

LOGICAL FORMS IN THE CORE LANGUAGE ENGINE

Hiyan Alshawi & Jan van Eijck
SRI International Cambridge Research Centre
23 Millers Yard, Mill Lane, Cambridge CB2 1RQ, U.K.

Keywords: logical form, natural language, semantics

ABSTRACT

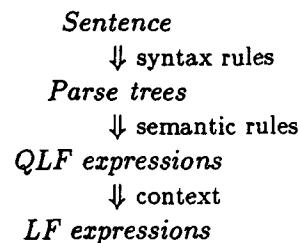
This paper describes a 'Logical Form' target language for representing the literal meaning of English sentences, and an intermediate level of representation ('Quasi Logical Form') which engenders a natural separation between the compositional semantics and the processes of scoping and reference resolution. The approach has been implemented in the SRI Core Language Engine which handles the English constructions discussed in the paper.

INTRODUCTION

The SRI Core Language Engine (CLE) is a domain independent system for translating English sentences into formal representations of their literal meanings which are capable of supporting reasoning (Alshawi *et al.* 1988). The CLE has two main levels of semantic representation: quasi logical forms (QLFs), which may in turn be scoped or unscoped, and fully resolved logical forms (LFs). The level of quasi logical form is the target language of the syntax-driven semantic interpretation rules. Transforming QLF expressions into LF expressions requires (i) fixing the scopes of all scope-bearing operators (quantifiers, tense operators, logical operators) and distinguishing distributive readings of noun phrases from collective ones, and (ii) resolving referential expressions such as definite descriptions, pronouns, indexical expressions, and underspecified relations.

The QLF level can be regarded as the natural level of sentence representation resulting

from linguistic analysis that applies compositional semantic interpretation rules independently of the influence of context.



The QLF expressions are derived on the basis of syntactic structure, by means of semantic rules that correspond to the syntax rules that were used for analysing the sentence. Having QLFs as a well-defined level of representation allows the problems of compositional semantics to be tackled separately from the problems of scoping and reference resolution. Our experience so far with the CLE has shown that this separation can effectively reduce the complexity of the system as a whole. Also, the distinction enables us to avoid multiplying out interpretation possibilities at an early stage. The representation languages we propose are powerful enough to give well-motivated translations of a wide range of English sentences. In the current version of the CLE this is used to provide a systematic and coherent coverage of all the major phrase types of English. To demonstrate that the semantic representations are also simple enough for practical natural language processing applications, the CLE has been used as an interface to a purchase order processing simulator and a database query system, to be described elsewhere.

In summary, the main contributions of the

work reported in this paper are (i) the introduction of the QLF level to achieve a natural separation between compositional semantics and the processes of scoping and reference resolution, and (ii) the integration of a range of well-motivated semantic analyses for specific constructions in a single coherent framework.

We will first motivate our extensions to first order logic and our distinction between LF and QLF, then describe the LF language, illustrating the logical form translations produced by the CLE for a number of English constructions, and finally present the additional constructs of the QLF language and illustrate their use.

EXTENDING FIRST ORDER LOGIC

As the pioneer work by Montague (1973) suggests, first order logic is not the most natural representation for the meanings of English sentences. The development of Montague grammar indicates, however, that there is quite a bit of latitude as to the scope of the extensions that are needed. In developing the LF language for the CLE we have tried to be conservative in our choice of extensions to first order logic. Earlier proposals with similar motivation are presented by Moore (1981) and Schubert & Pelletier (1982).

The ways in which first order logic—predicate logic in which the quantifiers \exists and \forall range over the domain of individuals—is extended in our treatment can be grouped and motivated as follows:

- Extensions motivated by lack of expressive power of ordinary first order logic: for a general treatment of noun phrase constructions in English generalized quantifiers are needed ('Most A are B' is not expressible in a first order language with just the two one-place predicates A and B).
- Extensions motivated by the desire

for an elegant compositional semantic framework:

- use of lambda abstraction for the translation of graded predicates in our treatment of comparatives and superlatives;
- use of tense operators and intensional operators for dealing with the English tense and auxiliary system in a compositional way.
- Extensions motivated by the desire to separate out the problems of scoping from those of semantic representation.

- Extensions motivated by the need to deal with context dependent constructions, such as anaphora, and the implicit relations involved in the interpretation of possessives and compound nominals.

The first two extensions in the list are part of the LF language, to be described next, the other two have to do with QLF constructs. These QLF constructs are removed by the processes of quantifier scoping and reference resolution (see below).

The treatment of tense by means of temporal operators that is adopted in the CLE will not be discussed in this paper. Some advantages of an operator treatment of the English tense system are discussed in (Moore, 1981).

We are aware of the fact that some aspects of our LF representation give what are arguably overly neutral analyses of English constructions. For example, our uses of event variables and of sentential tense operators say little about the internal structure of events or about an underlying temporal logic. Nevertheless, our hope is that the proposed LF representations form a sound basis for the subsequent process of deriving the fuller meaning representations.

RESOLVED LOGICAL FORMS

NOTATIONAL CONVENTIONS

Our notation is a straightforward extension of the standard notation for first order logic. The following logical form expression involving restricted quantification states that every dog is nice:

$\text{quant}(\text{forall}, x, \text{Dog}(x), \text{Nice}(x)).$

To get a straightforward treatment of the collective/distributive distinction (see below) we assume that variables always range over sets, with ‘normal’ individuals corresponding to singletons. Properties like *being a dog* can be true of singletons, e.g. the referent of *Fido*, as well as larger sets, e.g. the referent of *the three dogs we saw yesterday*.

The LF language allows formation of complex predicates by means of lambda abstraction: $\lambda x \lambda d. \text{Heavy_degree}(x, d)$ is the predicate that expresses degree of heaviness.

EVENT AND STATE VARIABLES

Rather than treating modification of verb phrases by means of higher order predicate modifiers, as in (Montague, 1973), we follow Davidson’s (1967) quantification over events to keep closer to first order logic. The event corresponding to a verb phrase is introduced as an additional argument to the verb predicate. The full logical form for *Every representative voted* is as follows:

$\text{quant}(\text{forall}, x, \text{Repr}(x),$
 $\text{past}(\text{quant}(\text{exists}, e, \text{Ev}(e),$
 $\text{Vote}(e, x))).$

Informally, this says that for every representative, at some past time, there existed an event of that representative voting.

The presence of an event variable allows us to treat optional verb phrase modifiers as predications of events, as in the translation of *John left suddenly*:

$\text{past}(\text{quant}(\text{exists}, e, \text{Ev}(e),$

$\text{Leave}(e, \text{john}) \wedge \text{Sudden}(e))).$

The use of event variables in turn permits us to give a uniform interpretation of prepositional phrases, whether they modify verb phrases or nouns. For example, *John designed a house in Cambridge* has two readings, one in which *in Cambridge* is taken to modify the noun phrase *a house*, and one where the prepositional phrase modifies the verb phrase, with the following translations respectively:

$\text{quant}(\text{exists}, h,$
 $\text{House}(h) \wedge \text{In_location}(h, \text{Cambridge}),$
 $\text{past}(\text{quant}(\text{exists}, e, \text{Ev}(e),$
 $\text{Design}(e, \text{john}, h))).$

$\text{quant}(\text{exists}, h, \text{House}(h) \wedge$
 $\text{past}(\text{quant}(\text{exists}, e, \text{Ev}(e),$
 $\text{Design}(e, \text{john}, h) \wedge$
 $\text{In_location}(e, \text{Cambridge}))).$

In both cases the prepositional phrase is translated as a two-place relation stating that something is located in some place. Where the noun phrase is modified, the relation is between an ordinary object and a place; in the case where the prepositional phrase modifies the verb phrase the relation is between an event and a place. Adjectives in predicative position give rise to *state variables* in their translations. For example, in the translation of *John was happy in Paris*, the prepositional phrase modifies the state. States are like events, but unlike events they cannot be instantaneous.

GENERALIZED QUANTIFIERS

A generalized quantifier is a relation Q between two sets A and B , where Q is insensitive to anything but the cardinalities of the ‘restriction set’ A and the ‘intersection set’ $A \cap B$ (Barwise & Cooper, 1981). A generalized quantifier with restriction set A and intersection set $A \cap B$ is fully characterized by a function $\lambda m \lambda n. Q(m, n)$ of m and n , where $m = |A|$ and $n = |A \cap B|$. In the LF language of the CLE, these quantifier relations are expressed by means of predicates on two

numbers, where the first variable abstracted over denotes the cardinality of the restriction set and the second one the cardinality of the intersection set. This allows us to build up quantifiers for complex specifier phrases like *at least three but less than five*. In simple cases, the quantifier predicates are abbreviated by means of mnemonic names, such as **exists**, **notexists**, **forall** or **most**. Here are some quantifier translations:

- *most* $\rightsquigarrow \lambda m \lambda n. (m < 2n)$ [abbreviation: **most**].
- *at least three but less than seven* $\rightsquigarrow \lambda m \lambda n. (n \geq 3 \wedge n < 7)$.
- *not every* $\rightsquigarrow \lambda m \lambda n. (m \neq n)$.

A logical form for *Not every representative voted* is:

```
quant(λmλn.(m ≠ n), x, Rep(x),
      past(quant(exists, e, Ev(e),
                 Vote(e, x)))).
```

Note that in one of the quantifier examples above the abstraction over the restriction set is vacuous. The quantifiers that do depend only on the cardinality of their intersection set turn out to be in a linguistically well-defined class: they are the quantifiers that can occur in the NP position in “There are NP”. This quantifier class can also be characterized logically, as the class of *symmetric* quantifiers: “At least three but less than seven men were running” is true just in case “At least three but less than seven runners were men” is true; see (Barwise & Cooper, 1981) and (Van Eijck, 1988) for further discussion. Below the logical forms for symmetric quantifiers will be simplified by omitting the vacuous lambda binder for the restriction set. The quantifiers for collective and measure terms, described in the next section, seem to be symmetric, although linguistic intuitions vary on this.

COLLECTIVES AND MEASURE TERMS

Collective readings are expressed by an extension of the quantifier notation using **set**.

The reading of *Two companies ordered five computers* where the first noun phrase is interpreted collectively and the second one distributively is expressed by the following logical form:

```
quant(set(λn.(n = 2)), x, Company(x),
      quant(λn.(n = 5), y,
            Computer(y),
            past(quant(exists, e, Ev(e),
                       Order(e, x, y))))).
```

The first quantification expresses that there is a collection of two companies satisfying the body of the quantification, so this reading involves five computers and five buying events. The operator **set** is introduced during scoping since collective/distributive distinctions—like scoping ambiguities—are not present in the initial QLF.

We have extended the generalized quantifier notation to cover phrases with measure determiners, such as *seven yards of fabric* or *a pound of flesh*. Where ordinary generalized quantifiers involve *counting*, *amount* generalized quantifiers involve *measuring* (according to some measure along some appropriate dimension). Our approach, which is related to proposals that can be found in (Pelletier, ed., 1979) leads to the following translation for *John bought at least five pounds of apples*:

```
quant(amount(λn.(n ≥ 5), pounds),
      x, Apple(x),
      past(quant(exists, e, Ev(e),
                 Buy(e, john, x)))).
```

Measure expressions and numerical quantifiers also play a part in the semantics of comparatives and superlatives respectively (see below).

NATURAL KINDS

Terms in logical forms may either refer to individual entities or to natural kinds (Carlson, 1977). Kinds are individuals of a specific nature; the term **kind**(*x*, *P*(*x*)) can loosely be interpreted as the typical individual satisfying *P*. All properties, including composite ones, have a corresponding natural kind in

our formalism. Natural kinds are used in the translations of examples like *John invented paperclips*:

```
past(quant(exists,e,Ev(e),
Invent(e,john,kind(p,Paperclip(p))))).
```

In reasoning about kinds, the simplest approach possible would be to have a rule of inference stating that if a “kind individual” has a certain property, then all “real world” individuals of that kind have that property as well: if the “typical bear” is an animal, then all real world bears are animals. Of course, the converse rule does not hold: the “typical bear” cannot have all the properties that any real bear has, because then it would have to be both white all over and brown all over, and so on.

COMPARATIVES AND SUPERLATIVES

In the present version of the CLE, comparatives and superlatives are formed on the basis of degree predicates. Intuitively, the meaning of the comparative in *Mary is nicer than John* is that one of the two items being compared possesses a property to a higher degree than the other one, and the meaning of a superlative is that an item possesses a property to the highest degree among all the items in a certain set. This intuition is formalised in (Cresswell, 1976), to which our treatment is related.

The comparison in *Mary is two inches taller than John* is translated as follows:

```
quant(amount(λn.(n = 2),inches),
h,Degree(h),
more(λxλd.tall_degree(x,d),
mary,john,h)).
```

The operator **more** has a graded predicate as its first argument and three terms as its second, third and fourth arguments. The operator yields true if the degree to which the first term satisfies the graded predicate exceeds the degree to which the second term satisfies the predicate by the amount specified in the final term. In this example *h* is a

degree of height which is measured, in inches, by the amount quantification. Examples like *Mary is 3 inches less tall than John* get similar translations. In *Mary is taller than John* the quantifier for the degree to which Mary is taller is simply an existential.

Superlatives are reduced to comparatives by paraphrasing them in terms of the number of individuals that have a property to at least as high a degree as some specific individual. This technique of comparing pairs allows us to treat combinations of ordinals and superlatives, as in *the third tallest man smiled*:

```
quant(ref(the,...),a,
Man(a) ∧ quant(λn.(n = 3),b,
Man(b)),
quant(amount(λn.(n ≥ 0),units),h,
more(λxλd.tall_degree(x,d),b,a,h),
past(quant(exists,e,Ev(e),
Smile(e,a))))).
```

The logical form expresses that there are exactly three men whose difference in height from *a* (the referent of the definite noun phrase, see below) is greater than or equal to 0 in some arbitrary units of measurement.

QUASI LOGICAL FORMS

The QLF language is a superset of the LF language; it contains additional constructs for unscoped quantifiers, unresolved references, and underspecified relations. The ‘meaning’ of a QLF expression can be thought of as being given in terms of the meanings of the set of LF expressions it is mapped to. Ultimately the meaning of the QLF expressions can be seen to depend on the contextual information that is employed in the processes of scoping and reference resolution.

UNSCOPED QUANTIFIERS

In the QLF language, unscoped quantifiers are translated as terms with the format

```
qterm(<quantifier>,<number>,
<variable>,<restriction>).
```

Coordinated NPs, like *a man or a woman*, are translated as terms with the format

```
term_coord(<operator>,<variable>,
           <term>, <term>).
```

The unscoped QLF generated by the semantic interpretation rules for *Most doctors and some engineers read every article* involves both qterms and a term_coord (quantifier scoping generates a number of scoped LFs from this):

```
quant(exists,e,Ev(e),
      Read(e,
            term_coord(∧,x,
                        qterm(most,plur,y,Doctor(y)),
                        qterm(some,plur,z,Engineer(z))),
                        qterm(every,sing,v,Art(v)))).
```

Quantifier scoping determines the scopes of quantifiers and operators, generating scoped logical forms in a preference order. The ordering is determined by a set of declarative rules expressing linguistic preferences such as the preference of particular quantifiers to outscope others. The details of two versions of the CLE quantifier scoping mechanism are discussed by Moran (1988) and Pereira (Alshawi *et al.* 1988).

UNRESOLVED REFERENCES

Unresolved references arising from pronoun anaphora and definite descriptions are represented in the QLF as ‘quasi terms’ which contain internal structure relevant to reference resolution. These terms are eventually replaced by ordinary LF terms (constants or variables) in the final resolved form. A discussion of the CLE reference resolution process and treatment of constraints on pronoun reference will be given in (Alshawi, in prep.).

Pronouns. The QLF representation of a pronoun is an anaphoric term (or a_term). For example, the translations of *him* and *himself* in *Mary expected him to introduce himself* are as follows:

```
a_term(ref(pro,him,sing,[mary]),
      x, Male(x))
a_term(ref(refl,him,sing,[x,mary]),
      y, Male(y)).
```

The first argument of an a_term is akin to a category containing the values of syntactic and semantic features relevant to reference resolution, such as those for the reflexive/non-reflexive and singular/plural distinctions, and a list of the possible intrasentential antecedents, including quantified antecedents.

Definite Descriptions. Definite descriptions are represented in the QLF as unscoped quantified terms. The qterm is turned into a quant by the scoper, and, in the simplest case, definite descriptions are resolved by instantiating the quant variable in the body of the quantification. Since it is not possible to do this for descriptions containing bound variable anaphora, such descriptions remain as quantifiers. For example, the QLF generated for the definite description in *Every dog buried the bone that it found* is:

```
qterm(ref(def,the,sing,[x]),sing,y,
      Bone(y) ∧ past(quant(exists,e,Ev(e),
                           Find(e,a_term(ref(pro,it,sing,[y,x]),
                                         w,Impersonal(w)),y)))).
```

After scoping and reference resolution, the LF translation of the example is as follows:

```
quant(forall,x,Dog(x),
      quant(exists-one,y,
            Bone(y) ∧ past(quant(exists,e,Ev(e),
                                  Find(e,x,y))),
            quant(exists,e',Ev(e'),Bury(e',x,y)))).
```

Unbound Anaphoric Terms. When an argument position in a QLF predication must co-refer with an anaphoric term, this is indicated as a_index(*x*), where *x* is the variable for the antecedent. For example, because *want* is a subject control verb, we have the following QLF for *he wanted to swim*:

```

past(quant(exists, e, Ev(e),
Want(e, a_term(ref(pro, he, sing, [/]), x,
Male(x)),
quant(exists, e', Ev(e'),
Swim(e', a_index(x))))).

```

If the `a_index` variable is subsequently resolved to a quantified variable or a constant, then the `a_index` operator becomes redundant and is deleted from the resulting LF. In special cases such as the so-called ‘donkey-sentences’, however, an anaphoric term may be resolved to a quantified variable v outside the scope of the quantifier that binds v . The LF for *Every farmer who owns a dog loves it* provides an example:

```

quant(forall, x,
Farmer(x) ∧
quant(exists, y, Dog(y),
quant(exists, e, Ev(e), Own(e, x, y))),
quant(exists, e', Ev(e'),
Love(e', x, a_index(y))).

```

The ‘unbound dependency’ is indicated by an `a_index` operator. Dynamic interpretation of this LF, in the manner proposed in (Groenendijk & Stokhof, 1987), allows us to arrive at the correct interpretation.

UNRESOLVED PREDICATIONS

The use of unresolved terms in QLFs is not sufficient for covering natural language constructs involving implicit relations. We have therefore included a QLF construct (`a_form` for ‘anaphoric formula’) containing a formula with an unresolved predicate. This is eventually replaced by a fully resolved LF formula, but again the process of resolution is beyond the scope of this paper.

Implicit Relations. Constructions like possessives, genitives and compound nouns are translated into QLF expressions containing uninstantiated relations introduced by the `a_form` relation binder. This binder is used in the translation of *John’s house* which says that a relation, of type `poss`, holds between John and the house:

```

qterm(exists, sing, x,
a_form(poss, R, House(x) ∧ R(john, x))).

```

The implicit relation, R , can then be determined by the reference resolver and instantiated, to *Owes* or *Lives_in* say, in the resolved LF.

The translation of indefinite compound nominals, such as *a telephone socket*, involves an `a_form`, of type `cn` (for an unrestricted compound nominal relation), with a ‘kind’ term:

```

qterm(a, sing, s,
a_form(cn, R, Socket(s) ∧
R(s, kind(t, Telephone(t))).

```

The ‘kind’ term in the translation reflects the fact that no individual telephone needs to be involved.

One-Anaphora. The `a_form` construct is also used for the QLF representation of ‘one-anaphora’. The variable bound by the `a_form` has the type of a one place predicate rather than a relation. Resolving these anaphora involves identifying relevant (parts of) preceding noun phrase restrictions (Webber, 1979). For example the scoped QLF for *Mary sold him an expensive one* is:

```

quant(exists, x,
a_form(one, P, P(x) ∧ Expensive(x)),
past(quant(exists, e, Ev(e),
Sell(e, mary, x, a_term(⋯))).

```

After resolution (if the sentence were preceded, say, by *John wanted to buy a futon*) the resolved LF would be:

```

quant(exists, x,
Futon(x) ∧ Expensive(x),
past(quant(exists, e, Ev(e),
Sell(e, mary, x, john))).

```

CONCLUSION

We have attempted to evolve the QLF and LF languages gradually by a process of adding minimal extensions to first order logic, in order to facilitate future work on

natural language systems with reasoning capabilities. The separation of the two semantic representation levels has been an important guiding principle in the implementation of a system covering a substantial fragment of English semantics in a well-motivated way. Further work is in progress on the treatment of collective readings and of tense and aspect.

ACKNOWLEDGEMENTS

The research reported in this paper is part of a group effort to which the following people have also contributed: David Carter, Bob Moore, Doug Moran, Barney Pell, Fernando Pereira, Steve Pulman and Arnold Smith. Development of the CLE has been carried out as part of a research programme in natural-language processing supported by an Alvey grant and by members of the NATTIE consortium (British Aerospace, British Telecom, Hewlett Packard, ICL, Olivetti, Philips, Shell Research, and SRI). We would like to thank the Alvey Directorate and the consortium members for this funding. The paper has benefitted from comments by Steve Pulman and three anonymous ACL referees.

REFERENCES

- Alshawi, H., D.M. Carter, J. van Eijck, R.C. Moore, D.B. Moran, F.C.N. Pereira, S.G. Pulman and A.G. Smith. 1988. *Interim Report on the SRI Core Language Engine*. Technical Report CCSRC-5, SRI International, Cambridge Research Centre, Cambridge, England.
- Alshawi, H., in preparation, "Reference Resolution In the Core Language Engine".
- Barwise, J. & R. Cooper. 1981. "Generalized Quantifiers and Natural Language", *Linguistics and Philosophy*, 4, 159–219.
- Cresswell, M.J. 1976. "The Semantics of Degree", in: B.H. Partee (ed.), *Montague Grammar*, Academic Press, New York, pp. 261–292.
- Carlson, G.N. 1977. "Reference to Kinds in English", PhD thesis, available from Indiana University Linguistics Club.
- Davidson, D. 1967. "The Logical Form of Action Sentences", in N. Rescher, *The Logic of Decision and Action*, University of Pittsburgh Press, Pittsburgh, Pennsylvania.
- van Eijck, J. 1988. "Quantification". Technical Report CCSRC-7, SRI International, Cambridge Research Centre, Cambridge, England. To appear in A. von Stechow & D. Wunderlich, *Handbook of Semantics*, De Gruyter, Berlin.
- Groenendijk, J. & M. Stokhof 1987. "Dynamic Predicate Logic". Preliminary report, ITLI, Amsterdam.
- Montague, R. 1973. "The Proper Treatment of Quantification in Ordinary English". In R. Thomason, ed., *Formal Philosophy*, Yale University Press, New Haven.
- Moore, R.C. 1981. "Problems in Logical Form". *19th Annual Meeting of the Association for Computational Linguistics*, Stanford, California, pp. 117–124.
- Moran, D.B. 1988. "Quantifier Scoping in the SRI Core Language Engine", *26th Annual Meeting of the Association for Computational Linguistics*, State University of New York at Buffalo, Buffalo, New York, pp. 33–40.
- Pelletier, F.J. (ed.) 1979. *Mass Terms: Some Philosophical Problems*, Reidel, Dordrecht.
- Schubert, L.K. & F.J. Pelletier 1982. "From English to Logic: Context-Free Computation of 'Conventional' Logical Translations". *American Journal of Computational Linguistics*, 8, pp. 26–44.
- Webber, B. 1979. *A Formal Approach to Discourse Anaphora*, Garland, New York.