Logical Form and Knowledge Representation: towards a reconciliation

Enrico Franconi
Knowledge Representation and Reasoning group
IRST, I-38050 Povo TN, Italy
franconi@irst.itc.it

Abstract

This paper arguments – unlike (Allen 1993) – in favor of a *logical form* as a basis for semantic interpretation in natural language processing. To this purpose, a knowledge representation formalism that fully parallels the logical form is introduced – allowing for both underspecified semantic representations and encapsulation of contextual knowledge in the form of meaning postulates. In this way, the gap between researchers in computational linguistics – interested in expressiveness and in computing explicit derivations – and researchers in knowledge representation – interested in devising rigorous logic-based representations with robust and complete computational properties – can be filled. This paper tries to give an answer to most of the problems presented by (Allen 1993). As a matter of facts, the approach advocated here is exactly along the lines indicated by Allen. However, Allen's conclusions were in the direction of forgetting about logical forms, and of representing the literal and contextual meanings of utterances in a suitable NL-oriented knowledge representation formalism. I will show how such a formalism can be directly designed starting from logical forms. This paper does not examine the technical aspects of the devised knowledge representation – the reader should refer to the cited bibliography. The scope is only to demonstrate how it is possible to justify such a knowledge representation.

Introduction

Traditionally, the logical form has been considered in computational linguistics as representing the literal – i.e. context independent – meaning of an utterance, as clearly distinguished from the representation of the surface syntactical constituent structure, and from a deeper semantic representation – function of discourse context and world knowledge. Thus, the logical form plays an intermediate role between syntax and the deep semantics, and it is therefore not intended to fully contain the meaning in context of the utterance. Moreover, quite often a further distinction is introduced among *quasi* logical forms – i.e. literal underspecified semantic representations – and proper logical forms – i.e. literal unambiguous derivations.

The reasons for separating the literal underspecified, the literal unambiguous, and the deep meaning representations are mainly pragmatic rather than theoretical. Pure linguists would say that any sentence has just one meaning which is unambiguous, being the possible ambiguity introduced by underconstraining the processing – e.g. by not adequately considering the context. In such a case, they would speak of different possible ending paths in the derivation process, each one of them being again unambiguous. Clearly, this approach is unfeasable from a computational point of view: first, because the number of derivations might combinatorially increase; and secondly because the interdependencies among the derivations are lost.

On the other hand, computational linguists consider ambiguities as part of the meaning of utterances, with the ultimate goal of being able to reason with such underspecified expressions, in order to increase compactness in the representation and efficiency in the processing. (Allen 1993) argues that

> ... one of the crucial issues facing future natural language systems is the development of knowledge representation formalisms that can effectively handle ambiguity.

We can identify two main approaches. The classical *computational* approaches rely on the modularity of the semantic analysis process, starting from the underspecified representation and ending up with an unambiguous and context-dependent representation. The *semantic-oriented* approaches usually propose a very expressive logical language – possibly with an expressivity greater than FOL – with the goal of giving a clear semantics to many NL phenomena.

A main disadvantage of the first approach is that there are no solid formal grounds for the treatment of ambiguity, and that the process of disambiguation is often based on heuristics and ad-hoc procedures. This can be justified by the fact that reasoning on logical forms including – among other things – domain knowledge, incomplete and ambiguous terms, unsolved references, underspecified quantifications, is considered a hard computational task. Computational linguists have devised structural processing techniques based on syntax, selectional restrictions, case grammars, and structured information such as frames – carefully trying to avoid or to drastically reduce the inclusion in the computational machinery of logical inferencing mechanisms for treating ambiguities. Of course, these techniques often have difficulties to work when such ambiguities come into play.

In the second approach the computational aspects have usually little consideration – if considered at all – since the main accent is on the capability of the logic to express particular linguistic phenomena. However, we believe that the study of the computational properties of the logic under consideration is important. This requires decidability of the logic and completeness of the algorithms with respect to the proposed semantics – which is unfortunately a much harder task to solve for expressive logics. If an incomplete reasoning mechanism is provided, we can conclude either that the semantics of the representation language does not really capture the meaning of natural language, or that the algorithms can not infer all the things we would expect. In both cases, the picture remains partial. We can conclude that only if appropriate procedures are provided for reasoning on the logical forms, a natural language understanding system can be used in practical applications, even in simple situations.

Our work is closer to the semantic-oriented works, but it follows a minimalist conceptualization, and it emphasizes the computational aspects. Instead of trying to solve sophisticated semantic problems of natural language, it tries to logically reconstruct some *basic* issues in a general way, which is *homogeneous*, *principled*, and interesting from an applicative point of view. We introduce a logic able to express ambiguous logical forms together with domain and contextual knowledge. We consider here the basic cases of representing and completely reasoning with PP-attachment ambiguities, and with lexical ambiguities introduced by prepositions, nouns, and verbs. Then, we will propose also an extended framework considering simple cases of quantification scoping ambiguities. The basic reasoning task we consider is satisfiability of formulas.

The Logical Form

The main idea of this paper is to take logical forms seriously: they do not only represent the literal meaning of the fragment, but also lexical ambiguities, represent unresolved referents via variables and equality, interpret plural entities and (generalized) quantifiers, and are linked to a rich theory of the domain. To that purpose, an expressive logical language should have a proper reasoning mechanism, and nonetheless be compositional.

A basic property of a logical form as a semantic representation of a constituent is *compositionality* (Schubert & Pelletier 1982), i.e. the semantic representation of a constituent is a function of the semantic interpretation of its sub-constituents. Thus, a close correspondence between syntactic structure and logical form is allowed. In this way, a parser working according to some grammar rules can incrementally build up the semantic interpretation using the corresponding semantic *rules of logical composition* – specifying how the *logical terms* associated to the sub-constituents are to be combined in order to give the formula for the constituent. Thus, each lexeme has associated a (possibly complex) logical term, which forms its contribution to the meaning of the utterance it is part of.

In the context of such a formalism, an effective semantic lexical discrimination process could be carried on during parsing, by cutting out the exponential factor due to the explicit treatment all the possible derivations. Semantically unplausible interpretations can be discarded, by checking – whenever the parser tries to build a constituent – the inconsistency of the logical form compositionally obtained at that stage. This leaves out many syntactically plausible but semantically implausible interpretation. Such a discrimination step is highly effective in restricted domain applications, where the world knowledge considerably reduces the number of possible models. Clearly, the more the *contextual* knowledge is taken into consideration when evaluating a logical form, the more effective is the discrimination process. Thus, consistency checking of logical forms plays the role of a generalized selectional restrictions mechanism.

The Logical Form as a Conjunctive Query

The main assumption we take here is the following:

> A logical form is expressed as a conjunction of unary and binary predicates, where all the variables are existentially quantified; constants may also appear instead of existential variables.

In the knowledge representation and database literature such a class of closed first order formulae is called *Conjunctive Query* (Chandra & Merlin 1993). This very general assumption seems to be quite appropriate for the direct representation of literal propositional content. For example, *A book of Noam* might be given the following logical form

$$\exists \mathtt{b}.\ \mathtt{Book}(\mathtt{b}) \wedge \mathtt{OF}(\mathtt{b}, \mathtt{noam}).$$

The possibility of having variables and constants allows for the representation of referential ambiguities. Assuming the davidsonian view that eventualities explicitly introduce variables, *A book written by Noam* might be given the logical form

$$\exists \mathtt{b}, \mathtt{e}.\ \mathtt{Book}\ (\mathtt{b}) \wedge \mathtt{Write}\ (\mathtt{e}) \wedge \mathtt{AGENT}\ (\mathtt{e}, \mathtt{noam}) \wedge \mathtt{PATIENT}\ (\mathtt{e}, \mathtt{b}). \quad (1)$$

As we have pointed out above, consistency checking of the logical form may help in the semantic discrimination process. Thus, we would like to discard sentences like *Len eats the book*, since its logical form

$$\exists \mathtt{b}, \mathtt{e}.\ \mathtt{Book}\ (\mathtt{b}) \wedge \mathtt{Eat}\ (\mathtt{e}) \wedge \mathtt{AGENT}\ (\mathtt{e}, \mathtt{len}) \wedge \mathtt{PATIENT}\ (\mathtt{e}, \mathtt{b})$$

is inconsistent with respect to a general theory of books and edible things that we attach to the lexicon:

$$\forall \mathtt{x}, \mathtt{y}.\ \mathtt{Eat}(\mathtt{x}) \rightarrow (\mathtt{PATIENT}(\mathtt{x}, \mathtt{y}) \rightarrow \mathtt{Edible}(\mathtt{y}))$$

$$\forall \mathtt{x}.\ \mathtt{Book}(\mathtt{x}) \rightarrow \neg\mathtt{Edible}(\mathtt{x}).$$

Such an axiomatic theory plays the role of *meaning postulates* for the predicates appearing in the logical form; they can be also considered as a set of *predicate level axioms*. We will use from here on – as an alternative simplified notation for *predicate level theories* – the formalism based on Description Logics which will be intuitively introduced in section :

$$\mathtt{Eat} \sqsubseteq \forall \mathtt{PATIENT}.\mathtt{Edible}$$

$$\mathtt{Book} \sqsubseteq \neg\mathtt{Edible}.$$

In a similar way, a general theory of common-sense knowledge will allow only for one interpretation of *Len reads the book with the glasses* where the PP attaches to the reading event – "*reads with the glasses*" – ruling out the interpretation "*the book wearing glasses*". Clearly, we exclude here any creative or metaphoric aspect of language. General theories – so called *Upper Models* – representing the basic ontological properties of common-sense world knowledge supporting lexical discrimination tasks have been studied (see, e.g., (Bateman, Magnini, & Fabris 1995)). It is conceivable to integrate such Upper Models with other linguistically motivated knowledge bases, most notably WordNet.

Representing Ambiguities

Lexical Ambiguities Let us consider the *deep* meaning of *A book of Noam*. The sentence is ambiguous (at least) with respect the two readings *A book written by Noam* and *A book owned by Noam*. We could enumerate such an ambiguity explicitly in the logical form, by disjoining the logical forms of the two readings. However, it is infeasible to explicit enumerate all the (possibly exponential) number of readings. If we assume to have a language at predicate level with expressivity at least as propositional calculus, we can write

```
∃b. Book (b) ∧
    (WRITTEN-BY ⊔ OWNED-BY)(b, noam)     (2)
```

with the meaning that the relation among `b` and `noam` is the union of the relations `WRITTEN-BY` and `OWNED-BY`. Each of the two readings clearly entails this *ambiguous* representation. In this way, the lexicon – which can be considered as an associated theory – may contain a meaning postulate for the relation `OF`:

$$\forall \mathtt{x}, \mathtt{y}.\ \mathtt{OF}(\mathtt{x}, \mathtt{y}) \Leftrightarrow \mathtt{WRITTEN{-}BY}(\mathtt{x}, \mathtt{y}) \vee \mathtt{OWNED{-}BY}(\mathtt{x}, \mathtt{y})$$

which can be rewritten as

$$\mathtt{OF} \doteq \mathtt{WRITTEN{-}BY} \sqcup \mathtt{OWNED{-}BY}.$$

Moreover, if we were allowed to write *reification* axioms (see (Franconi & Rabito 1994) for a translation in Description Logics) of the kind

```
∀x,y. WRITTEN-BY(y,x) ⇔
      ∃z. Write(z) ∧
          AGENT(z,x) ∧ PATIENT (z,y)
```

then, the logical form (1) with the explicit event also entails the ambiguous representation (2). The ambiguity of *A book of Noam* can be *monotonically* refined later on in the dialog by uttering, e.g., either *Noam wrote the book in 1966* or *Noam sold the book*. The refinement process is monotonic, since it is not necessary to revise the knowledge asserted by means of the logical form (2).

In a similar way, lexical ambiguities of nouns can be represented, as in the example *The pilot was out* – where pilot can be a small flame used to start a furnace, or a person who flies airplane. The sentence *He was at the toilet* monotonically refines the previous one, because the pronoun *he* may refer just to a person, thus leaving out the reading with flame. Of course, in order to make possible such a reasoning by cases, axioms at the predicate level having negation and, more generally, partitioning capabilities have to be added to the theory – specifying and reducing the possible models:

$$\mathtt{Pilot} \doteq \mathtt{Flame} \sqcup \mathtt{Aviator}$$

$$\mathtt{Flame} \sqsubseteq \mathtt{Process}$$

$$\mathtt{Aviator} \sqsubseteq \mathtt{Human};\ \mathtt{Human} \sqsubseteq \mathtt{Animate}$$

$$\mathtt{Animate} \sqcap \mathtt{Process} \doteq \bot.$$

Verb ambiguity is also captured in the same manner. For example, it is possible to rule out the sentence *The door opens the door*, given the two senses of *open* as "cause to open" – transitive, with an animate agent – and "become open" – intransitive. According to these two senses, both the constituents "*The door opens*" and "*opens the door*" are consistent, but the whole sentence is inconsistent.

Quantification Scoping Another source of ambiguity is quantification. Current proposals in the logical form of quantified linguistic expressions do not satisfy the *Conjunctive Query* assumption. For example, a common way of representing *Three men read the book* – where the ambiguity is among the cumulative / distributive/ collective readings – is something like

$$\exists \mathtt{b}.\ \mathtt{READ}(<\mathtt{3m}.\ \mathtt{Man}(\mathtt{m})>, \mathtt{b}).$$

Please note that for sake of simplicity we will not use thematic roles in the examples of this section.

Our proposal (Franconi 1993) shifts the quantifier representation – just like lexical ambiguity – at the predicate level, by introducing plural entities and plural quantification

$$\exists \mathtt{b}, \hat{\mathtt{m}}.\ (\mathtt{3.}\ \mathtt{Man})(\hat{\mathtt{m}}) \wedge \mathtt{Book}(\mathtt{b}) \wedge (\trianglelefteq\mathtt{READ})(\hat{\mathtt{m}}, \mathtt{b}).$$

This approach is fully compositional and satisfies the *Conjunctive Query* assumption. The numeric operator introduces the plural entity $\hat{\mathtt{m}}$, made out of three men. The $\trianglelefteq$ plural quantifier denote the cumulative reading of the relation READ with respect to the first plural argument position. This representation concisely captures all the possible readings of the sentence: there are three distinct events of reading, the agent of each one being each man; there is a single event of reading, whose actor is the whole group of three men; there is an intermediate situation with two reading events, with actor sub-groupings of the men. It can be shown that any possible explicit reading of the sentence semantically entails the above concise logical form. A subsequent sentence could inform that either *Each one of them reads the book* – thus, monotonically refining the left cumulative reading into a left *distributive* reading

$$(\lhd\mathtt{READ})(\hat{\mathtt{m}},\mathtt{b}),$$

or *They read the book all together* – thus, monotonically refining the left cumulative reading into a *collective* reading

$$\mathtt{READ}(\hat{\mathtt{m}},\mathtt{b}).$$

The lexicon may contain some axioms like the necessity for an eventive predicate of a certain type to be inherently, say, distributive with respect to a thematic role; this is the case of *to be born*. In fact, the sentence *John and Mary are born in Tucson*:

$$\exists \mathtt{g}.\ \ni(\mathtt{g,john}) \wedge \ni(\mathtt{g,mary}) \wedge \\ (\lhd\ \mathtt{BE\text{-}BORN\text{-}IN})(\mathtt{g,tucson})$$

means both "*John is born in Tucson*" and "*Mary is born in Tucson*":

$$\mathtt{BE-BORN-IN}(\mathtt{john},\mathtt{tucson});$$

$$\mathtt{BE-BORN-IN}(\mathtt{mary},\mathtt{tucson}).$$

On the other hand, an event like *meet* is inherently collective with respect to its thematic role:

$$\exists \mathtt{g}.\ \ni(\mathtt{g},\mathtt{john}) \wedge \ni(\mathtt{g},\mathtt{mary}) \wedge \mathtt{MEET-IN}(\mathtt{g},\mathtt{tucson}).$$

So, *John and Mary met in Tucson* means that the whole plural entity composed by both John and Mary is the theme of a meet event.

This well founded approach for plurals and natural language quantifiers makes more effective the consistency checking of the logical form during the semantic discrimination process. For example, it is possible to discard *The mothers of John*, if in the lexicon it is specified that the relation `being-mother` admits only a *singular* entity of type `animate` as its domain. In (Franconi 1993) it is shown how to represent generalized quantifiers with cumulative, distributive and collective reading within this approach – currently limited to logical symmetric determiners. It is outside the scope of this paper to go into the details of this approach.

Knowledge Representation

We have tried to motivate an approach where logical forms are simple conjunctive queries together with a collection of meaning postulates at the predicate level. We do not claim that *in general* conjunctive queries are enough for the semantic representation of natural language. However, the advantages of this principled and compositional approach in handling in a simple and homogeneous way semantic discrimination and the basic phenomena of PP-attachment and lexical ambiguities make it interesting from an applicative point of view. Thus, it turns out that what is needed by a knowledge representation system is the ability to correctly *reason* with conjunctive queries *together with* rich predicate level theories.

Description Logics (DL; see, e.g., (Buchheit, Donini, & Schaerf 1993) for a survey) – i.e. knowledge representation languages of the KL-ONE family, also called sometimes Frame-Based Description Languages, Terminological Logics, Taxonomic Logics or Concept Languages – are the right candidate for representing predicate level theories. The strength of DL is the attention which is devoted to the study of the computational properties: DL are a decidable fragment of predicate logic. Most DL do have complete algorithms, whose complexity has been formally studied. Basic types are *concepts* (unary predicates), *roles* (binary predicates), *features* (functions) and *individuals* (constants). The basic DL is $\mathcal{ALC}$, which includes full propositional calculus plus universal and existential quantifications over roles. Sound and complete reasoning procedures for $\mathcal{ALC}$ do exist, and efficient implementations are currently being used in the research community (Bresciani, Franconi, & Tessaris 1995). We will not give here any technical detail about description logics; (Franconi 1994) provides examples of using DL for capturing some NL phenomena.

Of course, in order to justify the strong statement of this section, we need to exhibit the following remarks:

Expressivity. All the examples of meaning postulates in the previous section can be expressed within $\mathcal{ALC}$, with the exception of reification and plural quantifiers; the papers (Franconi & Rabito 1994) and (Franconi 1993) show extensions of $\mathcal{ALC}$ dealing with these cases.

Reasoning with Conjunctive Query. The satisfiability problem (i.e. consistency checking) for conjunctive queries with an associated theory expressed with description logics is a decidable problem if the DL itself has a decidable satisfiability problem and if it includes the `singleton` operator (Schaerf 1994). In particular, if we use $\mathcal{ALC}$ then satisfiability is in PSPACE. The algorithm computing satisfiability for conjunctive queries is a simple variant of the tableaux-based algorithm of the DL itself. This result is quite new and it is still unpublished.

The two above observations give a strong motivation for at least trying to study this approach. In fact, it is said that the problem of consistency checking of logical form can be reduced to the corresponding problem in a pure description logic setting. Moreover, it is shown how the basic needs for representing lexical discrimination, ambiguities and some quantifier scoping phenomena are already satisfied. Until now description logics have been used in NLP mainly for representing case frames attached to world knowledge. Now, studies extending description logics in the direction of capturing NL phenomena (Franconi 1994) can be seen under a new light: they can be immediately integrated for augmenting the effectiveness of logical form consistency checking.

This approach explains what we've said in the introduction: a KR formalism like DL is attached to the logical form in a principled and complete way. Thus, it is possible to maintain both the logical form approach to natural language semantic interpretation, and the description logic based knowledge representation language for expressing meaning postulates and for reasoning purposes. Moreover, this representation satisfies the three requirements discussed by (Allen 1993):

a. The KR must distinguish between ambiguity and disjunction;

b. The KR must support inference over ambiguity; and

c. The KR must support disambiguation techniques.

We are successfully testing such an approach in the context of an application for the categorization of information extracted from text. A bi-directional bottom-up chart parser is connected to a TFS (Typed Feature Structure) unification based system, which includes an HPSG-like grammar and the lexicon (Ciravegna, Lavelli, & Pianesi 1996). Whenever a unification among two constituents is attempted, a consistency check for the resulting logical form is performed, using the CRACK description logic based system (Bresciani, Franconi, & Tessaris 1995).

Conclusions

It has been shown how a Description Logics formalism can be profitably used for representing meaning postulates for the predicates appearing in a logical form of a natural language utterance. We do not know how far this framework can be pushed. DL have the required expressivity to correctly and completely handle lexical discrimination, PP-attachment and lexical ambiguities, and are a promising technique also for handling quantification scoping ambiguities. Within this new picture, studies for representing NL phenomena with DL are well motivated.

Acknowledgment

I would like to thank Werner Nutt – who introduced me into the technicalities of Conjunctive Queries – and Fabio Pianesi – who tried to make me understand the point of view of a (computational) linguist.

References

Allen, J. F. 1993. Natural language, knowledge representation, and logical form. In Bates, M., and Weischedel, R., eds., *Challenges in Natural Language Processing*. Cambridge University Press.

Bateman, J. A.; Magnini, B.; and Fabris, G. 1995. The generalized upper model knowledge base: Organization and use. In *Proceedings of the Second International Conference on Bulding and Sharing of Very Large-Scale Knowledge Bases.*

Bresciani, P.; Franconi, E.; and Tessaris, S. 1995. Implementing and testing expressive description logics: a preliminary report. In *Proceedings of the 1995 International Workshop on Description Logics*, 131–139. Also in the Proceedings of the International KRUSE Symposium, Santa Cruz; August 1995.

Buchheit, M.; Donini, F. M.; and Schaerf, A. 1993. Decidable reasoning in terminological knowledge representation systems. In *Proc. of the 13th IJCAI*, 704–709.

Chandra, A. K., and Merlin, P. M. 1993. Optimal implementation of conjunctive queries in relational databases. In *Proc. 9th Annual ACM Symposium on Theory of Computing*, 77–90. New York (NY, USA): ACM.

Ciravegna, F.; Lavelli, A.; and Pianesi, F. 1996. Linguistic processing of texts using Geppetto. Technical Report 9602-06, IRST, Povo TN, Italy.

Franconi, E., and Rabito, V. 1994. A relation-based description logic. In *Proceedings of the International Workshop on Description Logics.* Bonn, Germany. 62–66.

Franconi, E. 1993. A treatment of plurals and plural quantifications based on a theory of collections. *Minds and Machines* 3(4):453–474. Kluwer Academic Publishers. Special Issue on Knowledge Representation for Natural Language Processing.

Franconi, E. 1994. Description logics for natural language processing. In *Working Notes of the 1994 AAAI Fall Symposium on "Knowledge Representation for Natural Language Processing in Implemented Systems"*, 37–44. Also presented at the European Research Conference on "Logic, Language and Information", organized by the European Science Foundation (ESF); Espinho, Portugal, December 1994.

Schaerf, A. 1994. Reasoning with individuals in concept languages. *Data and Knowledge Engineering* 13(2):141–176.

Schubert, L. K., and Pelletier, F. J. 1982. From english to logic: Context-free computation of 'conventional' logical translations. *Computational Linguistics* 8:165–176.