Lite Natural Language

Raffaella Bernardi, Diego Calvanese, Camilo Thorne
Free University of Bozen-Bolzano, Italy
lastname@inf.unibz.it

1 Introduction

Many have been the attempts in the Seventies and Eighties to build Natural Language Interfaces (NLI) to Databases which have turned out to some level of disappointment towards the Nineties [1]. Nowadays, a similar question is calling for attention again due to the new development in Knowledge Representation and Language Technologies and the widespread interest in Ontology and the Semantic Web [5, 22]. The emphasis is on how to build NLI where only a suitable fragment of natural language (a *controlled natural language*) is used [14, 16, 25]. Systems have been proposed that guide the user to formulate his/her question via an ontology that incrementally shows the possible concepts on which the remaining part of the question could be about [11, 10]. Others guide the user via an incremental parser [6, 23]. Both approaches aim to allow the user to build only those questions that the system can handle.

The present paper addresses the question of *which* should be the natural language fragment to be used for such a purpose, and *how* we can define it. To this end we consider of particular value the studies carried out by Ian Pratt [21] who is investigating the satisfiability of sets of sentences in fragments of natural language and its computational complexity. Our proposal is to merge Pratt's approach with the research mentioned above and use as controlled language for accessing ontologies those fragments with a desirable computational complexity. We use a Description Logic (DL) [2] as the starting point to answer the *which* part of our question, viz. to pinpoint the most suitable fragment, and Categorial Grammar (CG) as the system to answer the *how*, viz. aiming to capture exactly those syntactic structures corresponding to all and only the meaning representations allowed in the chosen DL.

Our work is quite close to the research presented in [24] in that our Controlled Natural Language expressions have meaning representations that can be expressed in a DL. The difference lies on the one hand in the kind of DL we have considered, and on the other hand in the Grammar. W.r.t. the kind of DL, we focus our attention on DL-Lite, which is a DL (more precisely, a family of DLs) studied in the context of ontology-based access to (relational) databases [7, 8]. As opposed to OWL DL, which is the DL considered in [24], DL-Lite is specifically optimised w.r.t. the size of the data (rather than the size of the intensional descriptions in the ontology), when considering the well-known tradeoff between expressive power and computational complexity of inference. Indeed, it is, in a precise technical sense [9], the maximal DL that has the ability to efficiently and effectively manage very large data repositories by relying on industrial-strength relational database management systems (RDBMS). Hence, our work is particularly relevant in all those contexts where NLI to data-intensive systems need to be provided, and where the expressive power granted by richer DLs (e.g., OWL DL) does not provide a sufficient guarantee for effectiveness. As for the formal grammar, we have used a logical grammar whose categories are recursively defined and are mapped to typed lambda terms. We believe this logical approach to parsing could help addressing the issue of defining the fragment of natural language suitable for the tasks we are interested in.

In Section 2 we define the question we address and provide the motivations for our choices. In Section 3 and Section 4 we introduce the DL and the Grammar we work with. In Section 5 we describe in details how CG can capture exactly the fragment of Natural Language we are looking for.

2 The framework

Description Logics (DLs) are the logics, typically fragments of First Order Logic (FOL), that provide the formal underpinning to ontologies and the Semantic Web [13]. In this setting, the ontologies we deal with consist of an intensional component (called TBox, for "Terminological Box"), and an extensional one (called ABox, for "Assertional Box"), viz. the former consists of a set of universal statements and the latter of a set of atomic facts, the data.

Our aim is to build a natural language interface that helps users to specify and query an ontology expressed in a DL. In other words, to (i) enter facts in the ABox and universal statements in the TBox (*specify* the ontology), as well as (ii) extract information from them (*query* the ontology). To this end, we follow the Controlled Language approach to the problem. Since we are interested in performing these tasks over large data, efficiency in managing such data through the ontology is of primary importance. For this reason, we focus our attention on DL-Lite [7], a family of DLs specifically tailored to manage large amounts of data efficiently. More precisely, answering unions of conjunctive queries[1] over a DL-Lite ontology is polynomial in the size of the TBox, and LogSpace in the size of the ABox, and, most importantly, it can be done by relying on a commercial RDBMS system for storing the ABox and managing the access to its data. Moreover, the DLs in the DL-Lite family are essentially the maximal DLs that exhibit such nice computational properties [9].

We want to capture the fragment of English that consists of all and only those sentences whose meaning representation belongs to DL-Lite. This fragment is rather restricted and might turn out to be too constrained for a user. Therefore, we plan to study how natural language structures that outscope the defined fragments can be rephrased so to maintain their meaning at least partially while satisfying the constraints imposed by the defined grammar, and therefore be reduced to eligible constructs.

The idea of studying a Controlled Language, though brought up for different motivations, is not too far away from Montague's proposal to restrict attention to fragments of natural languages [26]. His program was based on the thesis that the relation be-

[1]Such queries correspond to unions of select-project-join SQL queries, which constitute the vast majority of queries posed to RDBMS systems.

tween syntax and semantics in a natural language such as English could be viewed as not essentially different from the relation between syntax and semantics in a formal language such as the language of FOL. The main components of his framework are: (i) the principle of compositionality (i.e., the meaning of the whole is built out of the meaning of its parts) together with the idea that the construction of meaning is guided by the syntactic structure; (ii) the view of words (and phrases) as complete (e.g., noun phrases) and incomplete (e.g., verbs) expressions, and consequently their representation as functions by means of lambda terms and the assignment of categorial grammar types as syntactic categories; and finally (iii) the model-theoretic interpretation of the obtained meaning representations based on standard FOL interpretation. Following Montague, we use a Categorial Grammar (CG) to capture the needed natural language fragment. Furthermore, we exploit the syntax-semantics interface provided by the Curry-Howard Correspondence between the Lambek Calculus (the logic version of CG) and the lambda calculus to obtain DL-Lite meaning representation compositionally while parsing [3, 19].

3 DL-Lite and Natural Language

3.1 Introduction to DL-Lite

DLs [2] are logics that 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 concepts 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, storing intensional information, and an ABox, storing assertional information about individual objects of the domain of interest.

In this work, we consider DLs belonging to the DL-Lite family [7], and 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 DL-Lite$_{core}$ and DL-Lite$_{\mathcal{R},\sqcap}$, and defined below. In fact, DL-Lite$_{core}$ represents a core part shared by all logics of the DL-Lite family.

Definition 3.1 (DL-Lite$_{core}$ and DL-Lite$_{\mathcal{R},\sqcap}$) *In DL-Lite$_{core}$, Cl and Cr are defined as follows:*[2]

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

where A denotes an atomic concept, and R denotes an atomic role.

In DL-Lite$_{\mathcal{R},\sqcap}$, in addition to the clauses of DL-Lite$_{core}$, we have also:

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

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

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

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

$$\begin{array}{ll} \varphi(A,x) = A(x) & \varphi(R,x,y) = R(x,y) \\ \varphi(\neg C,x) = \neg\varphi(C,x) & \varphi(\exists R,x) = \exists y.\, \varphi(R,x,y) \\ \varphi(C_1 \sqcap C_2,x) = \varphi(C_1,x) \wedge \varphi(C_2,x) & \varphi(\exists R.C,x) = \exists y.\, \varphi(R,x,y) \wedge \varphi(C,y) \end{array}$$

An inclusion assertion $Cl \sqsubseteq Cr$ of the TBox corresponds then to the universally quantified FOL sentence $\forall x.\, \varphi(Cl, x) \rightarrow \varphi(Cr, x)$.

Finally, in DL-Lite, an ABox is constituted by a set of assertions on *individuals*, of the form $A(c)$ or $R(a, b)$, where A and R denote respectively an atomic concept and role, and a, and b denote constants. As in FOL, each constant is interpreted as an element of the interpretation domain. The above ABox assertions correspond to the analogous FOL facts, or, by resorting to the above mapping, to $\varphi(A, x)(c)$ and $\varphi(R, x, y)(a, b)$, respectively.

We are interested in studying the linguistic structures that correspond to such constructs. In the following we look at their straightforward translations into natural language.

3.2 Fragment of Natural Language for DL-Lite

The constraints expressed in the TBox are universals. They are of the form $Cl \sqsubseteq Cr$ that translates into FOL as $\forall x.Cl(x) \rightarrow Cr(x)$ and in natural language as

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

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

Hence, the determiner "every" and the quantifier "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 the Cl and Cr of DL-Lite can be expressed in English.

First of all, notice that since an atomic concept A is a unary predicate, it can be either a noun "student", e.g., (2) or an intransitive verb phrase "left"(1); and its negation, $\neg A$, can be either "is not a boy"(3), or "does not leave"(4).

The introduction of the $\exists R$ in the Cl part can be performed by means of the quantifier "everyone" followed by the relative pronoun "who"(5 and 6) (or by the conjunction that would correspond to the use of $\sqcap$ on the Cl part allowed in DL-Lite$_{\mathcal{R},\sqcap}$ fragment, see (13) below).

(1) Every student left $[Student \sqsubseteq Left]$

(2) Every student is a boy $[Student \sqsubseteq Boy]$

(3) Every student is not a boy $[Student \sqsubseteq \neg Boy]$

(4) Every student does not leave $[Student \sqsubseteq \neg Leave]$

(5) Everyone who eats left $[\exists Eats \sqsubseteq Left]$

(6) Everyone who knows something left $[\exists Know \sqsubseteq Left]$

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, "something" (7), and its negation to the use of "does not" to negate such construction (8).

(7) Every student knows something $[Student \sqsubseteq \exists Know]$

(8) Every student does not know something $[Student \sqsubseteq \neg\exists Know]$

Note, that as the DL-Lite clause show the only reading of the ambiguous sentence in (8) is the one with *every* having wide scope and *something* be in the scope of *not*[3].

When we move to DL-Lite$_{\mathcal{R},\sqcap}$, the addition of the conjunction in the Cl part corresponds to the use of adjective (9), or relative clauses modifying the noun quantified by "every" (10-12), or the "and" coordinating two verb phrases (13).

(9) Every nice student left.
$\forall x.(\texttt{student}(x) \wedge \texttt{nice}(x)) \rightarrow \texttt{left}(x)$ $[Student \sqcap Nice \sqsubseteq Left]$

(10) Every student who studies left.
$\forall x.(\texttt{student}(x) \wedge \texttt{study}(x)) \rightarrow \texttt{left}(x)$ $[Student \sqcap \exists Study \sqsubseteq Left]$

(11) Every student who is a boy left.
$\forall x.(\texttt{student}(x) \wedge \texttt{Boy}(x)) \rightarrow \texttt{left}(x)$ $[Student \sqcap Boy \sqsubseteq Left]$

(12) Every student who eats something left.
$\forall x.(\texttt{student}(x) \wedge \exists y.\texttt{eats}(x,y)) \rightarrow \texttt{left}(x)$ $[Student \sqcap \exists Eats \sqsubseteq Left]$

(13) Everyone who drinks something and eats something left.
$\forall x.(\exists y.\texttt{drink}(x,y) \wedge \exists z.\texttt{eats}(x,z)) \rightarrow left(x)$ $[\exists Drinks \sqcap \exists Eats \sqsubseteq Left]$

Furthermore, the introduction of the qualified existential on the Cr is performed by the determiner "a" (14).

(14) Every student knows a girl.
$\forall x.\texttt{student}(x) \rightarrow \exists y.\texttt{girl}(y) \wedge \texttt{know}(x,y)$ $[Student \sqsubseteq \exists Know.Girl]$

[3]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 [4].

An important remark to emphasize is the presence of the relative pronoun in the above fragment of sentences. Pratt [21] has shown how the uncontrolled use of such expression brings to `NP-complete` fragments when allowing the use only of the copula or even `EXPTIME-completeness` when adding transitive verbs. Below, we will show how relative pronouns can be used in a controlled grammar and preserve tractability of inferences.

We now turn to show how the Lambek Calculus can be used to capture the Natural Language fragment consisting *only* of linguistic structures listed above or a recursive extension of them. Notice that the latter can happen only by means of either conjunction (*and*) or an adjective.

4 CG and Natural Language for DL-Lite

4.1 Introduction to CG

As most of the linguistically motivated formal grammars currently in use, Categorial Grammar (CG) is a lexicalised grammar, i.e., the lexicon carries most of the information about how words can be assembled to form grammatical structures. The peculiarity of CG is that it captures the tight correspondence between syntax and semantics of natural language. In the logical version we employ, namely the (non associative) Lambek Calculus (NL) [18, 19], this aspect is even stronger thanks to the Curry-Howard Correspondence that holds between the logical rules of NL and (a fragment of) typed lambda calculus [3]. In this framework, syntactic categories are seen as *formulas* and their category forming operators as connectives, i.e., *logical constants*. As a result, the rules for category combination are the (very few) rules of inference for these connectives (function application and abstraction[4]). 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 Lambek Calculus.[5]

Information both about the syntactic structure where the word could occur and its meaning are stored in the lexicon.

Definition 4.1 (Term Labelled Lexicon) *Given a set Σ of basic expressions of a natural language, a term labelled categorial lexicon is a relation,*

$$\mathsf{LEX} \subseteq \Sigma \times (\mathsf{CAT} \times \mathsf{TERM}) \textit{ such that if } (w, (A, \alpha)) \in \mathsf{LEX}, \textit{ then } \alpha \in \mathsf{TERM}_{\mathtt{type}(A)}$$

where TERM is the set of all lambda terms and $\mathsf{TERM}_{\mathtt{type}(A)}$ denotes the set of lambda terms whose type is mapped to the category A. CAT is defined as follows

$$\mathsf{CAT} ::= \mathsf{ATOM} \mid \mathsf{CAT}\backslash\mathsf{CAT} \mid \mathsf{CAT}/\mathsf{CAT}$$

In the following, we will use the set of atoms $\mathsf{ATOM} = \{np, n, s\}$, and the function $\mathtt{type}$*: $\mathsf{CAT} \rightarrow \mathsf{TYPE}$ mapping syntactic categories to semantic types given below, where the atomic types are e (entity) and t (truth values), and (a, b) denotes the functional type $a \rightarrow b$ as always.*

[4]In this paper we use the product free version of NL.

[5]The lexicon we present in this article has been tested using `Grail` [20].

$$\begin{array}{ll} \mathtt{type}(np) = e; & \mathtt{type}(A/B) = (\mathtt{type}(B), \mathtt{type}(A)); \\ \mathtt{type}(s) = t; & \mathtt{type}(B\backslash A) = (\mathtt{type}(B), \mathtt{type}(A)); \\ \mathtt{type}(n) = (e,t). & \end{array}$$

This constraint on lexical entries enforces the requirement that if the expression w is assigned a syntactic category A and term α, then the term α is of the appropriate type for the category A. We will assign lambda terms whose body is a FOL formula, λ-FOL.

We look at the determiner *every*, by means of example, since it has a crucial role in our grammar. The reader is referred to [15, 12] for an in depth explanation of this and similar expression.

Example 4.2 (Determiner) *The meaning of "every* `NOUN`*" (e.g., "every man") is the set of those properties that every NOUN (e.g., man) has*

$$[\![\mathtt{every\ NOUN}]\!] = \{X | [\![\mathtt{NOUN}]\!] \subseteq X\}.$$

Therefore, in a functional perspective, it is seen as a two argument function that corresponds to the following syntactic category

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

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]* `VP`*]). The typed lambda term corresponding to this syntactic category is:* $\lambda X_{(e,t)}.\lambda Y_{(e,t)}.\forall x_e X(x) \to Y(x)$ *that properly represents the set theoretical meaning given above. In the following, we won't use types on the lambda term unless necessary.*

Our proposal for the definition of the proper fragment of natural language exploits this correspondence between syntactic categories and lambda terms.

Furthermore, it takes advantage of derivability relations among categories of the same semantic type carried out by unary operators decorating CAT [4]. For reason of space, we won't go into the details of this part which would require the introduction of the multi modal extension of NL. It suffices to provide the intuitive idea behind the proposed solution: a function $A \to B$ can be applied to either an argument A or to an argument C such that C derives A ($C \Rightarrow A$). In our case, $\Rightarrow$ is the derivability relation of the logical grammar we use.

Finally, the "parsing as deduction" approach, gives us a mean to reduce the problem of identifying a proper set of linguistic structures to a problem of defining the allowed logical formulas. In other words, instead of looking at linguistic strings $w_1 \dots w_n$, we can exploit the formally well defined structures corresponding to them. Parsing a string $w_1 \dots w_n$ amounts to prove that $\Gamma \vdash B : \phi$, where Γ consists of pairs of categories and terms as defined in the lexicon (viz. $(w_i, (A_i, \alpha_i))$), $A_1 : \alpha_1, \dots A_n : \alpha_n$, to be proved to be of category B. As by-product of this derivation one derives also the meaning representation of the structure assigned to the string, i.e., the lambda term ϕ. For instance, parsing *Every nice student left* means to prove that the following is a theorem of NL:

$$(((s/(np\backslash s))/n : \mathtt{every} \circ (n/n : \mathtt{nice} \circ n : \mathtt{student})) \circ np\backslash s : \mathtt{left}) \vdash s : \phi$$

by using the proper lambda terms, through the proof of the above entailment, ϕ results to be $\forall x.(\texttt{nice}(x) \wedge \texttt{student}(x)) \rightarrow \texttt{left}(x)$.

5 CG-lite

In the present paper, our task is to find syntactic categories that lexically control the restrictions imposed by the DL-Lite constructs. We proceed step by step, by first looking at the requests of DL-Lite$_{core}$ and consider the two constraints regarding the use of negation:

1. negation of atomic concepts can occur in the Cr but not in the Cl part: $Cl \longrightarrow A,\ Cr \longrightarrow A \mid \neg A$

2. an unqualified existential can occur both in Cl and Cr, but its negation can occur only in Cr: $Cl \longrightarrow \exists R, Cr \longrightarrow \exists R \mid \neg\exists R$

The Cl and Cr parts correspond to the restrictive scope (the `NOUN`), and nuclear scope (the `VP`) of *every*, respectively. We need to constrain the linguistic structures that occur in them. In particular, we need to block the occurrences of the negation in Cl and express the fact that `NOT` cannot have its scope on any `VP` occurring in the restrictive scope of *every*. As emphasised in [4], in CG scope is determined by the sentential categories s of the complex formulas, and different scope distribution can be accounted for by differentiating the sentential levels of the scope constructors at work, and exploiting the derivability relations among categories.

We mark the structures that can occur in the two parts of the DL-Lite clauses and the negative and positive ones, by means of the four sentential levels s_{cl}, s_{cr}, $s_{\neg}$, and s, respectively, and establish the derivability relation below.[6] They 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 ($s \Rightarrow s_{cl}, s \Rightarrow s_{cr}$).

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

These constraints are lexically anchored by means of the lexical assignments below.

Example 5.1 (Lexicon for DL-Lite$_{core}$) The lexicon entries to use are as below[7]

- Every $\in (s_t/(np\backslash s_{cr}))/n$: $\lambda X.\lambda Y.\forall x.X(x) \rightarrow Y(x)$
- 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 $\in (np\backslash s_{\neg})/(np\backslash s)$: $\lambda X.\lambda z.\neg X(z)$
- left $\in np\backslash s$: $\lambda z.\texttt{left}(z)$

[6] We actually use residuated unary operators to carry out these derivability relations [17] 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".

[7] 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.

- studies $\in np\backslash s$: $\lambda z.\exists x.\texttt{study}(z,x)$
- student $\in n$: $\lambda z.\texttt{student}(z)$
- everyone: $(s_t/(np\backslash s_{cr}))/(np\backslash s_{who})$: $\lambda X.\lambda Y.\forall x.X(x) \rightarrow Y(x)$
- who: $(np\backslash s_{who})/(np\backslash s_{cl})$, $\lambda P.\lambda z.P(z)$
- something: $((np\backslash s_{\exists})/np)\backslash(np\backslash s)$, $\lambda Z.\lambda y.\exists x.Z(y,x)$
- studies: $(np\backslash s_{\exists})/np$, $\lambda x.\lambda z.\texttt{studies}(z,x)$

First of all, notice that the categories assigned to *every* and *everyone* rule out the possibility for them to occur in object position –they can only be in a subject position. Moreover, since they are the only entries yielding a TBox sentence (s_t), 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*.

Finally, recall, that since, in this fragment we do not have the $\sqcap$ on the Cl part, 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 part correspond to the use of a transitive verb followed by an existential quantifier, *something*. The lexical entries for *everyone*, *who*, *something* and *studies* 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 request.[8] Finally, the category assigned to "something" is such that it can occur only in object position.

The described fragment recognises as grammatical all the structures in (1)-(8) above.

The reader can gain a better understanding of the mechanisms involved by checking how the lexicon predicates the ungrammaticality of the sentences below whose meaning representations are not in DL-Lite

(15) Everyone who does not know something left $[\neg\exists Know \sqsubseteq left]$

(16) Everyone who is not a boy left. $[\neg Boy \sqsubseteq left]$

We now move to DL-Lite$_{\mathcal{R},\sqcap}$, and account for the following additions

1. the conjunction can occur in the Cl part, $Cl \longrightarrow Cl_1 \sqcap Cl_2$
2. the qualified existential can occur in the Cr part, $Cr \longrightarrow \exists R.A$

Example 5.2 (Lexicon extension for DL-Lite$_{\mathcal{R},\sqcap}$) In order to move to DL-Lite$_{\mathcal{R},\sqcap}$, we need to add into the lexicon the following lexical entries.

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

[8] 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 (9)-(14). Notice, the need of having the conjunction operating at the sentential level s_{cl}: this blocks the composition of negation (*does not*) with a verb phrase built by *and*, that would wrongly give: *does not walk and speak* with *not* having wide scope over *and*, viz. $\lambda z.\neg(\texttt{walk}(z) \wedge \texttt{speak}(z))$ that is not part of DL-Lite. For similar reasons, we have to block the composition of *is not a* with noun phrases built by means of an adjective. Again this composition would result into terms with the negation having scope over the conjunction, e.g., *is not a nice student* with term: $\lambda z.\neg(\texttt{nice}(z) \wedge \texttt{student}(z))$. The introduction of the category n_{cl} with $n \Rightarrow n_{cl}$ helps blocking the construction of these terms. Furthermore, we have considered the version of DL-Lite with qualified existential of the form $\exists R.A$, rather than $\exists R.C$, 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 are the ones that bring recursion into the language.

The fragment of sentences whose meaning representation belongs to a DL-Lite ABox is rather easy to built since an ABox consists only of unary or binary predicates whose arguments are constants. In other words, the lexicon is built only with noun, intransitive verbs, the copula (i.e., unary predicates), transitive verbs (i.e., binary predicates), and personal nouns. Since we can see any subset of ABox assertion as conjunction of such clauses, we could have in our lexicon also adjectives and relative pronoun.

6 Conclusions and Further Work

In this paper we have presented a first step towards the definition of the fragment of English that corresponds to a fragment of FOL suitable for specifying and querying ontologies, DL-Lite. The obtained results shed lights on the possibility of further exploiting the logical nature of the Grammar we employed to capture *all and only* sentences in the given logic space. In particular, we will investigate the possibility of defining categories in the CG framework so to guarantee that their corresponding meaning representation won't outscope DL-Lite when composed by means of the CG inference rules. Furthermore, we plan to investigate how natural language structures could be re-written into the defined grammar. Finally, we will extend our analysis of TBox and ABox clauses to queries too.

References

[1] I. Androutsopoulos, G.D. Ritchie, and P. Thanish. Natural langauge interface to databases – an introduction. *Natural Langauge Engineering*, 1:29–81, 1995. Cambridge University Press.

[2] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[3] J. van Benthem. Categorial grammar and lambda calculus. In D. Skordev, editor, *Mathematical Logic and its Applications*, pages 39–60. Plenum, New York, 1987.

[4] R. Bernardi. *Reasoning with Polarity in Categorial Type Logic*. PhD thesis, UiL, OTS, Utrecht University, 2002.

[5] A. Bernstein, E. Kaufmann, A. Göhring, and C. Kiefer. Querying ontologies: A controlled english interface for end-users. In *4th International Semantic Web Conference (ISWC 2005)*, pages 112–126, November 2005.

[6] A. Bernstein, E. Kaufmann, C. Kaiser, and C. Kiefer. Ginseng: A guided input natural language search engine for querying ontologies. In *2006 Jena User Conference*, May 2006.

[7] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. DL-Lite: Tractable description logics for ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.

[8] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tailoring OWL for data intensive ontologies. In *Proc. of the Workshop on OWL: Experiences and Directions (OWLED 2005)*, 2005.

[9] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 260–270, 2006.

[10] P. Dongilli and E. Franconi. An intelligent query interface with natural language support. In *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2006)*, Melbourne Beach, Florida, USA, May 2006.

[11] P. Dongilli, E. Franconi, and S. Tessaris. Semantics driven support for query formulation. In *Proceedings of the 2004 International Workshop on Description Logics (DL 2004)*, volume 104, Whistler, BC, Canada, June 2004.

[12] J. van Eijck. *Aspects of Quantification in Natural Language*. PhD thesis, University of Groningen, 1985.

[13] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.

[14] W. O. Hujsen. Controlled language – an introduction. In *Proceedings of CLAW 1998*, pages 1–15, Pittsburg, 1998.

[15] E. Keenan and L. Faltz. *Boolean Semantics for Natural Language.* Reidel, Dordrecht, 1985.

[16] R. I. Kittredge. Sublanguages and controlled languages. In R. Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, pages 430–447. Oxford University Press, 2003.

[17] N. Kurtonina and M. Moortgat. Structural control. In P. Blackburn and M. de Rijke, editors, *Logic, Structures and Syntax.* Dordrecht: Kluwer, 1995.

[18] J. Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.

[19] M. Moortgat. Categorial Type Logics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 93–178. The MIT Press, Cambridge, Massachusetts, Cambridge, 1997.

[20] R. Moot. Grail: An automated proof assistant for categorial grammar logics. In R. Backhouse, editor, *Proceedings of the 1998 User Interfaces for Theorem Provers Conference*, pages 120–129, 1998.

[21] I. Pratt-Hartmann and A. Third. More fragments of language: the case of ditransitive verbs. *Notre Dame Journal of Formal Logic*, 2006.

[22] R. Schwitter and M. Tilbrook. Controlled natural language meets the semantic web. In S. Wan A. Asudeh, C. Paris, editor, *Proceedings of the Australasian Language Technology Workshop 2004*, pages 55–62. Macquarie University, 2004.

[23] R. Schwitter and M. Tilbrook. Dynamic semantics at work. In *Proceedings of the International Workshop on Logic and Engineering of Natural Language Semantics*, pages 49–60, May 2004.

[24] R. Schwitter and M. Tilbrook. Let's talk in description logic via controlled natural language. In *Proceedings of the Third International Workshop on Logic and Engineering of Natural Language Semantics (LENLS2006)*, pages 193–207, 2006.

[25] J. F. Sowa. Common logic controlle english. Draft, 24 February 2004.

[26] R. Thomason, editor. *Formal Philosophy: Selected papers of Richard Montague.* Yale University Press, New Haven, 1974.