# Designing Efficient Controlled Languages for Ontologies

Camilo Thorne, Raffaella Bernardi, and Diego Calvanese

## 1 Introduction

The attempts made in the 70s and 80s to build natural language interfaces (NLIs) to information systems and databases turned into disappointments towards the 90s (Androutsopoulos et al., 1995). One of the reasons were the challenges posed by structural and semantic ambiguity in arbitrary natural language input. As a way to overcome the ambiguity problem, the *controlled natural language* (CL) paradigm was proposed (Hujsen, 1998; Kittredge, 2003; Sowa, 2004), to build NLIs where only a restricted fragment of a natural language can be used. An important area of application of such NLIs and CLs is to provide front-ends to ontologies and ontology-based systems. By an *ontology* we mean here a conceptualization of a domain of interest, expressed as a set of logical assertions. Specifically, ontologies formulated in variants of description logics (DLs) (Baader et al., 2003), which are fragments of first-order logics with well understood computational properties and for which logical reasoning (to, e.g., detect inconsistencies in a specification) is decidable and in significant cases also computationally tractable. DLs provide the formal underpinning for the Web Ontology Language OWL (Horrocks et al., 2003), which is the ontology specification language standardized by the W3C.[1]

In addition to NLIs to OWL ontologies (Schwitter and Tilbrook, 2006b; Kaljurand and Fuchs, 2006), systems have been proposed that, e.g., guide the user to formulate his/her natural language (NL) question via an ontology that incrementally shows the possible concepts that could be involved in the question (Franconi

Camilo Thorne
Free University of Bozen-Bolzano, Bolzano, Italy, e-mail: cthorne@inf.unibz.it

Raffaella Bernardi
University of Trento, Trento, Italy, e-mail: bernardi@disi.unitn.it

Diego Calvanese
Free University of Bozen-Bolzano, Bolzano, Italy, e-mail: calvanese@inf.unibz.it

[1] http://www.w3.org/TR/owl2-primer/

et al., 2010; Dongilli and Franconi, 2006). Others guide the user via an incremental parser (Bernstein et al., 2006; Damljanovic, 2010), or engage the user in clarification dialogs (Gunning et al., 2010).

The present chapter addresses the questions of *(1) which* should be the CL to be used to manage ontologies efficiently, and *(2) how* can it be defined. Specifically, our proposal is to determine a methodology for defining *exactly* those fragments with a *desirable computational complexity*. We use DLs as the starting point to answer question *(1)*, viz. which is the most suitable NL fragment, and categorial grammars (CGs) to provide an answer to question *(2)*, viz. how to capture the syntactic structures corresponding exactly to the semantic representations allowed by the chosen, efficient DL.

With respect to the kind of DL, we focus our attention on *DL-Lite*, which is a family of DLs studied in the context of ontology-based access to (relational) databases (Calvanese et al., 2007). *DL-Lite* is specifically optimized for efficient reasoning in the presence of large datasets, as in ontology-based systems the size of the data (stored in possibly very large triple stores or relational databases) largely dominates the size of the ontology's intensional descriptions. Indeed, in *DL-Lite*, reasoning is computationally tractable in general, and lower w.r.t. data alone. This can be contrasted with the computational properties of more expressive DLs, such as *SHOIN*, the DL underlying OWL, in which reasoning is computationally intractable. With respect to the CG, we define a grammar that relies on the sub-categorization of syntactic constituents to capture exactly the intended logic.

We exploit the syntax-semantics interface as realized by CGs to obtain *DL-Lite* meaning representations compositionally while parsing (van Benthem, 1987; Moortgat, 1997). To this end we consider of particular value the studies carried out by Pratt and Third in (Pratt and Third, 2006) who have investigated the satisfiability of sets of sentences in fragments of natural language and their computational complexity, but start instead from the logic (viz., an OWL fragment) as Fuchs and Kaljurand in (Kaljurand and Fuchs, 2006).

The rest of the chapter is structured as follows. In Section 2, we provide an overview of controlled languages and semantic complexity, highlighting the open questions that motivate our contributions. In Sections 3 and 4, we introduce respectively the DL and the grammar we work with. In Section 5, we describe in detail how CGs can capture exactly the desired fragments of natural language. In Section 6, we show how corpora analysis can be used to justify further CL design choices. In Section 7, we provide an overview of related work, in the form of related results on CLs obtained and published elsewhere by the authors, and in the form of other CLs for ontologies that have been proposed in the literature. Finally, in Section 8, we summarize our results and outline our ongoing work regarding the computational properties of controlled languages, both declarative and interrogative.

## 2 Controlled Languages and Semantic Complexity

A *controlled language* (CL) is a fragment of natural language such as English with a limited lexicon and a small set of grammar rules (Hustadt et al., 2004; Meng and Chu, 1999; Sowa, 1999). Importantly, CLs are engineered to handle natural language ambiguity, so that their utterances "compile", via, e.g., a rule-based, symbolic and compositional syntax-directed translation algorithm (in a way similar to programming languages' compilation (Aho et al., 1986)), in *unambiguous* logical axioms and/or queries, due to their restricted syntax and lexicon.

This tight integration with formal ontology and query languages gives rise to a more general phenomenon: the property of *semantic complexity* as defined and investigated by Pratt and Third in (Pratt and Third, 2006). Pratt and Third show that each (controlled) fragment of English generates a logic fragment: the set of its *meaning representations* (MRs); semantic complexity is then naturally defined as the computational complexity of reasoning with its MRs (i.e., the computational complexity of the associated satisfiability problem). Furthermore, they show that semantic complexity correlates to coverage by considering the impact that particular combinations of English constructs (negation, relatives, transitive verbs, etc.) have on semantic complexity, pinpointing combinations that are: *(i)* tractable (**PTime** semantic complexity), *(ii)* intractable (**NP**-hard semantic complexity), or *(iii)* undecidable.

In our work, we extend both their methodology and their results: their methodology, by using categorial grammars to "reverse engineer" English controlled fragments from logics that exhibit desirable computational properties; their results, by considering semantic *data complexity*, viz., the semantic complexity of CLs for ontologies measured only in the size of the (typically very large) data repositories they are meant to manage as opposed to the size of the complete logical specification derived from the natural language utterances. More precisely, we *(i)* consider logic constructs that give rise to tractable data complexity, *(ii)* pinpoint those structures of English that map (following Montagovian semantics) into those logic constructs, and *(iii)* propose grammars that generate such structures. In this way one can determine the best trade-off between coverage and tractability holding for NLIs to ontologies and ontology-based systems.

**Pratt and Third's Fragments of English.** The work of Pratt and Third provides hints on how to determine which fragments hold the right expressiveness for ontology-based systems, via their notion of semantic complexity. We give now a brief overview of their *controlled fragments of English* (Pratt and Third, 2006), which are subsets of standard English meant to capture some simple, albeit for our purpose important, structures of English.

The fragments of Pratt and Third are built incrementally, starting with copula, nouns, negation, and the universal and existential quantifiers, and extending later coverage to larger portions of English – relative constructions, ditransitive verbs, and anaphora, as summarized in Table 1. The fragments are named after such com-

**Table 1** Fragments of English studied by Pratt and Third (Pratt and Third, 2006).

| **Fragment** | **Coverage** | **Semantic Complexity** |
|---|---|---|
| COP | Copula, common, proper nouns, negation, universal and existential quantifiers | **PTime** |
| COP+TV+DTV | COP+transitive verbs (“reads”) + ditransitive verbs (“gives”) | **PTime** |
| COP+Rel | COP+relative pronouns (“who”, “that”, “which”) | **NP**-complete |
| COP+Rel+TV | COP+Rel+transitive verbs | **ExpTime**-complete |
| COP+Rel+TV+DTV | COP+Rel+TV+ditransitive verbs | **NExpTime**-complete |
| COP+Rel+TV+RA | COP+Rel+TV+restr. anaphora (“him”, “she”, “itself” with *bounded* anaphoric co-references) | **NExpTime**-complete |
| COP+Rel+TV+GA | COP+Rel+TV+gen. anaphora (*unbounded* anaphoric pronouns) | undecidable |

binations, COP if their sentences contain only the copula, COP+TV if they contain in addition transitive verbs, and COP+TV+DTV if they contain both transitive and ditransitive verbs. Further differences are due to the presence in the lexicon of the relative pronoun (Rel) and of anaphora in a general (GA) or restricted form (RA).

Each NL construct has a MR introducing an $n$-ary predicate or a logical operation in First Order Logic (FO): The MRs of relatives (e.g., “who”) introduce conjunction ($\wedge$); negations (e.g., “no”, “not”) introduce logical negation ($\neg$); intransitive verbs (e.g., “runs”) and nouns (e.g., “man”) correspond to unary predicates; transitive verbs (e.g., “loves”) correspond to binary predicates, and ditransitive verbs (e.g., “sells to”) to ternary predicates; universal quantifiers (“every”, “all”, “everyone”) to universal quantification ($\forall$), and existentials (“some”, “someone”) to existential quantification ($\exists$).

*Example 1.* COP and COP+TV+DTV generate English utterances such as:

(1) Some people are weak. $[\exists x(\mathit{People}(x) \wedge \mathit{Weak}(x))]$

(2) Every husband has a wife. $[\forall x(\mathit{Husband}(x) \rightarrow \exists y(\mathit{Wife}(y) \wedge \mathit{Has}(x,y)))]$

(3) Every salesman sells some item to some customer.
$[\forall x(\mathit{Salesman}(x) \rightarrow \exists y(\mathit{Customer}(y) \wedge \exists z(\mathit{Item}(z) \wedge \mathit{Sells}(x,z,y))))]$

Note that in (2) and (3) above, other translations might be possible due to NL ambiguity. However, these are discarded by the grammar, which follows only the surface order of constituents. ♣

**Boolean- and non-Boolean-closed fragments.** As shown in Table 1, where we report the results of (Pratt and Third, 2006), the most expressive fragment of English considered by Pratt and Third is undecidable. As a matter of fact, only the first two fragments, COP and COP+TV+DTV, are tractable, i.e, have **PTime** semantic complexity. Notice that as soon as we add rules dealing with the relative clause we lose tractability. COP+Rel (i.e., COP with relative clauses) is already **NP**-hard. This is because relatives express conjunctions which, together with negation, generate logics (i.e., fragments of FO) that contain the propositional calculus (for which reasoning is **NP**-complete). In other words, COP+Rel and all the fragments containing it are “Boolean-closed”, and allow negation to be freely combined with conjunction and relatives. Instead, COP and COP+TV+DTV are “non-Boolean-closed”. The challenge that we face here is to develop a methodology for defining tractable, “non-Boolean-closed” CLs that capture tractable ontology languages.

## 3 *DL-Lite* and its Computational Properties

Description logics (DLs) (Baader et al., 2003) are the logics, typically fragments of FO, that provide the formal underpinning to ontologies and the Semantic Web (Horrocks et al., 2003). They allow one to structure the domain of interest by means of *concepts*, denoting sets of objects, and *roles*, denoting binary relations between (instances of) concepts. Complex concept and role expressions are constructed starting from a set of atomic concepts and roles by applying suitable constructs. The domain of interest is then represented by means of a DL knowledge base, consisting of a TBox (for “terminological box”), storing intensional information, and an ABox (for “assertional box”), storing extensional information about individual objects of the domain of interest.

We focus our attention on *DL-Lite* (Calvanese et al., 2007), a family of DLs specifically tailored to manage large amounts of data efficiently. Specifically, we consider variants of *DL-Lite* in which the TBox is constituted by a set of *inclusion assertions* of the form

$$Cl \sqsubseteq Cr$$

where *Cl* and *Cr* denote concepts that may occur respectively on the left and right-hand side of inclusion assertions. The form of such concepts depends on the specific variant of *DL-Lite*. Here, we consider two variants, called $\mathit{DL\text{-}Lite}_{core}$ and $\mathit{DL\text{-}Lite}_{R,\sqcap}$ , which we define below. In fact, $\mathit{DL\text{-}Lite}_{core}$ represents a core part shared by all logics of the *DL-Lite* family.

**Definition 1 ($\mathit{DL\text{-}Lite}_{core}$ and $\mathit{DL\text{-}Lite}_{R,\sqcap}$).** In $\mathit{DL\text{-}Lite}_{core}$, *Cl* and *Cr* are defined as follows:[2]

$$Cl \longrightarrow B \mid \exists R \qquad\qquad Cr \longrightarrow B \mid \neg B \mid \exists R \mid \neg\exists R$$

where *B* denotes an atomic concept, and *R* denotes an atomic role. In $\mathit{DL\text{-}Lite}_{R,\sqcap}$, in addition to the clauses of $\mathit{DL\text{-}Lite}_{core}$, we have also:

[2] We have omitted *inverse* roles $R^-$ from the DLs to simplify the presentation of the main idea we are investigating.

$$\begin{aligned} \mathit{MScStudent} &\sqsubseteq \mathit{Student} \\ \mathit{MScStudent} &\sqsubseteq \mathit{Works} \\ \mathit{MScStudent} &\sqsubseteq \neg \mathit{BScStudent} \\ \exists \mathit{Reads} &\sqsubseteq \mathit{Works} \\ \mathit{Student} &\sqsubseteq \exists \mathit{Reads} \end{aligned} \qquad \begin{aligned} \mathit{Student} \sqcap \mathit{Busy} &\sqsubseteq \mathit{Works} \\ \mathit{Student} \sqcap \exists \mathit{Reads} &\sqsubseteq \mathit{Works} \\ \exists \mathit{Reads} \sqcap \exists \mathit{Writes} &\sqsubseteq \mathit{Works} \\ \mathit{Student} &\sqsubseteq \exists \mathit{Reads}.\mathit{Book} \end{aligned}$$

**Fig. 1** An example *DL-Lite*$_{core}$ TBox (left part), and some additional *DL-Lite*$_{R,\sqcap}$ assertions (right part).

$$Cl \longrightarrow Cl_1 \sqcap Cl_2 \qquad\qquad Cr \longrightarrow \exists R.B$$

where $R$ denotes again an atomic role. ♠

The $\sqcap$ construct denotes conjunction, and $\neg$ negation (or complement). The $\exists R$ construct is called *unqualified existential quantification*, and intuitively denotes the *domain* of role $R$, i.e., the set of objects that are connected through role $R$ to some (not further specified) object.[3] Finally, the $\exists R.Cr$ construct, called *qualified existential quantification*, allows one to further qualify the object connected through role $R$ as an instance of concept $Cr$.

As an example, consider the *DL-Lite*$_{core}$ TBox depicted in the left part of Figure 1, which makes use of various concepts (*Student*, *MScStudent*, *BScStudent*, *Works*) and roles (*Reads*, *Writes*) to express some simple knowledge about the student domain. Specifically, the TBox assertions state that every MSc-student is a student, that MSc-students work, and that no MSc-student is a BSc-student, i.e., the two concepts are *disjoint*. Note that in *DL-Lite*, negation is used only to express disjointness, as in the statement in Figure 1. Additionally, making use of unqualified existential quantification, we can express that everyone who reads something (i.e., is in the domain of the *Reads* role) works, and that every student reads something. The latter is also called a *participation constraint*, since it forces instances of *Student* to participate to the *Reads* role. In the right part of Figure 1, we have shown also some *DL-Lite*$_{R,\sqcap}$ inclusion assertions, which make use of conjunction in the left-hand side to express that busy students work, that students who read something work, and that everyone who reads something and writes something works. Finally, to express that every student reads some book, we can make use of qualified existential quantification (allowed to appear only in the right-hand side of inclusion assertions).

To formally specify the semantics of *DL-Lite*, we provide its standard translation to FO. Specifically, we map each concept $C$ (we use $C$ to denote an arbitrary concept, constructed applying the rules above) to a FO formula $\varphi(C,x)$ with one free variable $x$ (i.e., a unary formula), and each role $R$ to a binary formula $\varphi(R,x,y)$ as follows:

[3] Instead, $\exists R^-$, for an inverse role $R^-$, denotes the *range* of role $R$.

**Table 2** Combined complexity and data complexity of consistency in different DLs.

| DL | Combined Complexity | Data Complexity |
|---|---|---|
| *DL-Lite*$_{core}$ | in **NLogSpace** | $\mathbf{AC}^0$ |
| *DL-Lite*$_{R,\sqcap}$ | **PTime**-complete | $\mathbf{AC}^0$ |
| *ALC* | **ExpTime**-complete | **coNP**-complete |
| *SHOIN* | **NExpTime**-hard | **coNP**-hard |

$$\begin{array}{rclcrcl} \varphi(B,x) &=& \mathtt{B}(x) & \qquad & \varphi(R,x,y) &=& \mathtt{R}(x,y) \\ \varphi(\neg C,x) &=& \neg\varphi(C,x) & & \varphi(\exists R,x) &=& \exists y \varphi(R,x,y) \\ \varphi(C_1 \sqcap C_2,x) &=& \varphi(C_1,x) \land \varphi(C_2,x) & & \varphi(\exists R.C,x) &=& \exists y \varphi(R,x,y) \land \varphi(C,y) \end{array}$$

In the translation of $\exists R.C$, the variable $y$ is considered to be a fresh variable. An inclusion assertion $Cl \sqsubseteq Cr$ of the TBox corresponds then to the universally quantified FO sentence $\forall x.\, \varphi(Cl,x) \to \varphi(Cr,x)$.

We observe that the above translation actually generates a formula in the guarded fragment of FO (Grädel, 1999a). This holds not only for *DL-Lite* but for many other expressive DLs as well, and accounts for the good computational properties of such logics (Grädel, 1999b).

Finally, in *DL-Lite*, an ABox is constituted by a set of *assertions on individuals*, of the form $B(a)$ or $R(a,b)$, where $B$ and $R$ denote respectively an atomic concept and role, and $a$, $b$ denote constants. As in FO, each constant is interpreted as an element of the interpretation domain. The above ABox assertions correspond to the analogous FO facts, or, by resorting to the above mapping, to $\varphi(B,x)(a)$ and $\varphi(R,x,y)(a,b)$, respectively. A *DL-Lite knowledge base* is simply a pair $(Tbox, Abox)$, where *Tbox* is a TBox and *Abox* an ABox. A *model* of such knowledge base is a FO interpretation in which the (closed) FO formulae resulting from the translation of all assertions in $Tbox \cup Abox$ evaluates to true.

To study efficiency we consider the computational reasoning problems relevant to DL ontologies and knowledge bases. The key problem, to which most other ones can be reduced, is the problem of *knowledge base consistency*, in which, given a knowledge base $(Tbox, Abox)$, we ask whether it has a model. Following Vardi in (Vardi, 1982), when we consider the computational complexity measured only in terms of the *size* of the ABox (defined as the number of constants the ABox contains), we speak about *data complexity*. When instead the complexity is measured in terms of the size of the whole input, we speak of *combined complexity*. A DL can be considered as "efficient" for ontology-based data management, whenever such complexity is *tractable* (in **PTime**).

It turns out that *DL-Lite*, and in particular *DL-Lite*$_{core}$ and *DL-Lite*$_{R,\sqcap}$, are "optimally efficient", in the sense that their data complexity is even lower. Indeed, relatively to consistency, the problem we are interested in this chapter, they are in $\mathbf{AC}^0$[4]

[4] The class $\mathbf{AC}^0$, is a complexity class strictly contained in (and hence easier than) **PTime**. SQL query evaluation in relational databases is in $\mathbf{AC}^0$ in data complexity, which accounts for the efficiency of database management systems in dealing with large amounts of data.

in data complexity and in **PTime** in combined complexity.[5] The DLs in the *DL-Lite* family are essentially the maximal DLs that exhibit such nice computational properties (Calvanese et al., 2007; Artale et al., 2009). This is a consequence of suitable syntactic restrictions that have been imposed in such logics:

- Concepts are not closed under Boolean operations: negation is restricted to basic concepts within the scope of a *right Cr*, and the use of disjunction is ruled out.
- Value restriction, a typical DL construct corresponding to a form of universal quantification, is not allowed, and the use of qualified existential quantification is restricted to the right-hand side of inclusion assertions.

These restrictions ensure that the *DL-Lite* logics are contained in the Horn fragment of FO, and therefore they admit a (in general infinite) *canonical model*, which can be considered as representative of all models as far as reasoning is concerned.

The *DL-Lite* constructs are nevertheless sufficiently expressive to cover the main features of conceptual modeling languages such as UML class diagrams and of concept hierarchies in ontologies and ontology-based systems. This is important, since it implies that in practice reasoning does indeed scale to very large ontologies that can capture several naturally arising domains of interest. This has to be compared with the much higher computational complexity of more expressive DLs. For illustration, consider in Table 2 the complexity of the DL *ALC*, which is the smallest logic containing the *DL-Lite* logics that we have considered here[6] and closed under Boolean operations. Both for *ALC* and for *SHOIN*, the DL that underpins OWL DL, reasoning is **coNP**-hard in data complexity, and provably exponential in combined complexity. Notice that as soon as a DL becomes closed under Boolean operations, it is intractable, and hence reasoning does not really scale well with data growth.

We are interested in studying the linguistic structures that correspond to the *DL-Lite* constructs. In what follows (Section 5 below), we will look at straightforward ways to express them in natural language.

## 4 Categorial Grammars

As most of the linguistically motivated formal grammars currently in use, categorial grammars (CGs) are a class (or family of classes) of lexicalised grammars, i.e., grammars where the lexicon carries most of the information about how words can be assembled to form grammatical structures. In this framework, syntactic categories are seen as *formulas* and their category forming operators as connectives, i.e., *logical constants*. In addition, the Curry-Howard correspondence ensures the

[5] Notice that Pratt and Third's complexity results do not distinguish between data and combined complexity.

[6] All *DL-Lite* logics include also the *inverse role* constructor, which cannot be captured in *ALC*. Moreover, some *DL-Lite* variants use (complex) role inclusions, which also would lead the logic outside the scope of *ALC*.

Montagovian homomorphism a.k.a. syntax-semantics interface between the (logical) calculus of syntactic categories and FO MRs (van Benthem, 1987).

The peculiarity of CGs is that word assembly is carried out by natural deduction logical rules (that take care of natural language syntax); such natural deduction rules are coupled with (via the Curry-Howard correspondence) $\lambda$-calculus operations dealing with the FO meaning assembly, via the intermediate $\lambda$-FO formalism, viz., FO extended with (typed) $\lambda$-calculus $\lambda$-abstractions and $\lambda$-applications. In so doing, CGs capture better and more elegantly the tight correspondence between syntax and semantics of NL and its fragments than other equivalent grammatical formalisms such as semantically-enriched context-free grammars or some simple kinds of definite clause grammars.

This aspect of the formalism significantly simplifies the implementation task, since one has to focus only on the construction of the lexicon and can rely on any existing parser for the calculus. Information both about the syntactic structure where the word could occur and its meaning are stored in the lexicon. As derivation or logical deduction rules, we use the product free version of the (non associative) Lambek calculus (Lambek, 1958; Moortgat, 1997)[7].

**Definition 2 (Term Labelled Lexicon, Categorial Grammar).** A (syntactic) *category A* is defined as follows

$$A \longrightarrow np \mid n \mid s \mid A_1 \backslash A_2 \mid A_2 / A_1$$

where *np* (*noun phrases*), *n* (*nouns*) and *s* (*complete sentences*) are *atomic* categories. Complex categories are built out of atomic categories by means of the directional *left* and *right* functional connectives $\backslash$ and $/$ ($A_1 \backslash A_2$, resp. $A_2/A_1$, applied to a category $A_1$ situated to its left, resp. its right, yield category $A_2$). We denote by $\textsc{Cat}$ the set of all such categories and by $\textsc{Atom}$ the set $\{np, n, s\}$.

We map each syntactic category $A$ to a (semantic) *type* $typ(A)$ as follows:

$$typ(np) = e; \quad typ(s) = t; \quad typ(n) = (e,t),$$
$$typ(A_1/A_2) = (typ(A_2), typ(A_1)); \quad typ(A_2 \backslash A_1) = (typ(A_2), typ(A_1)).$$

where the atomic types are *e* (*entities*) and *t* (*Booleans*), and $(\tau, \tau')$ denotes the *functional* type (the type of functions from $\tau$ into $\tau'$).

Given a set $\Sigma$ of natural language basic expressions (i.e., a natural language vocabulary), a *term labeled categorial lexicon* is a relation,

$$\textsc{Lex} \subseteq \Sigma \times (\textsc{Cat} \times \textsc{Term}) \text{ s.t. if } (w, (A, \alpha)) \in \textsc{Lex},\ \alpha \in \textsc{Term}_{typ(A)}$$

where $\textsc{Term}$ is the set of all lambda terms and $\textsc{Term}_{typ(A)}$ denotes the set of lambda terms whose type is mapped to the category $A$.

Given a term labeled lexicon $\textsc{Lex}$, a *categorial grammar* is any finite subset $G \subsetneq \textsc{Lex}$. ♠

[7] The lexicon we present in this article has been tested using the GRAIL parser (Moot, 1998), based on the Lambek calculus.

This constraint on lexical entries categories and terms enforces the following requirement: if the expression (or word) $w$ is assigned the syntactic category $A$ and the term $\alpha$, then the term $\alpha$ must be of a type appropriate for the category $A$. We assign lambda terms whose body is a FO formula, viz., $\lambda$-FO terms. We look at the determiner *every*, by means of example, since it has a crucial role in our grammar. The reader is referred to (Keenan and Faltz, 1985; Eyck, 1985) for an in-depth explanation of this example in particular and the relationships between CGs and $\lambda$-FO in general.

*Example 2 (Determiner).* The meaning of "every NOUN" (e.g., "every man") is the set of those properties that "every NOUN" (e.g., "man") has

$$[\![\text{every NOUN}]\!] = \{X \mid [\![\text{NOUN}]\!] \subseteq X\}.$$

In a functional perspective, the determiner "every" is seen as a two-argument function taking a noun and a verb phrase (a property) as arguments. The syntactic category expressing this functional view as well as word order is the following

$$(s/(np\backslash s))/n$$

where the $n$ is the first argument that must occur on the right of "every" and $np\backslash s$, i.e., a verb phrase, is its second argument to occur still on the right of "every NOUN" (viz. "every NOUN VERB_PHRASE"). The typed lambda term (according to generalized quantifier theory (Barwise and Cooper, 1980)) corresponding to this syntactic category and representing the set-theoretical interpretation given above is: $\lambda Y_{(e,t)}.\lambda X_{(e,t)}.\forall x_e(Y(x) \rightarrow X(x))$. In the following, we will not use types on lambda terms unless necessary. ♣

An important feature of CGs is their "parsing as deduction" approach, which reduces the problem of checking whether a linguistic string is grammatical to the problem of proving that the string is of a certain syntactic category. More precisely, instead of directly recognizing linguistic word strings $w_1 \cdots w_n$, we work on the corresponding set of Lambek calculus formulae: to each lexicon entry $(w_i, (A_i, \alpha_i))$, for $i \in \{1, \ldots, n\}$ we associate a (Lambek calculus) *sequent* $A_i \vdash A_i : \alpha_i$; thereafter, following the inference rules of the calculus, a proof (a tree-shaped derivation) of a sequent $\Gamma \vdash s : \phi$, with $\phi$ of type $t$ (a $\lambda$-FO formula) is constructed. More formally:

**Definition 3 (Recognized Language).** Given a categorial grammar $G$ the *language recognized by* $G$, denoted $L(G)$ is the set of all word strings $w_1 \cdots w_n$ such that the sequent $\Gamma \vdash s : \phi$, has a proof in the Lambek calculus; where $\Gamma$ consists of a set $\{A_1 : \alpha_1, \ldots, A_n : \alpha_n\}$ of pairs of categories and terms as defined in the term labeled lexicon $\{(w_i, (A_i, \alpha_i)) \mid i = 1, \ldots, n\}$, and $\phi$ is a $\lambda$-FO formula (a term of type $t$). ♠

As by-product of the derivation one derives also the MR of the structure assigned to the string, i.e., the $\lambda$-FO term $\phi$ which after reduction gives rise to a FO closed formula or sentence. As such, NLs (and fragments thereof) recognized by a CG that does not cover purely higher-order NL constructs such as, e.g., the second-order determiner “most”, can induce (in a way similar, though more general, to Pratt and Third’s fragments) a fragment of FO: the set of all the first-order MRs associated to its (grammatical) complete sentences. We will exploit this particular feature of the formalism to define a CL in the next section that generates the *DL-Lite* logics.

## 5 Lite English and its Grammar CG-lite

As we said, the goal of our methodology is to define CLs for ontologies that are *efficient*, i.e., tractable w.r.t. semantic data complexity. We propose to this end to define them vis-à-vis those ontology constructs that give rise to tractable data complexity. More precisely, we propose to identify English syntactic categories that lexically control the restrictions imposed by the *DL-Lite* constructs. Such categories will naturally induce a CG (i.e., a term-labeled categorial lexicon) expressing *exactly* the *DL-Lite* family of logics as described earlier. In this section we outline such syntactic categories and how they were obtained. We proceed in three steps. Firstly, we outline the key constraints to be satisfied for a CL to induce *DL-Lite*. Secondly, we provide a sample CG (a finite term-labeled lexicon). Thirdly, we describe the main features of the fragment thus generated. Notice also that the methodology proposed is not, per se, grammar dependent, since our CLs can be equally, although less succinctly and not as elegantly, defined using semantically-enriched context-free grammars as we did in (Thorne, 2010), Ch. 4.[8] We call the resulting CL Lite-English and its CG, CG-lite .

### *5.1 Fragment of Natural Language for DL-Lite*

The constraints expressed in the TBox are universally quantified FO sentences. They are of the form $Cl \sqsubseteq Cr$, which translates into FO as $\forall x.\varphi(Cl,x) \rightarrow \varphi(Cr,x)$ and can be expressed by the following NL sentence patterns:

(a) $[\text{Every } \underbrace{\texttt{NOUN}}_{Cl}] \ \underbrace{\texttt{VERB\_PHRASE}}_{Cr}$

(b) $[[\text{Everyone } \underbrace{[\text{who } \texttt{VERB\_PHRASE}]}_{Cl}] \ \underbrace{\texttt{VERB\_PHRASE}}_{Cr}]$

The determiner “every” and the quantifier phrase “everyone” play a crucial role in determining the linguistic structures that belong to the natural language fragment corresponding to a *DL-Lite* TBox. In the following, we zoom into the NOUN and VERB_PHRASE constituents. In other words, we spell out how *DL-Lite* *Cl* and *Cr* concepts can be expressed in English. In doing so, we follow Definition 1.

[8] We refer the reader to the appendix for the formal proofs of the claims made in this section.

First of all, a *Cl* or a *Cr* could be an atomic concept $A$. An atomic concept $A$ corresponds to a unary predicate, which following standard formal semantic theory can be expressed either by a noun such as "student" (see (4) below), or an intransitive verb such as "work" (see (5) below).

The introduction of negation $\neg A$ on atomic concepts $A$, however, can occur only in a *Cr* and can thus be expressed only by a *predicate* VERB_PHRASE such as "is not a BSc-student" (6), or "does not work" (7).

The introduction of the $\exists R$ in a *Cl* can be performed by means of the quantifier phrase "everyone" followed by the relative pronoun "who"( 9) (or by the conjunction that would correspond to the use of $\sqcap$ on the *Cl* part allowed in the *DL-Lite*$_{R,\sqcap}$ fragment, see (16) below).

(4) Every MSc-student is a student. [$\mathit{MScStudent} \sqsubseteq \mathit{Student}$]
(5) Every MSc-student works. [$\mathit{MScStudent} \sqsubseteq \mathit{Works}$]
(6) Every MSc-student is not a BSc-student. [$\mathit{MScStudent} \sqsubseteq \neg \mathit{BScStudent}$]
(7) Every BSc-student does not work. [$\mathit{BScStudent} \sqsubseteq \neg \mathit{Works}$]
(8) Everyone who learns works. [$\mathit{Learns} \sqsubseteq \mathit{Works}$]
(9) Everyone who reads something works. [$\exists \mathit{Reads} \sqsubseteq \mathit{Works}$]

On the other hand, the introduction of $\exists R$ on the *Cr* part corresponds to the use of a transitive verb followed by an existential quantifier phrase, "something" (10), and its negation to the use of "does not" to negate such construction (11).

(10) Every student reads something. [$\mathit{Student} \sqsubseteq \exists \mathit{Reads}$]
(11) Every student does not read something. [$\mathit{Student} \sqsubseteq \neg\exists \mathit{Reads}$]

Note that, as the *DL-Lite* clause shows, the only reading of the ambiguous sentence in (11) is the one with *every* having wide scope and *something* being in the scope of *not*[9].

Also, the VERB_PHRASE, in (a) and the second VERB_PHRASE in (b) (i.e., the VERB_PHRASE of the main clause expressing a *DL-Lite Cr* concept) can be of any of the structures in (4)–(11). On the other hand, the first VERB_PHRASE in (b) (i.e., the VERB_PHRASE of the relative clause expressing a *DL-Lite Cl* concept) cannot contain negation: for it only the cases 5-4 above hold.

When we move to *DL-Lite*$_{R,\sqcap}$, the addition of the conjunction in the *Cl* corresponds to the use of adjective (12), or relative clauses modifying the noun quantified by "every" (13-15), or the "and" coordinating two VPs (16).

(12) Every nice student works. [$\mathit{Student} \sqcap \mathit{Nice} \sqsubseteq \mathit{Works}$]
(13) Every student who learns works. [$\mathit{Student} \sqcap \mathit{Learns} \sqsubseteq \mathit{Works}$]
(14) Every student who is a BSc-student works. [$\mathit{Student} \sqcap \mathit{BScStudent} \sqsubseteq \mathit{Works}$]
(15) Every student who reads something works. [$\mathit{Student} \sqcap \exists \mathit{Reads} \sqsubseteq \mathit{Works}$]

[9] For ease of explanation we do not consider the distinction between *something* and the negative polarity item *anything*. This distinction could be incorporated into the fragment, as studied in (Bernardi, 2002).

(16) Everyone who reads something and writes something works.
[$\exists Reads \sqcap \exists Writes \sqsubseteq Works$]

Furthermore, the introduction of the qualified existential on the *Cr* is performed by the determiner "a" (17).

(17) Every student reads a book. [$Student \sqsubseteq \exists Reads.Book$]

**Non-Boolean-closedness (tractability) of the fragment.** An important point to emphasize is the presence of the relative pronoun in the above fragment of sentences. Pratt and Third have shown how the uncontrolled use of such expression leads to **NP**-complete fragments when allowing the use only of the copula, or even to **ExpTime**-completeness when adding transitive verbs. Below, we will show how relative pronouns can be used in a controlled grammar while preserving tractability of inferences.

## 5.2 *Expressing DL-Lite$_{core}$*

We start again by looking at the main syntactic constraints over *DL-Lite$_{core}$* concepts and consider, in particular, the two constraints regarding the use of negation:

1. negation of atomic concepts can occur in a *Cr* but not in a *Cl*: $Cl \longrightarrow B$, $Cr \longrightarrow B \mid \neg B$;
2. an unqualified existential can occur both in a *Cl* and a *Cr*, but its negation can occur only in *Cr*s: $Cl \longrightarrow \exists R$, $Cr \longrightarrow \exists R \mid \neg\exists R$.

As we anticipated before , *Cl* and *Cr* concepts correspond respectively to the so-called "restrictive scope" (the subject `NOUN` constituent), and "nuclear scope" (the predicate `VERB_PHRASE` constituent) of the sentence-building `DET` *every*. We need to constrain the linguistic structures that occur within them. In particular, we need to block the occurrences of negation within *Cl*s and express the fact that `NOT` cannot outscope any `VERB_PHRASE` that occurs within the restrictive scope of the determiner *every*. As emphasized in (Bernardi, 2002), in CGs scope is determined by the sentential categories *s* that arise from complex CG syntactic categories. Different (possibly mutually exclusive) scope distributions can be enforced by multiplying sentential categories via *sentential levels*, and exploiting the derivability relations (and restrictions) among CG categories. It suffices to provide the intuition behind the proposed solution without going into its details: a complex category $A_1 \backslash A_2$, can be applied to either category $A_1$ or to a category $A_3$ that derives $A_2$ ($A_3 \Rightarrow A_1$). In our case, $\Rightarrow$ is the *derivability* relation of the logical grammar we use.

We mark the structures that express *DL-Lite Cl*s and *Cr*s and those that are negative or positive, by means of the four *sentential levels* $s_{cl}$, $s_{cr}$, $s_{\neg}$, and $s$, respectively, and establish the derivability relation below (we rule out any other derivability relations between atomic categorical formulae).[10] These stential levels state that a negated sentence can be in the *Cr* construct ($s_\neg \Rightarrow s_{cr}$) while it cannot be in the *Cl* part ($s_\neg \not\Rightarrow s_{cl}$) and a positive sentence can be in both ($s \Rightarrow s_{cl}, s \Rightarrow s_{cr}$):

$$s_\neg \not\Rightarrow s_{cl}, \quad s_\neg \Rightarrow s_{cr}, \quad s \Rightarrow s_{cl}, \quad s \Rightarrow s_{cr}, \quad \text{and} \quad s_{cl} \not\Leftrightarrow s_{cr}.$$

Note that this induces a derivability relation between complex categories built with or containing these atomic sentential categories; for instance, from $s \Rightarrow s_{cr}$ it follows that $np\backslash s \Rightarrow np\backslash s_{cr}$. Besides these sentential levels, as we will show below, we use two other sentential levels: one to mark TBox sentences ($s_{tb}$) and one to mark constituents built by the relative pronoun who ($s_{who}$). All the constraints on these sentential levels are lexically anchored by means of the lexical assignments below.

*Example 3 (Lexicon for DL-Lite$_{core}$).* The lexicon entries to use are as below[11]. The content words (intransitive verbs and nouns) are only given by way of example.

- Every $\in (s_{tb}/(np\backslash s_{cr}))/n$: $\lambda X.\lambda Y.\forall x.(X(x) \rightarrow Y(x))$
- is a $\in (np\backslash s)/n$: $\lambda X.\lambda z.X(z)$
- is not a $\in (np\backslash s_\neg)/n$: $\lambda X.\lambda z.\neg X(z)$
- does not $\in (np\backslash s_\neg)/(np\backslash s)$: $\lambda X.\lambda z.\neg X(z)$
- works $\in np\backslash s$: $\lambda z.\mathtt{Works}(z)$
- learns $\in np\backslash s$: $\lambda z.\mathtt{Learns}(z)$
- student $\in n$: $\lambda z.\mathtt{Student}(z)$
- MSc-student $\in n$: $\lambda z.\mathtt{MScStudent}(z)$
- BSc-student $\in n$: $\lambda z.\mathtt{BScStudent}(z)$
- everyone: $(s_{tb}/(np\backslash s_{cr}))/(np\backslash s_{who})$: $\lambda X.\lambda Y.\forall x.(X(x) \rightarrow Y(x))$
- who: $(np\backslash s_{who})/(np\backslash s_{cl})$: $\lambda P.\lambda z.P(z)$
- something: $((np\backslash s_\exists)/np)\backslash(np\backslash s)$: $\lambda Z.\lambda y.\exists x.Z(y,x)$
- reads: $(np\backslash s_\exists)/np$: $\lambda x.\lambda z.\mathtt{Reads}(z,x)$ ♣

**A. Using universal quantification to express concept subsumption.** Notice that in Example 3 the categories assigned to *every* and *everyone* rule out the possibility for them to occur in object position –they can only be in subject position. Moreover, since they are the only entries yielding a TBox sentence ($s_{tb}$), only sentences starting with them will be considered as grammatical. The negation brings sentences to the negative sentential level, and once they are there, they are blocked from occurring in the restrictive scope of *every* and *everyone*.

[10] We actually use residuated unary operators to carry out these derivability relations (Kurtonina and Moortgat, 1995) exploiting their logical properties: $\Diamond_j\Box_j s \Rightarrow s \Rightarrow \Box_i\Diamond_i s$ etc. Examples of residuated unary operators are "possibility in the past" and "necessity in the future".

[11] Notice, in the present work we do not handle features of any sort (morphological etc). Their usage will make the lexical entries more complex but won't have any effect on the main idea we are presenting.

**B. Using existential quantification, relatives, and conjunction to express existentially qualified roles and their conjunctions.** Since in the fragment described by Example 3, we do not have the $\sqcap$ on the *Cl*, the introduction of the unqualified existential $\exists R$ in it can be performed only by means of the quantifier *everyone* followed by the relative pronoun "who" and a transitive verb composed with *something*. The introduction of $\exists R$ on the *Cr* corresponds to the use of a transitive verb followed by an existential quantifier, *something*. The lexical entries for *everyone*, *who*, *something*, and *reads* above account for these facts. The need of the $s_{who}$ categories is due to the fact that *everyone* must be followed by a relative clause, i.e., sentences like *everyone left* or *everyone walks and speaks* cannot be part of the grammar. Similarly, transitive verbs can occur on the *Cr* part but only if followed by *something*, hence we use the category $s_{\exists}$ to guarantee this requirement.[12] Finally, the category assigned to "something" is such that it can occur only in object position.

**C. Controlling the behavior of negation.** As the reader can see, negation in Example 3 can only occur within a VERB_PHRASE expressing a *Cr*. The reader can gain a better understanding of the mechanisms involved by checking how our sample lexicon, combined with the constraints CG-lite imposes over its sentential levels, ensures the ungrammaticality of the sentences below (blocked by $s_{\neg} \not\Rightarrow s_{cl}$). Such sentences generate MRs that are not *DL-Lite* expressible:

(18) Everyone who does not read something works $[\neg\exists Reads \sqsubseteq Works]$

(19) Everyone who is not a BSc-student works. $[\neg BScStudent \sqsubseteq Works]$

**D. Expressing ABoxes.** The fragment of sentences whose meaning representation belongs to a $DL\text{-}Lite_{core}$ ABox is rather easy to build since an ABox consists of a conjunction of (ground) unary and binary logical atoms. In other words, the lexicon is built only with nouns, intransitive verbs, the copula (i.e., unary predicates), transitive verbs (i.e., binary predicates), individual names and adjectives.

## 5.3 Expressing $DL\text{-}Lite_{R,\sqcap}$

We now move to $DL\text{-}Lite_{R,\sqcap}$, and account for the following additions

1. conjunctions are allowed in *Cl*s: $Cl \longrightarrow Cl_1 \sqcap Cl_2$;
2. the qualified existential can occur in *Cr*s: $Cr \longrightarrow \exists R.B$.

*Example 4 (Lexicon extension for $DL\text{-}Lite_{R,\sqcap}$).* In order to move to $DL\text{-}Lite_{R,\sqcap}$, we need to add into the lexicon the following lexical entries. The (intersective, qualitative) adjective *nice* is given only by way of example.

- nice: $n_{cl}/n_{cl}$, $\lambda X.\lambda z.(X(z) \wedge \mathtt{Nice}(z))$

[12] Since we have neither $np$ nor $np/n$ entries we could also avoid the use of this extra sentential level $s_{\exists}$ in the sample example we are considering.

- who: $(n_{cl} \backslash n_{cl})/(np \backslash s_{cl})$: $\lambda X.\lambda Y.\lambda z.(X(x) \wedge Y(z))$
- and: $((np \backslash s_{cl}) \backslash (np \backslash s_{cl}))/(np \backslash s_{cl})$: $\lambda X.\lambda Y.\lambda z.(X(z) \wedge Y(z))$
- a: $(((np \backslash s_{\exists})/np) \backslash (np \backslash s_{cr}))/n$: $\lambda Y.\lambda Z.\lambda y.\exists x.(Z(y,x) \wedge Y(x))$

Again, we use sentential levels to control the occurrence of these constructs. The extended lexicon accounts also for the structures in (12)–(17). ♣

**A. Controlling the interaction of conjunction and negation.** Notice the need of having a conjunction operating at the sentential level $s_{cl}$: this blocks the composition of negation (*does not*) with a verb phrase built with an *and* that would wrongly give or recognize: *does not walk and speak* with *not* outscoping *and*; such constituent would yield the MR $\lambda z.\neg(\texttt{Walk}(z) \wedge \texttt{Speak}(z))$ that is not *DL-Lite* expressible, and would moreover give rise to intractable data complexity (Calvanese et al., 2007). For similar reasons we have to block the composition of *is not a* with a noun phrase built using an intersective adjective. The resulting NOUN_PHRASE constituent would yield non-*DL-Lite*-expressible $\lambda$-FO formulas where negation outscopes conjunction; e.g., a phrase like *is not a nice student* with MR $\lambda z.\neg(\texttt{Nice}(z) \wedge \texttt{Student}(z))$. The introduction of the category $n_{cl}$ with $n \Rightarrow n_{cl}$ makes such phrases ungrammatical.

**B. Qualified existential restrictions and recursive constituents.** We have considered a DL, *DL-Lite*$_{R,\sqcap}$, with qualified existentials of the form $\exists R.A$. Hence the argument taken by the determiner *a* can only be a bare noun *n*. Finally, notice that the lexical entries for the adjective, conjunction and qualified existential bring recursion into the language.

## 6 Distribution of Boolean- and non-Boolean-closed Fragments

As we have shown, reverse-engineering efficient CLs from ontologies is a promising path. Further, as shown in (Thorne, 2010), our methodology can be easily extended to define interrogative CLs with tractable data complexity. The question however remains as to how to identify CLs that, while enjoying the properties we desire them to have (express ontology and query languages, give rise to at most **PTime** data complexity), remain appealing to users.

In this section we propose a distributional methodology which may help in identifying desirable English constructs by focusing on their frequency in both interrogative and declarative English corpora. We believe that this method can yield techniques to pinpoint, in particular, CLs that may offer good trade-offs between coverage and semantic complexity. The intuition behind being that when we trade-off language coverage for performance (i.e., to attain tractable data complexity) in CLs, it makes sense to cover constructs that are frequently used and thus preferred by speakers. Specifically, we study the co-occurrence of crucial (for semantic complexity) logic constructs: negations, conjunctions, disjunctions, and universal and existential quantification, in English questions and sentences.

**Table 3** Corpora analyzed in this chapter.

| Corpus | Size | Domain | Sentence type |
|---|---|---|---|
| Brown corpus subset | 19,741 sentences | Open (news) | Declarative[17] |
| Geoquery corpus | 364 questions | Geographical | Interrogative |
| Clinical questions | 12,189 questions | Clinical | Interrogative |
| TREC 2008 | 436 questions | Open | Interrogative |

To obtain a representative sample we considered corpora of multiple domains and with sentences of arbitrary type (declarative and interrogative), since, when managing an ontology and/or an ontology-based system, users are required not only to assert but also to update and query (intensional and extensional) information belonging to different domains. We thus considered: *(i)* a subset (A: press articles) of the Brown corpus[13]; *(ii)* a subset of one (Geoquery880) of the Geoquery corpora[14]' *(iii)* a corpus of clinical questions[15]; and *(iv)* a sample from the TREC 2008 corpus[16]. Table 3 summarizes their main features.

To this end we exploited the availability of wide-coverage (statistical) deep semantic parsers such as Boxer (Bos, 2008), that output first-order MRs. We checked, for each such MR, the co-occurrence of a subset of the set $\{\forall, \exists, \neg, \wedge, \vee\}$ of FO operators (and only of that subset). Each such subset identifies MRs belonging, modulo logical equivalence, to a distinct fragment of FO. For instance, the combination $\{\forall, \exists, \wedge, \vee\}$ identifies MRs from the so-called positive fragment of FO. But it also identifies the class of corpora sentences that give rise to such MRs, and approximates the (controlled) fragment whose formal semantics may induce such FO fragment. Finally, with these considerations in mind, we observed the distribution of:

1. "Boolean-closed" fragments, viz.: $\{\exists, \wedge, \neg\}$, $\{\exists, \wedge, \neg, \forall\}$, $\{\exists, \wedge, \neg, \forall, \vee\}$, $\{\neg, \forall\}$, $\{\exists, \wedge, \forall\}$ and $\{\exists, \wedge, \forall, \vee\}$.
2. "Non-Boolean-closed" fragments, viz.: $\{\exists, \wedge\}$ and $\{\exists, \wedge, \vee\}$.

By "Boolean-closed", we recall, we mean fragments expressive enough to encode Boolean satisfiability and which give rise to intractable semantic complexity. A "non-Boolean-closed" combination, by contrast, cannot express Boolean functions and gives rise only to tractable semantic complexity.

The pipeline of Boxer consists in the following three basic steps: *(i)* each part of speech in a sentence is annotated with its most likely (categorial grammar) syntactic category; *(ii)* the most likely of the resulting possible combinatorial categorial grammar derivations (or proofs) is computed and returned; and *(iii)* a neo-Davidsonian

[13] `http://nltk.googlecode.com/svn/trunk/nltk_data/index.xml`

[14] `http://www.cs.utexas.edu/users/ml/nldata/geoquery.html`

[15] `http://clinques.nlm.nih.gov/`

[16] `http://trec.nist.gov/`

[17] The sample contained only 36 questions.

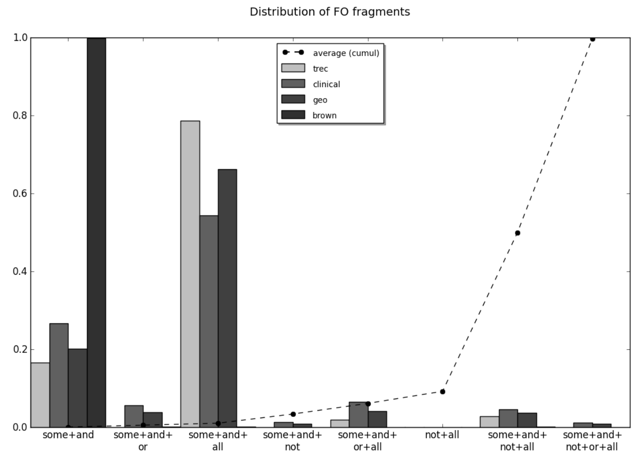

**Fig. 2** Relative frequency of *co-occurring* FO operators in sample corpora. Notice the distribution of "non-Boolean-closed" sentences.

semantically weakened[18] FO meaning representation is computed using discourse representation theory (DRT).

*Example 5.* When parsing Wh-questions from the TREC 2008 corpus such as "What is one common element of major religions?", Boxer outputs a FO semantic representation of the form

$$\exists y \exists z \exists e \exists u (\texttt{card}(y,u) \wedge \texttt{c1num}(u) \wedge \texttt{nnumeral1}(u) \wedge \\ \texttt{acommon1}(y) \wedge \texttt{nelement1}(y) \wedge \texttt{amajor1}(z) \wedge \texttt{nreligions1}(z) \wedge \\ \texttt{nevent1}(e) \wedge \texttt{rof1}(y,z))$$

where $\wedge$ and $\exists$ co-occur, but not $\vee$, $\neg$, or $\rightarrow$. ♣

Figure 2 shows the co-occurrence distribution obtained, expressed in terms of relative frequency (i.e., number of MRs per class/total number of MRs per corpus). As the figure shows, positive existential, $\{\exists, \wedge\}$ and $\{\exists, \wedge, \vee\}$, MRs occur quite frequently. Also, it seems that the same holds for sentences expressing universal quantification whereas the opposite is true for negation (low frequency overall).

This analysis can be compared to the more linguistics-based methodology followed in (Bernardi et al., 2007), in which we analyzed the distribution in (solely) interrogative corpora of classes of *logical words* which express FO operators, e.g., "all", "both", "each", "every", "everybody", "everyone", "any", "none", "nothing". See Figure 3.

[18] In this settings, the semantics of verbs is represented in terms of events connected via thematic roles to verb arguments (agents, themes, etc.). In addition, the semantics of non-FO constructs such as "most" is weakened to some FO representation.

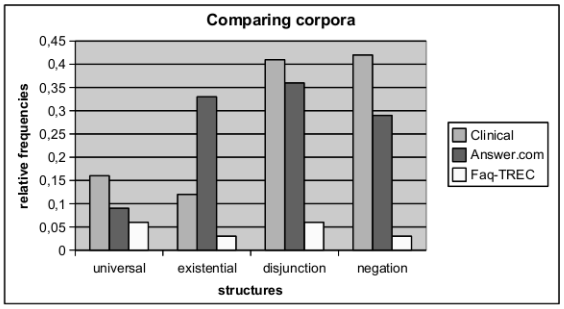

**Fig. 3** Relative frequency of FO operators in question corpora (Bernardi et al., 2007).

While Boxer is considered to have a reasonably good performance (covering over 95% of English, with approx. 75% accuracy) only when parsing and semantically interpreting declarative sentences and corpora, but not necessarily so over interrogative sentences and corpora, and commit us to a neo-Davidsonian semantics, we believe the results obtained are interesting. In particular, they may suggest that, while users use negation or disjunction words as frequently as conjunction and existential words, and all these more than universal words, when combining them *within* sentences "non-Boolean-closed" combinations are preferred.

## 7 Related Work

The work described in this chapter has been complemented by related results obtained by the authors and published elsewhere. In particular, we have applied and generalized the methodology defined in this chapter to determine which are the tractable (i.e., at most **PTime** in data complexity) fragments of ACE-OWL and the intractable (i.e., at least **NP**- or **coNP**-hard in data complexity) fragments (Thorne and Calvanese, 2012). Table 4 summarizes what these results mean in terms of language coverage, viz., which *maximal* combinations of (English) function and content words give rise to *tractable* ("non-Boolean closed") controlled fragments, and which *minimal* combinations give rise to *intractable* ("Boolean closed") controlled fragments.

Intractability arises with any combination capable of expressing full Boolean negation ("not") and full Boolean conjunction (conjunction, relative pronouns). Note that the good computational properties of Lite-English depend ultimately on the fact that, while expressing Boolean conjunction, it cannot express full Boolean negation, but rather a very limited form of it. We observe that a similar analysis carried out on Pratt and Third's fragments (and extending their own) by us in (Thorne, 2011; Thorne and Calvanese, 2009a) yielded similar results.

**Table 4** Non-“Boolean-closed” and “Boolean-closed” controlled English constructs.

| Complexity | Constructs |
|---|---|
| Tractable | Negation (“not”) in a predicate VERB_PHRASE<br>Relatives (“who”, “which”) everywhere<br>Conjunction (“and”) everywhere<br>Transitive verbs (“loves”) everywhere<br>Existential quantification (“some”) everywhere |
| Intractable | Negation (“not”) in a subject NOUN<br>Universal quantification (“only”) in a subject NOUN<br>Disjunction (“or”) in predicate VERB_PHRASE |

We have also applied the methodology used in this chapter to study controlled fragments of English *questions* for which the data complexity of reasoning (or evaluation) against an ontology, authored using any of their fragments, is tractable; see (Thorne, 2010), Ch. 5, and (Thorne and Calvanese, 2009b). This work shows that positive questions (questions built with “some”, relative pronouns, conjunction and eventually, disjunction) and restricted to proper and common nouns, intransitive and transitive verbs as content words, give rise to tractability. It also shows that they can be enriched with so-called *aggregate determiners*, i.e., English constructs such as “the total number of”, “the number of”, “the average of”, etc., that express *aggregate functions*[19], in formal query languages without negatively impacting on the semantic complexity of the controlled fragments.

As we hinted in the introduction, several CLs, most of which are equipped with a compositional semantics[20], have been proposed to provide NLIs to ontologies and ontology-based systems. In particular, to provide English front ends to *(i)* ontology authoring systems, specifically, semantic web ontologies in the form of OWL DL ontologies (for which its fragment ACE-OWL was engineered) and *(ii)* controlled English querying to such ontologies. Table 5 provides an overview of the best known and used, viz., PENG (Schwitter et al., 2003), Rabbit (Schwitter et al., 2008) and OWL CNL (Schwitter and Tilbrook, 2006a), which are to a big extent siblings and/or children of the main two: Attempto Controlled English (ACE) its fragment ACE-OWL (Fuchs et al., 2006; Kaljurand, 2007).

While the coverage of ACE, ACE-OWL and its relatives is way greater than of any of the CLs defined in this chapter, they suffer from our perspective from the fact of being too expressive. Query evaluation over OWL DL (viz., *SHOIN*) ontologies and a fortiori in ACE-OWL NLIs is **coNP**-hard in the size of the data, and hence intractable and unsuited for managing large data repositories. In more expressive CLs such as Rabbit or (full) ACE, reasoning is undecidable.[21] However ACE-OWL (and

[19] That is, second-order functions such as, resp., $sum(\cdot)$, $\#(\cdot)$, $avg(\cdot)$, defined over *sets* of individuals or data values.

[20] Defined by means of an extension of Montague semantics to English discourse, Discourse Representation Theory (DRT), developed by H. Kamp in the 70s and 80s (Kamp, 1988).

[21] Reasoning on OWL Full or FO is undecidable (Baader et al., 2003).

**Table 5** An overview of some CLs; DCG stands for "definite clause grammar" and the other acronyms for known English parsers and/or parser APIs such as GATE.

| CL (English) | Compositional | Maps to | Parser | Goal |
|---|---|---|---|---|
| ACE | yes | FO | APE | Know. rep. |
| ACE-OWL | yes | OWL DL | APE | Ontology man. |
| PENG | yes | OWL DL | ECOLE | Ontology man. |
| OWL CNL | yes | OWL DL | DCG parser | Ontology man. |
| Rabbit | no | OWL Full | GATE | Ontology man. |

kindred CLs) contain (grammatically correct) fragments which may exhibit better computational properties, which we believe can be defined using our methodology.

## 8 Conclusions

In this chapter we have outlined a methodology for defining efficient controlled fragments (CLs) of English for managing ontologies and ontology-based systems that give rise to scalable NLIs to such systems, while expressing key ontology language constructs via a symbolic translation formally underpinned by formal semantics in the Montagovian tradition.

We have argued that this can be achieved as follows: *(i)* On the one hand, by focusing on semantic complexity, viz., the computational complexity of logical reasoning in such CLs, which can be studied via the FO fragment induced by their formal, compositional semantics. We have stressed that a key requirement is for semantic complexity to be at most polynomial in the size of the ontology (or ontology-based system), and in $\mathbf{AC}^0$ in the size of the data stored therein, that is, to have efficient semantic data complexity. *(ii)* On the other hand, by considering English constructs that express ontology languages with efficient data complexity. *(iii)* Finally, by putting together those English constructs via CGs to build a CL that expresses such low complexity ontology languages and that possesses appropriate semantic data complexity while expressing key ontology language constructs.

Following our methodology, we have identified the fragment of English that corresponds to an ontology language suitable for specifying and querying ontologies with optimal data complexity, namely *DL-Lite*; and based on this we have defined an efficient CL, Lite English, using CGs (via the CG-lite grammar).

We have also performed a preliminary corpus analysis regarding the distribution of relevant English constructs. We believe that such method could, if further developed, help the CL community in identifying suitable CLs that provide good trade-offs between coverage and tractability.

# Appendix

In this appendix we sketch how CG-lite formally captures *DL-Lite*$_{R,\sqcap}$ (and a fortiori *DL-Lite*$_{core}$). That is, we show that for every *DL-Lite*$_{R,\sqcap}$ TBox assertion $Cl \sqsubseteq Cr$, there exists a CG-lite derivation $D$ rooted in $s_{tb} \vdash s_{tb} : \forall x(\varphi(Cl,x) \rightarrow \varphi(Cr,x))$.

*Remark 1 (Cl and Cr vs. $\lambda$-FO).* Recall that $Cl$ and $Cr$ concepts are defined as below.

$$Cl \longrightarrow B \mid \exists R \mid Cl_1 \sqcap Cl_2 \quad \text{and} \quad Cr \longrightarrow B \mid \neg B \mid \exists R \mid \neg\exists R \mid \exists R.B.$$

Left concepts correspond to: *(i)* $B$, i.e, $\lambda x.(\mathtt{B}(x))$ (in $\lambda$-FO), *(ii)* $\exists R$, i.e., $\lambda x.\exists y.\mathtt{R}(x,y)$ (in $\lambda$-FO), and *(iii)* $Cl_1 \sqcap Cl_2$, i.e., $\lambda x.(\varphi(Cl,x) \wedge \varphi(Cl,x))$ (in $\lambda$-FO). Regarding right concepts, the new concepts that are not $Cl$s are: *(i')* $\neg B$, i.e., $\lambda x.\neg\mathtt{B}(x)$ (in $\lambda$-FO), *(ii')* $\neg\exists R$, i.e., $\lambda x.\neg\exists y.\mathtt{R}(x,y)$ (in $\lambda$-FO), and *(iii')* $\exists R.B$, i.e., $\lambda x.\exists y.(\mathtt{R}(x,y) \wedge \mathtt{B}(y))$ (in $\lambda$-FO). †

*Remark 2.* In a CG-lite derivation of $\Gamma \vdash A : \alpha$, the resulting category $A$ will match a subcategory $A'$ occurring in a positive position within the categories occurring in $\Gamma$. This means that, when expressing left and right concepts we are interested in derivations where $Cl_1)\ A = n$, $Cl_2)\ A = np\backslash s_{cl}$, $Cl_3)\ A = np\backslash s_{who}$ and $Cr)\ A = np\backslash s_{cr}$. †

**Lemma 1 (Left *Cl* concepts).** For every *DL-Lite*$_{R,\sqcap}$ left concept $Cl$, there exists a CG-lite derivation that expresses it.

*Proof.* (Sketch) We show, by (structural) induction on left concepts $Cl$, that there exists a CG-lite derivation $D$ rooted in either of the following three Lambek sequents: (1) $n \vdash n : \lambda x.\varphi(Cl,x)$ or (2) $np\backslash s_{cl} \vdash np\backslash s_{cl} : \lambda x.\varphi(Cl,x)$ or (3) $\vdash np\backslash s_{who} \vdash np\backslash s_{who} : \lambda x.\varphi(Cl,x)$, with categories found in or derived from CG-lite's lexicon $\mathrm{C{\small AT}}_{lex}$.

- Base cases: $Cl$ is an atomic concept $B$ or a qualified existential $\exists R$.
  1. Consider the lexicon entry $n \vdash n : \lambda x.\mathtt{Student}(x)$; (1) holds.
  2. Consider the lexicon entry $np\backslash s \vdash np\backslash s : \lambda x.\mathtt{Left}(x)$; (2) holds.
  3. Consider the two entries $((np\backslash s_{\exists})/np)\backslash(np\backslash s) \vdash ((np\backslash s_{\exists})/np)\backslash(np\backslash s) : \lambda Z.\lambda y.\exists x.Z(y,x)$ and $(np\backslash s_{\exists})/np \vdash (np\backslash s_{\exists})/np : \lambda x.\lambda z.\mathtt{Reads}(z,x)$. By applying one to each other, we derive $np\backslash s \vdash np\backslash s : \lambda x.\exists y\mathtt{Reads}(x,y)$. Since $s \Rightarrow s_{cl}$, (1) holds.
- Inductive cases: $Cl$ is a complex concept $Cl_1 \sqcap Cl_2$. By I.H. the property holds for $Cl_1$ and $Cl_2$. There are several cases. As they are similar, we deal only with one.
  1. Consider the lexicon entry $((np\backslash s_{cl})\backslash(np\backslash s_{cl}))/(np\backslash s_{cl}) : \lambda X.\lambda Y.\lambda z.(X(z) \wedge Y(z))$, expressing conjunction. By I.H., we may combine it in turn with the (derived) sequents $np\backslash s_{cl} \vdash np\backslash s_{cl} : \lambda x.\varphi(Cl_1,x)$ and $np\backslash s_{cl} \vdash np\backslash s_{cl}$ :

$\lambda x.\varphi(Cl_2,x)$ (i.e., verifying (2)). This results in a derivation rooted in $np\backslash s_{cl} \vdash np\backslash s_{cl} : \lambda x.(\varphi(Cl_1,x) \wedge \varphi(Cl_2,x))$, which satisfies (2). □

**Lemma 2 (Right *Cr* concepts).** For every *DL-Lite*$_{R,\sqcap}$ right concept *Cr*, there exists a CG-lite derivation *D* that expresses it.

*Proof.* (Sketch) The claim can be proven by case analysis on *Cr* as in the preceding lemma (there is no inductive clause in the *Cr* definition), viz., by showing that a derivation *D* rooted in (4) $np\backslash s_{cr} \vdash np\backslash s_{cr} : \lambda x.\varphi(Cr,x)$ exists. □

**Theorem 1.** *For every DL-Lite*$_{R,\sqcap}$ *TBox assertion* $Cl \sqsubseteq Cr$*, there exists a CG-lite derivation D that expresses it.*

*Proof.* The proof follows from the two Lemmas above and by the fact that the only two lexical entries with $s_{tb}$ in a positive position are those for:

1. "every", i.e., $(s_{tb}/(np\backslash s_{cr}))/n_{cl} \vdash (s_{tb}/(np\backslash s_{cr}))/n_{cl} : \lambda X.\lambda Y.\forall x.(X(x) \rightarrow Y(x))$; and, on the other hand,
2. "everyone", i.e., $(s_{tb}/(np\backslash s_{cr}))/(np\backslash s_{cl}) \vdash (s_{tb}/(np\backslash s_{cr}))/(np\backslash s_{cl}) : \lambda X.\lambda Y.\forall x.(X(x) \rightarrow Y(x))$.

Now, by Lemmas 1 and 2, we know that left concepts *Cl* and right concepts *Cr* are CG-lite-expressible, i.e., that there exist derivations for them rooted in $np\backslash s_{cl} \vdash np\backslash s_{cl} : \lambda x.\varphi(Cl,x)$ and $np\backslash s_{cr} \vdash np\backslash s_{cr} : \lambda x.\varphi(Cr,x)$, resp.

When we combine such sequents with the entry for "every", we obtain immediately $s_{tb} \vdash s_{tb} : \forall x(\varphi(Cl,x) \rightarrow \varphi(Cr,x))$. In the case of "everyone", we need to combine them with the entry for "who", viz., $(np\backslash s_{who})/(np\backslash s_{cl}) \vdash np\backslash s_{who})/(np\backslash s_{cl}) : \lambda P.\lambda z.P(z)$, and we again derive $s_{tb} \vdash s_{tb} : \forall x(\varphi(Cl,x) \rightarrow \varphi(Cr,x))$. □

For reasons of space, we omit the proof of the converse, viz., that every (complete) sentence *w* in Lite-English expresses a *DL-Lite*$_{R,\sqcap}$ assertion. It can be constructed in a manner similar to Theorem 1, by induction on CG-lite derivations, i.e., by showing how every CG-lite constituent of category *n* or $np\backslash s_{cr}$ (resp. *n* or $n\backslash s_{cl}$) gives rise to a right (resp. left) concept. Such constituents are then combined together into a sentence expressing an assertion via the function words "every" or by "everyone who". The derivability relations (see Sections 5.2 and 5.3) preventing over-generation.

## References

Aho, A. V., J. D. Ullman, and R. Sethi (1986). *Compilers. Principles, Techniques and Tools*. Addison Wesley - Longman.

Androutsopoulos, I., G. Ritchie, and P. Thanish (1995). Natural langauge interface to databases – an introduction. *Natural Langauge Engineering 1*, 29–81. CUP.

Artale, A., D. Calvanese, R. Kontchakov, and M. Zakharyaschev (2009). The *DL-Lite* family and relations. *J. of Artificial Intelligence Research 36*, 1–69.

Baader, F., D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider (Eds.) (2003). *The Description Logic Handbook: Theory, Implementation and Applications*. CUP.

Barwise, J. and R. Cooper (1980). Generalized quantifiers and natural language. *Linguistics and Philosophy 4*(2), 159–219.

Bernardi, R. (2002). *Reasoning with Polarity in Categorial Type Logic*. Ph. D. thesis, UiL, OTS, Utrecht University.

Bernardi, R., F. Bonin, D. Carbotta, D. Calvanese, and C. Thorne (2007). English querying over ontologies: E-QuOnto. In *Proc. of the 10th Cong. of the Italian Association for Artificial Intelligence (AI*IA 2007)*.

Bernstein, A., E. Kaufmann, C. Kaiser, and C. Kiefer (2006). Ginseng: A Guided Input Natural language Search Engine for querying ontologies. In *2006 Jena User Conference*.

Bos, J. (2008). Wide-coverage semantic analysis with Boxer. In *Proc. of the 2008 Conf. on Semantics in Text Processing (STEP 2008)*.

Calvanese, D., G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati (2007). Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning 39*(3), 385–429.

Calvanese, D., G. De Giacomo, M. Lenzerini, D. Lembo, A. Poggi, and R. Rosati (2007). MASTRO-I: Efficient integration of relational data through DL ontologies. In *Proc. of the 20th Int. Workshop on Description Logics (DL 2007)*.

Damljanovic, D. (2010). Towards portable controlled natural languages for querying ontologies. In *Proc. of the 2nd Workshop on Controlled Natural Languages (CNL 2010)*.

Dongilli, P. and E. Franconi (2006). An intelligent query interface with natural language support. In *Proc. of the 19th Int. Florida Artificial Intelligence Research Society Conf. (FLAIRS 2006)*, Melbourne Beach, Florida, USA.

Eyck, J. v. (1985). *Aspects of Quantification in Natural Language*. Ph. D. thesis, University of Groningen.

Franconi, E., P. Guagliardo, and M. Trevisan (2010). Quelo: NL-based intelligent query interface. In *Proc. of the 2nd Workshop on Controlled Natural Languages (CNL 2010)*.

Fuchs, N. E., K. Kaljurand, and G. Schneider (2006). Attempto Controlled English meets the challenges of knowledge representation, reasoning, interoperability and user interfaces. In *Proc. of the 19th Int. Florida Artificial Intelligence Research Society Conf. (FLAIRS 2006)*.

Grädel, E. (1999a). Decision procedures for guarded logics. In *Proc. of the 16th Int. Conf. on Automated Deduction (CADE 1999)*, Volume 1632 of *Lecture Notes in Computer Science*, pp. 31–51. Springer.

Grädel, E. (1999b). Why are modal logics so robustly decidable? *Bulletin of the European Association for Theoretical Computer Science 68*, 90–103.

Gunning, D., V. K. Chaudhri, P. Clark, K. Barker, S. Chaw, M. Greaves, B. Grosof, A. Leung, D. McDonald, S. Mishra, J. Pacheco, B. Porter, A. Spaulding, D. Tecuci, and J. Tien (2010). Project Halo update – progress toward digital Aristotle. *AI Magazine 31*(3), 33–58.

Horrocks, I., P. F. Patel-Schneider, and F. van Harmelen (2003). From *SHIQ* and RDF to OWL: The making of a Web Ontology Language. *J. of Web Semantics 1*(1), 7–26.

Hujsen, W. O. (1998). Controlled language – An introduction. In *Proc. of the 2nd Int. Workshop on Controlled Language Applications (CLAW 1998)*.

Hustadt, U., R. Schimdt, and L. Georgieva (2004). A survey of decidable first-order fragments and description logics. *J. on Relational Methods in Computer Science 1*(1).

Kaljurand, K. (2007). *Attempto Controlled English as a Semantic Web Language*. Ph. D. thesis, University of Tartu. Available at `http://attempto.ifi.uzh.ch/site/pubs/`.

Kaljurand, K. and N. E. Fuchs (2006). Birectional mapping between OWL-DL and Attempto Controlled English. In *Proc. of the 4th Int. Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR 2006)*.

Kamp, H. (1988). Discourse Representation Theory: What it is and where it ought to go. In A. Bläser (Ed.), *Natural Language and the Computer*, pp. 84–111. Springer.

Keenan, E. and L. Faltz (1985). *Boolean Semantics for Natural Language*. Dordrecht: Reidel.

Kittredge, R. I. (2003). Sublanguages and controlled languages. In R. Mitkov (Ed.), *The Oxford Handbook of Computational Linguistics*, pp. 430–447. Oxford University Press.

Kurtonina, N. and M. Moortgat (1995). Structural control. In P. Blackburn and M. de Rijke (Eds.), *Logic, Structures and Syntax*. Dordrecht: Kluwer.

Lambek, J. (1958). The mathematics of sentence structure. *American Mathematical Monthly 65*, 154–170.

Meng, F. and W. Chu (1999). Database query formation from natural language using semantic modeling and statistical keyword meaning disambiguation. Technical report, University of California at Los Angeles. Available at `http://www.cobase.cs.ucla.edu/publications.html`.

Moortgat, M. (1997). Categorial Type Logics. In J. van Benthem and A. ter Meulen (Eds.), *Handbook of Logic and Language*, pp. 93–178. Cambridge: MIT.

Moot, R. (1998). Grail: An automated proof assistant for categorial grammar logics. In *Proc. of the 1998 User Interfaces for Theorem Provers Conf.*

Pratt, I. and A. Third (2006). More fragments of language. *Notre Dame J. of Formal Logic 47*(2), 151–177.

Schwitter, R., K. Kaljurand, A. Cregan, C. Dolbear, and G. Hart (2008). A comparison of three controlled natural languages for OWL 1.1. In *Proc. of the 4th Int. Workshop on OWL: Experiences and Directions (OWLED 2008)*.

Schwitter, R., A. Ljungberg, and D. Hood (2003). ECOLE - a look-ahead editor for a controlled language. In *Proc. of the 8th Int. Workshop of the European Association for Machine Translation and the 4th Controlled Language Applications Workshop (EAMT/CLAW 2003)*.

Schwitter, R. and M. Tilbrook (2006a). Let's talk in description logic via controlled language. In *Proc. of the 3rd Int. Work. on Logic and Engineering of Natural Language Semantics (LENLS 2006)*.

Schwitter, R. and M. Tilbrook (2006b). Let's talk in description logic via controlled natural language. In *Proc. of the 3rd Int. Workshop on Logic and Engineering of Natural Language Semantics (LENLS 2006)*.

Sowa, J. (1999). *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks Cole Publishing Co.

Sowa, J. F. (2004, 24 February). Common logic controlled english. Draft.

Thorne, C. (2010). *Query Answering over Ontologies Using Controlled Natural Languages*. Ph. D. thesis, Free University of Bozen-Bolzano, Faculty of Computer Science.

Thorne, C. (2011). Querying the fragments of english. In *Proc. of the 18th Int. Workshop on Logic, Language and Computation (WoLLIC 2011)*.

Thorne, C. and D. Calvanese (2009a). Controlled English ontology-based data access. In *Proc. of the 1st Workshop on Controlled Natural Languages (CNL 2009)*.

Thorne, C. and D. Calvanese (2009b). Tree-shaped aggregate queries over ontologies. In *Proc. of the 8th Int. Conf. on Flexible Query Answering Systems (FQAS 2009)*.

Thorne, C. and D. Calvanese (2012). Tractability and intractability of controlled languages for data access. *Studia Logica 100*, 787–813.

van Benthem, J. (1987). Categorial grammar and lambda calculus. In D. Skordev (Ed.), *Mathematical Logic and its Applications*, pp. 39–60. Plenum, New York.

Vardi, M. (1982). The complexity of relational query languages. In *Proc. of the 14th Annual ACM Symp. on Theory of Computing*.

# Index