logical theories called *knowledge bases* which can then be queried with simple formal queries called *conjunctive queries*. Querying is defined in terms of logical entailment. Given that both conjunctive queries and $\mathsf{DL\text{-}Lite}_R$ are decidable fragments of FOL, this definition makes sense. Moreover, QA can be decided efficiently. In this way, it satisfies the requirements **(i)** – **(iii)**. It is, as a matter of fact, a maximal description logic for which this property holds.

Definition 3.1. (Concepts) Let $\mathcal{P} = \{P_i | i \in \mathbb{N}\}$ and $\mathcal{R} = \{R_i | i \in \mathbb{N}\}$ be two countable sets of primitive concept and role symbols. DL-Lite *left hand side concepts* $\mathcal{C}_l$ and *right hand side concepts* $\mathcal{C}_r$ are defined as follows:

$$\begin{aligned}
\mathcal{C}_l &::= \mathcal{P} \mid \exists\mathcal{R} \mid \exists\mathcal{R}^- \mid \mathcal{C}_l \sqcap \mathcal{C}_l.\\
\mathcal{C}_r &::= \neg\mathcal{P} \mid \neg\exists\mathcal{R} \mid \neg\exists\mathcal{R}^- \mid \mathcal{C}_l \mid \mathcal{C}_r \sqcap \mathcal{C}_r \mid \exists\mathcal{R}\colon \mathcal{C}_r \mid \exists\mathcal{R}^-\colon \mathcal{C}_r.
\end{aligned}$$

Concepts stand for sets. Role symbols for binary relations. R^- stands for the inverse of R. Concepts of the form $\exists R$ are known as *unqualified existential roles*, and are built by existentially quantifying the second argument of the relation. Concepts of the form $\exists R\colon C$ are called *qualified existential roles* and are similar to unqualified roles, only that this time we assert as well that the quantified argument falls under concept C.

Definition 3.2. (Assertions) Let $\mathcal{C}_o = \{c_i | i \in \mathbb{N}\}$ be a set of constants. DL-Lite *facts* $\mathcal{A}$ and *terminological assertions* $\mathcal{T}$ are defined as follows:

$$\begin{aligned}
\mathcal{A} &::= \mathcal{P}(\mathcal{C}_o) \mid \mathcal{R}(\mathcal{C}_o, \mathcal{C}_o) && \text{(facts)}\\
\mathcal{T} &::= \mathcal{C}_l \sqsubseteq \mathcal{C}_r \mid \mathcal{R} \sqsubseteq \mathcal{R} && \text{(terminological assertions)}
\end{aligned}$$

Concept *subsumption* ($\sqsubseteq$) stands for set inclusion. Assertions are bundled into *knowledge bases* (KBs): tuples of the form $\langle \mathsf{ABox}, \mathsf{TBox} \rangle$, where the ABox is a set of facts and the TBox a set of terminological assertions. The TBox encodes, intuitively, the ontologies or conceptual models of the data (the data constraints) and the ABox the actual data to be declared or stored. The *size* of an ABox is given by the number of pairwise distinct constants of its ABox, denoted $\#(\mathsf{ABox})$. This notion is a.k.a. the *data complexity* of a KB.

Example 3.1. The following is an example of a $\mathsf{DL\text{-}Lite}_R$ KB. We want to specify in a KB $\mathcal{K}_0$ some properties of the domain of men, limited to a single individual, James. These properties are their being mortal and their owning a car:

TBox	**ABox**
$Man \sqsubseteq Mortal$	$Man(James)$
$Man \sqsubseteq \exists Owns\colon Car$	

We can extract data from KBs by using formal queries. They are queried with conjunctive queries:

Definition 3.3. (Conjunctive Queries) A *conjunctive query* (CQ) is an expression of the form $q(\vec{x}) \leftarrow \exists\vec{y}\beta(\vec{x},\vec{y})$ where $\vec{x}$ is a possibly empty finite sequence of *distinguished variables* and $\beta(\vec{x},\vec{y})$ (the *body*) a conjunction of atoms over $\vec{y}$, $\vec{x}$ using DL-Lite$_R$ basic roles and concepts, as well as individual constants. Relation q is its *head.* If $\vec{x}$ is empty, the query is said to be *boolean.*

As a matter of fact, a CQ is basically a FOL open formula with k free variables, where k is the arity of its head. They are also equivalent to the SELECT-PROJECT-JOIN fragment of SQL. We can now formally define QA for DL-Lite$_R$: it consists in computing the answer set of a CQ q over an ABox w.r.t. a TBox and, ultimately, in deciding the *entailment* stated in **(iii)**:

Definition 3.4. (Query Answering) The *semantics* of a CQ q is the set of constant sequences $\vec{c}$ such that the logical entailment $\langle \mathsf{TBox}, \mathsf{ABox}\rangle \models q(\vec{c})$ holds – where $q(\vec{c})$ denotes the grounding of the CQ q w.r.t. $\vec{c}$.

Example 3.2. Consider the following question: "Who owns a car?". The corresponding CQ is: $q := q(x) \leftarrow \exists y Owns(x,y)$. If we evaluate this query over $\mathcal{K}_0$ we will get as answer set $\{James\}$, since $\mathcal{K}_0 \models q() \leftarrow \exists y Owns(James,y)$. Note that the grounding of q is the boolean query obtained from q by erasing its distinguished variable and applying the closed substitution $[John/x]$ on its body.

But this entailment problem would not be interesting for our purposes if it had not the right properties. The fundamental result is that it does. It can be decided quite efficiently (w.r.t. data complexity):

Theorem 3.1. (Calvanese *et. al.* [5]) *Deciding whether* $\langle \mathsf{TBox}, \mathsf{ABox}\rangle \models q(\vec{c})$ *holds is* **LOGSPACE** *on data complexity that is, on* $\#(\mathsf{ABox})$.

Moreover, DL-Lite$_R$ is a *tractable* logic: TBox satisfiability is in **P**, which means that it is similar in expressivity to the tractable fragments of English. It is actually a fragment of FOL (cf. [2]) with some syntactic restrictions. Restrictions that explain why QA and SAT can be decided so efficiently, as opposed to other, more expressive, description logics (cf. [2, 10, 5]): terminological assertions are assumed to be (implicitly) universally quantified. There are no variables. But, most importantly: *negation* occurs only to the *right* of $\sqsubseteq$. Futhermore, facts contain *no* negation. Last, but not least, DL-Lite$_R$ assertions belong to the $\forall\exists^*$ FOL prefix class: quantifier prefixes occur in a *fixed* order.

4 Lite English

The second step to tackle the problem of managing data with NL involves defining a controlled language, that we have called Lite English (cf. [3]), and studying its expressivity. Why? Because now that we have chosen a suitable logic, we need to see which NL constructs can be safely translated into DL-Lite$_R$ concepts and assertions. Moreover, since SAT for any DL-Lite$_R$ TBox is in **P**, we must make clear to what extent Lite English differs from COP and COP+TV+DTV – the tractable FLs. This latter issue is particularly crucial, since it helps us in identifying the NL constructs that play a distinctive role in QA. Lite English has been "reverse-engineered" from DL-Lite and consists in two fragments:

- The *declarative fragment* of declarative sentences, based on DL-Lite$_R$. It translates into TBox and ABox assertions.
- The *interrogative fragment* of Wh and Y/N-questions based on the CQs Wh-questions are translated into CQs and Y/N-questions into boolean CQs with no free variables.

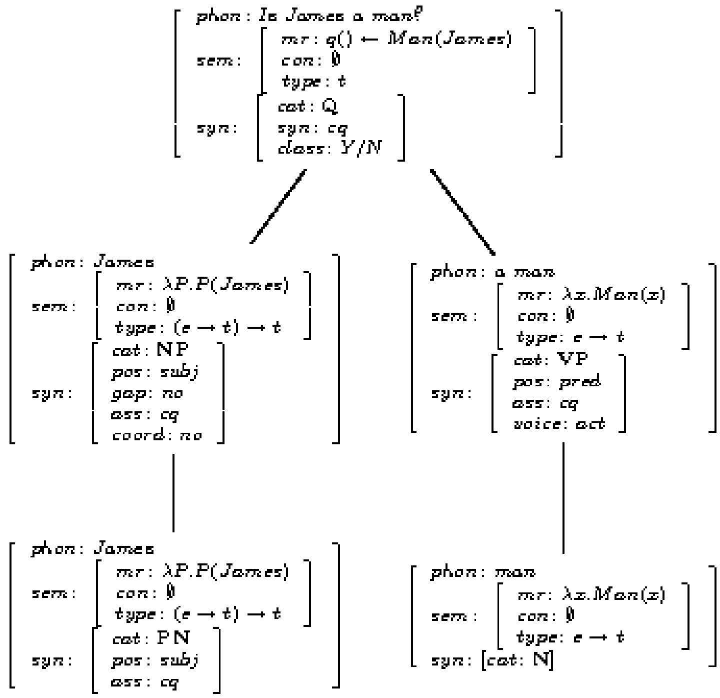
Clearly, as we use a compositional translation in the spirit of Pratt and Third, this implies that the *expressive power* of these fragments (and of Lite English itself) is simply that of CQs and DL-Lite. Clearly, Lite English satisfies data management constraints **(i)** – **(iii)** and captures QA in a fragment of English.

Example 4.1. The following table gives an idea of what we mean by the utterances Lite English can capture:

Lite English	Fragment	MR (DL-Lite+CQs)
Every salesman sells some merchandise *that* is expensive	declarative	$Salesman \sqsubseteq \exists Sells\colon Merchandise \sqcap Expensive$
John is uninteresting	declarative	$Uninteresting(John)$
No man is a woman	declarative	$Man \sqsubseteq \neg Woman$
Everybody *who* likes something succeds	declarative	$\exists Likes \sqsubseteq Succed$
Who knows Roger?	interrogative	$q(x) \leftarrow Knows(x, Roger)$
Does Julian rule?	interrogative	$q() \leftarrow Rules(Julian)$

4.1 Constraining Parse Trees

The compositional translation ϕ works as follows. First, higher order logic (HOL) expressions are associated to the words both of the function and of the content lexicon. Then, we mirror the composition of syntactic components in the parse tree with lambda application and reduction, ultimately yielding a FOL meaning representation at the sentence level, following the pattern set by Clifford in [9]:



In this parse tree of the Y/N-question "Is James a Man?", the feature structure *sem* returns the current values of the compositional translation ϕ at each node of the tree: The MR (the feature *mr*), a CQ of type t, is computed w.r.t. a semantic type (the feature *type*) and a context (the feature *con*), following the usual HOL assumptions, advocated since Montague, regarding the type associated to every NL component – like $(e \to t) \to t$, i.e., a function from properties onto truth values, for NPs or $e \to t$, a property or characteristic function, for Ns. The fact that the root context is empty implies that the whole expression (i.e., the MR) is well-typed (cf. [8]), ensuring the termination of the translation procedure.

The feature structure *syn*, on the other hand, returns the category (*cat*), the position (*pos*, which can be bound to two values, *subj*, subject, and *pred*, predicate) and the kind of utterance (*ass*, bound to *tbox*, *abox* and *cq*, i.e., to ABox and TBox assertions and CQs) of every non terminal component and of every function word. The *gap* feature associated to NPs indicates whether it contains or not the trace of a wh-movement (in the case of subordiate clauses) and the *coord*, if it contains a conjunction. A *voice* (for voice) feature, that can be set to *act* (active voice) or *pass* (active voice), is associated to VPs.

The grammar we have been using so far is a unification-based phrase structure grammar (UPSG), where nested feature structures comprising both syntactic and semantic features are associated to (and computed w.r.t.) each component. They have the advantage that parsing is based on constraint satisfaction and that it is thus easy to set constraints on components by means of semantic and syntactic features.

4.2 Expressive Power of the Declarative Fragment

We have studied the expressivity of the declarative fragment by comparing DL-Lite$_R$'s expressive power to that of the sets of MRs of the fragments COP and COP+DTV+TV, which we will denote, rspectively, Λ_{COP} and $\Lambda_{\mathsf{COP+DTV+TV}}$. This is based on the following standard model-theoretic characterization of the expressive power of a FOL fragment:

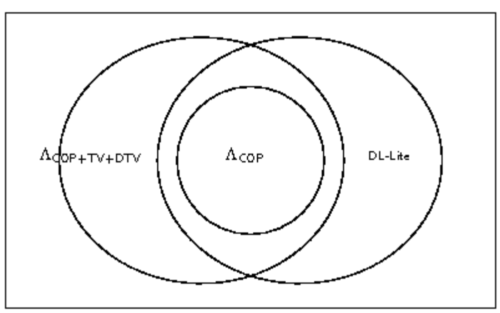
Definition 4.1. (Expressive Power) The *expressive power* of a fragment Λ of FOL over a signature $\mathcal{L}$ is defined in terms of the class $\mathcal{K}_\Lambda$ of the FOL *models* or *intepretation structures* over $\mathcal{L}$.

A FOL fragment Λ' is said to be *as expressive as* or to *contain* a fragment Λ if $\mathcal{K}_\Lambda \subseteq \mathcal{K}_{\Lambda'}$ – i.e., if it can express all of the models of Λ. If $\mathcal{K}_\Lambda \cap \mathcal{K}_{\Lambda'} \neq \emptyset$, they are said to *overlap*. Armed with these definitions, we can state the main results regarding Lite English's expressive power:

Theorem 4.1. *Lite English is as expressive as* COP *if we drop* DL-Lite*'s unique name assumption, but the converse is false.*

Theorem 4.2. *Lite English and* COP+TV+DTV *overlap w.r.t. expressive power but neither is as expressive as the other.*

We will not give the proofs here, which the reader can find in [2, 15]. The general picture can be summarized as follows:



The most interesting consequence of these results is the light they shed on the NL constructs Lite English covers, in particular, the use of relatives in the declarative fragment. We can summarize the relevant features of Lite English declarative sentences as follows:

- **(Quantification)** In $\mathsf{DL\text{-}Lite}_R$ and *a fortiori* in Lite English, a universal quantifier can be followed by $n \geq 0$ existential quantifiers. This is not the case in COP+TV+DTV. In this fragment, quantifiers may occur in any order whatsoever. Statements like "some woman loves every man" are in COP+TV+DTV but not in Lite English.

- **(Negation)** As the reader may recall from the definition of ABox assertions, $\mathsf{DL\text{-}Lite}_R$ precludes negated facts (i.e. negated ABox assertions). Negation in DL-Lite can moreover only occur on right hand side TBox concepts. But in COP and *a fortiori* in the fragment COP+TV+DTV this is possible – statements like "Julian is not an emperor" are in COP but not in Lite English. Negation is thus highly controlled.

- **(Relatives)** English Lite covers constructs that neither the fragment COP nor COP+TV+DTV can cover without yielding a state blowup. Lite English supports, for instance, relative clauses, without compromising tractability, as it happens with FLs (recall Pratt and Third's table above). Lite English outrules an unrestricted (i.e., uncontrolled) use of negation, which can only occur within the *predicate* of a *general* statement, mirroring $\mathsf{DL\text{-}Lite}_R$.

- **(Relations)** Lite English covers only transitive verbs (binary relations), even if, in priciple, nothing prevents us from extending coverage to ditransitive verbs: the properties of $\mathsf{DL\text{-}Lite}_R$ hold for n-ary relations (cf. [6]). However, due to the restricted behavior of quantifiers a good many properties of relations that can be expressed by, say, COP+TV+DTV, will not be expressible by $\mathsf{DL\text{-}Lite}_R$.

5 Conclusions and Further Work

We have studied the problem of managing structured data with natural language and propounded a controlled language (Lite English) approach to it based on description logics, $\mathsf{DL\text{-}Lite}_R$. We have shown how this language, or at any rate its declarative fragment, inherits the expressivity of $\mathsf{DL\text{-}Lite}_R$. It is thus, in theory, suitable for carrying on with data management tasks respecting their tight expressivity bounds. We have furthermore compared the expressive power of Lite English with that of the tractable fragments of English, singling out the NL constructs relevant to ontology-driven data access.

The main issue on which we are working now is that of applying the same techniques to the interrogative fragment, whose expressive power corresponds to that of conjunctive queries. The contribution of the NL contructs we enounter in questions to expressive power and computational complexity is still, as far as we know, an open question.

Bibliography

[1] I. ANDROUSTOPOULOS, G. D. RICHTIE, and P. Thanish. Natural Language Interface to Databases - An Introduction. www.cs.washington.edu/homes/amp/iui03.pdf, 1995.

[2] Raffaella BERNARDI, Diego CALVANESE, and Camilo THORNE. Expressing DL-Lite Ontologies with Controlled English. In *Proceedings of the 20th International Workshop on Description Logics (DL-2007)*, 2007.

[3] Raffaella BERNARDI, Diego CALVANESE, and Camilo THORNE. Lite Natural Language. In *Proceedings of the 7th International Workshop on Computational Semantics (IWCS-7)*, 2007.

[4] Diego CALVANESE, Giusseppe DE GIACOMO, Domenico LEMBO, Maurizio LENZERINI, and Riccardo ROSATI. Efficiently Managing Data Intensive Ontologies. In *Proceedings of the 2nd Italian Semantic Web Workshop: Semantic Web Applications and Perspectives (SWAP 2005)*, 2005.

[5] Diego CALVANESE, Giusseppe DE GIACOMO, Domenico LEMBO, Maurizio LENZERINI, and Riccardo ROSATI. Data Complexity of Query Answering in Description Logics. In *Proceedings of the 10th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2006)*, 2006.

[6] Diego CALVANESE, Giusseppe DE GIACOMO, and Maurizio LENZERINI. Conjunctive Query Containment Under Description Logic Constraints. *ACM Trans. on Computational Logic*, 2007.

[7] Diego CALVANESE, Giusseppe DE GIACOMO, Guido VETERE, Maurizio LENZERINI, and Riccardo ROSATI. DL-Lite: Practical Reasoning for Rich DLs. In *Proceedings of the 2004 Workshop on Description Logics (DL 2004)*, 2004.

[8] Bob CARPENTER. *Type-Logical Semantics*. The MIT Press, 1997.

[9] James CLIFFORD. Natural Language Querying of Historical Databases. *Computational Linguistics*, 14:10–35, 1988.

[10] Franz BAADER, Diego CALVANESE, Daniele Nardi, Peter PATEL-SCHNEIDER, and Deborah MCGUINNESS. *The Description Logic Handbook*. Cambridge University Press, 2003.

[11] Norbert E. FUCHS, Kaarel KALJURAND, and Gerold SCHNEIDER. Attempto Controlled English Meets the Challenges of Knowledge Representation, Reasoning, Interoperability and User Interfaces. http://www.ifi.unizh.ch/attempto/publications, 2005.

[12] R. HULL, S. ABITEBOUL, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[13] Ian PRATT and Allan THIRD. More Fragments of Language. *Notre Dame Journal of Formal Logic*, 2005.

[14] John SOWA. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks Cole Publishing Co., 1999.

[15] Camilo THORNE. Controlled English for DL-Lite. Technical report, KRDB Research Centre, Free University of Bozen-Bolzano, 2007. http://www.inf.unibz.it/krdb/pub/index.php.