RESOLVING QUASI LOGICAL FORMS

Hiyan Alshawi
SRI International
Cambridge Research Centre
23 Millers Yard
Cambridge CB2 1RQ
U.K.

The paper describes intermediate and resolved logical form representations of sentences involving referring expressions and a reference resolution process for mapping between these representations. The intermediate representation, Quasi Logical Form (or QLF), may contain unresolved terms corresponding to anaphoric noun phrases covering bound variable anaphora, reflexives, and definite descriptions. Implict relations arising in constructs such as compound nominals appear in QLF as unresolved formulae. The QLF representation is also neutral with respect to ambiguities corresponding to quantifier scope and the collective/distributive distinction, the latter being treated as quantifier resolution. Reference candidates are proposed according to an ordered set of "reference resolution rules" producing possible resolved logical forms to which linguistic and pragmatic constraints are then applied.

1 INTRODUCTION

This paper is concerned with the relationship between the linguistic analysis of sentences and their interpretation in context. Our model is described as implemented in the **Core Language Engine** (CLE), a domain-independent system for translating natural language sentences into logical form representations. One of the characteristics of this system is that it embodies an extreme form of a staged natural language processing architecture whereby modular processing components have the function of mapping between different levels of sentence representation. Representations at the level of quasi logical form (QLF) are the result of applying declarative rules for morphological, syntactic, and semantic analysis. These QLF representations have subexpressions corresponding to unresolved referring constructions. The QLF level can be regarded as the natural level of sentence representation resulting from compositional semantic analysis independently of the influence of context:

Sentence–LINGUISTIC ANALYSIS→
QLFs–QLF RESOLUTION IN CONTEXT→*LF.*

QLFs are also neutral with respect to quantifier scope, and the distributive/collective and referential/attributive distinctions. Collective and attributive readings are viewed as resolution to suitable quantifiers in a manner explained in the paper. We will not be discussing the process of quantifier scoping here: a scoping mechanism for the CLE has already been described by Moran (1988), and a later version is described by Pereira in Alshawi et al. (1988).

Formally, it may be possible to regard the resolution process as defining a function from ⟨QLF,context⟩ pairs to LFs. In practice, however, since the contextual information exploited by the CLE resolution process is incomplete, a particular QLF expression may correspond to several LF expressions.

2 DERIVING LFs FROM QLFs

2.1 QLF CONSTRUCTS

The QLF language is a superset of the **LF (logical form)** language used in the CLE (Alshawi and van Eijck 1989). LF is itself an extension of first order logic, including generalized quantifiers and some higher order operators and relations. The additional QLF constructs relevant to reference resolution are a_terms (**anaphoric terms**), q_terms (**quantified terms**), and a_forms (**anaphoric formulae**). In general, the first two arise from anaphoric and quantified noun phrases, while a_forms arise from linguistic expressions that embody implicit relations or ellipsis. a_terms, q_terms and a_forms include linguistic information represented as a 'reference category,' a variable, and a restriction (a QLF formula):

a_term(*Category, Entity Var, Restriction*)
q_term(*Category, Entity Var, Restriction*)
a_form(*Category, Predicate Var, Restriction*).

In a_terms and q_terms the variable stands for a referent, or is simply the variable of an LF quantification, and the formula is a restriction on that variable. In a_forms the variable appears in predicate position in the restriction. We will show categories as a list of feature-value pairs:

$$\langle Feature\ 1 = Value\ 1, \ldots, Feature\ N = Value\ N \rangle$$

(In fact the CLE uses a different, but equivalent, notation for categories.)

All three QLF constructs are introduced explicitly by the semantic analysis rules. For example, the sentence *She met a friend of John* has the following QLF representation in which the a_term is the semantic analysis for *she*, the q_term is the analysis of the quantified noun phrase *a friend of John*, and the a_form is the analysis of *friend of John*:[1]

```
[meet,
  a_term(⟨t=ref,p=pro,l=she,n=sing,a=⟨⟩⟩,
    Y,
    [female,Y]),
  q_term(⟨t=quant,n=sing,l=a⟩,
    X,
    a_form(⟨t=pred,p=genit⟩,
      R,
      [and,[friend,X],[R,john,X]]))]).
```

The QLF expressions corresponding to various linguistic phenomena are described in Section 4.

2.2 QLF Resolution

Basically, the process of QLF resolution involves determining possible LF terms to replace each QLF a_term or definite description, possible quantifiers to replace the determiners for q_terms, and possible formulae to replace each QLF a_form. The quantifier scoping phase determines the scope, or *body*, for each q_term producing scoped expressions that employ a four part generalized quantifier notation (Woods 1978):

quant (*Quantifier, Variable, Restriction, Body*).

In the example above, QLF resolution leads to the following LF in which the a_term is replaced by an intersentential referent mary, the noun phrase determiner is replaced by the quantifier **exists**, and the a_form is simply replaced by a formula involving the relational noun predicate friend_of taken from the lexicon:

```
quant (exists, X,
  [and,[friend,X],[friend_of,john,X]],
  [meet,mary,X]).
```

Proposing resolutions for a QLF is followed by the application of plausibility constraints, both linguistic and nonlinguistic (Section 6). The constraints are applied to a version of the QLF that includes proposed resolutions of referring expressions. This '**resolved quasi logical form,**' or **RQLF,** thus implicitly represents a QLF, an LF, and the correspondence between them. If the proposed RQLF fails to obey these constraints, then reference resolution will generate other possibilities for resolving the QLF; if all such proposed RQLFs fail the tests, other QLFs for the sentence will be considered. Once an RQLF is accepted, the CLE updates its model of currently salient discourse entities used for resolving intersentential references (Section 5). The LF implicit in the RQLF is then extracted to become the final output of the reference resolution and plausibility phases. To summarize, we can view a QLF as giving a *semantic analysis* for an utterance, an RQLF as a representation of its *interpretation* in context, and an LF as a representation of its *truth conditions*. The process of QLF resolution can be viewed schematically as follows:

QLF—SCOPING AND REFERENCE→
RQLFs—CONSTRAINTS AND PLAUSIBILITY→*LF*.

A general property of RQLFs is that they are formed by adding information to the QLFs from which they were derived; intuitively, this corresponds to fleshing out the QLFs in context. Thus referents are shown in-place in an RQLF, that is, the variables for resolved a_terms and definite descriptions are unified with their referents, and resolved a_form relations are instantiated in the a_form restriction. q_terms also remain in the RQLF, so the information about quantified noun phrases from the original (unscoped) QLF is preserved. This does mean that an RQLF can contain redundancies, but these will be removed when the final LF is extracted from it (though, for brevity, we will usually not show q_terms in scoped expressions). The RQLF interpretation leading to the LF shown above is

```
quant(exists, X,
  a_form(⟨t=pred,p=genit⟩,
    friend_of,
    [and,[friend,X],[friend_of,john,X]],
  [meet,
    a_term(⟨t=ref,p=pro,l=she,n=sing,a=⟨⟩⟩,
      mary,
      [female,mary]),
    q_term(⟨t=quant,n=sing,l=a⟩,
      X,
      a_form(...friend_of...))]].
```

In addition to the CLE procedures for proposing referents, checking constraints, and updating the salience model, the CLE provides an interface in terms of a set of application definable procedures (Section 5.2) for performing these functions. This interface allows the context imposed by a natural language processing application to influence reference resolution and plausibility checking, while keeping the CLE itself application independent.

3 Reference Resolution Rules

The motivation for having a set of rules for reference resolution is that this allows the basic resolution algorithm to be independent of the details of how anaphoric expressions are categorized and how different approaches to their resolution are ordered. This is in contrast to approaches (e.g., Hobbs 1976; Carter 1987) in which these details are encoded in the algorithm, leading to different success rates for different algorithms depending on the type of text

(Walker 1989 compares the performance of two such algorithms). Our approach should make it possible to write different sets of resolution rules for different natural languages, or sublanguages, without changing the mechanism applying these rules.

In the CLE, an ordered set of "reference resolution rules" determines the order in which proposed resolved LFs are derived from a QLF. Unresolved QLF expressions can be nested, so before the rules are applied to an unresolved expression, they are applied recursively to its restriction. Each rule specifies a "resolution method" that can be used to suggest a referent for an unresolved expression that matches a QLF expression specified in the rule:

refrule(*QLF_expression,Method*).

The resolution method corresponds to a relation between QLF expressions, contexts, and referents. In the CLE implementation, a method is the name of a Prolog predicate that takes an unresolved expression and returns candidate referents in a preference order by backtracking, so the method may succeed zero, one, or several times. The contextual information handed to the resolution method is in the form of an ordered list of discourse entities (Section 5.1) together with the whole QLF for the sentence being processed.

Resolution rules are ordered, with each rule being tried in this order against each unresolved expression in the sentence QLF. The QLF expression appearing in a rule is a term that can unify with either an a_term, a q_term or an a_form. A rule is taken to be applicable to the resolution of a QLF-expression if the expression unifies with the left hand side of the rule. In general, it is the category in unresolved QLF expressions that plays the major role in selecting resolution methods to be applied. The information represented in these categories is in principle open-ended in the same sense that holds for categories appearing in rules in unification grammar formalisms.

As well as the main reference resolution rules just described, the resolution process makes use of lexical declarations specifying, for example, the predicate corresponding to relational nouns or the quantifier cardinality predicate corresponding to determiners:

reflex(*WordSense,Category,Predicate*).

Examples of these "lexical reference declarations" will be given in Section 4.

The implementation of our QLF resolution model in the CLE uses a set of rules covering a range of reference phenomena that occur in English, including pronouns (Sections 4.1, 4.2, 4.3, 4.8), definite descriptions (Sections 4.4 and 4.9), quantifiers and collective readings (Sections 4.5 and 4.6), one-anaphora (Section 4.14), and implicit relations (Sections 4.11 and 4.12). The set of CLE resolution rules can, of course, be extended or changed, and we hope to develop a theory of ellipsis resolution based on this framework.

The possibilities available for resolving QLF expressions are governed by the set of reference rules and the context from which they propose referents. Choosing between these possibilities to arrive at a final LF depends on the constraints applied during the plausibility phase and on heuristic factors such as rule ordering and the salience weights associated with discourse entities. We will not be discussing these heuristic factors much, but instead will focus primarily on illustrating how a wide range of semantic and pragmatic phenomena can be viewed as QLF resolution (Section 4), and secondarily on the application of constraints on resolution to resulting RQLF interpretations (Section 6).

One way of viewing a QLF expression is as a conditional interpretation in the sense of Pollack and Pereira (1988). To take the simplest case, we can regard an a_term representing a pronoun in a QLF as an assumption that the variable for the a_term can be resolved by a matching reference rule. Our framework for QLF resolution can then be seen as a modular approach to representing and discharging such assumptions in a way that does not require their interactions, or the constraints on resolution, to be specified in the semantic interpretation rules of the system. It may be a consequence of the resulting simplicity of description that we have been able to cover a wide range of English constructions in the CLE implementation. We now turn to discussing these particular cases.

4 Referring Expressions

Various anaphoric phenomena handled by the CLE are considered in separate sections for expository purposes, but this does not correspond to a disjoint classification; bound variable anaphora, for example, overlap with both reflexive and nonreflexive pronouns. The particular features in the reference categories in this section are t, specifying the *type* of anaphoric expression (e.g. ref for noun phrase reference); p, for the *phrase* type (e.g. pro for pronoun); l, for *lexical* information; n, for *number;* and a, for specifying intra-sentential *antecedents.*

4.1 Pronouns

The QLF representation of a pronoun is an a_term. For example, the translation of *him* in *Mary expected him to introduce himself* is as follows:

a_term(⟨t=ref,p=pro,l=him,n=sing,a=⟨mary⟩⟩,X,[male,X])

The reference category states that the a_term is for a nonreflexive pronoun, with surface form him, expecting a singular referent, and that mary is a possible referent from within the sentence. The logical form restriction on the variable X constrains the referent to be male in this case. In the case of *there* and *then* it ensures that the referent is a place or time, respectively, this information being specified in the sense entries for the pronoun.

Inter-sentential reference possibilities are not included on the list of antecedents, since this list is built by the sentence-level semantic analysis rules. The list contains indices to the denotations of other noun phrases in the sentence. In our current semantic rules, the list is built up by unification by accumulating it in a pair of features that are "threaded" through the daughters of each constituent in a manner similar to the threading of long distance dependency gaps (Pereira 1981). Consequently, a pronoun cannot co-refer with a nonpronomial noun phrase following it in the order in which antecedents are threaded, unless this co-reference is made through an inter-sentential reference of the pronoun.

The threading order can be varied for different semantic rules, but tends to correspond to a recursive traversal of the nodes in the syntax tree in the order: mother, left-daughter, right-daughter. For semantic analysis rules covering topicalization, however, indices of noun phrases in the preposed constituent are threaded as though there was no topicalization, allowing a treatment of a common type of cataphoric reference. For example, in both of the sentences:

Mary heard a dog behind her
Behind her, Mary heard a dog

mary is on the antecedents list for *her* because in the second case *Behind her* is analyzed as preposed.

Several CLE reference rules cover pronouns, including the following:

```
refrule(a_term(⟨t=ref,p=pro,l=i⟩,_,_),
        hearer_speaker_ref)

refrule(a_term(⟨t=ref,p=pro⟩,_,_),
        intra_sentential)

refrule(a_term(⟨t=ref,p=pro⟩,_,_),
        inter-sentential).
```

The ordering of the rules shown above means that precedence is given to the method hearer_speaker_ref when the pronoun is "I," and that intra-sentential pronoun referents (taken from antecedents lists) are proposed before inter-sentential ones (discourse entities from the context model).

4.2 REFLEXIVE PRONOUNS

Reflexive pronouns are also translated into a_terms in QLF, the only difference being that the phrase type feature is refl instead of pro, the main reference rule being

```
refrule(a_term(⟨t=ref,p=refl⟩,_,_),
        intra_sentential).
```

Again taking the sentence *Mary expected him to introduce himself*, the a_term for *himself* is

```
a_term(⟨t=ref,p=refl,l=him,n=sing,a=⟨-X,mary⟩⟩,Y,[male,Y]).
```

The antecedents list for this reflexive pronoun includes mary as well as the variable from the other pronoun a_term; i.e. the variable that will eventually be unified with the referent of *him*.

The QLF level of representation does not take account of the linguistic constraints on co-reference for reflexive and nonreflexive pronouns. Instead, these constraints are handled separately (Section 6), so the antecedent lists for reflexive and nonreflexive pronouns are the same. Another possible approach is for the semantic rules to maintain two lists, one for reflexive and one for nonreflexive pronouns, but there are disadvantages to this: it adds considerable complexity to the semantic analysis rules, it is difficult to implement by unification (since it can involve moving items between the two lists), and it is also arguable that the constraints on reflexives are not purely linguistic (Section 6).

4.3 BOUND VARIABLE ANAPHORA

In the reference category of the example a_term just given in the previous section, we saw that the antecedent list for a pronoun may contain variables. Resolving one pronoun to the variable corresponding to another one ensures that they co-refer. In the case of *Mary expected him to introduce himself*, this means that both a_terms will be replaced by the same constant from outside the sentence. Bound variable anaphora, on the other hand, involves replacement of the a_term by a variable bound by a logical form quantifier (Partee 1978). A simple example is *Every bishop admires himself*, which has the following scoped QLF:

```
quant(⟨t=quant,p=det,l=every,n=sing⟩,X,
  [bishop,X],
  [admire,X,
    a_term (⟨t=ref,p=refl,l=him,n=sing,a=⟨-X⟩⟩,Y,
    male,Y])]).
```

In the resolved LF, the a_term will have been replaced by the variable X giving [admire, X, X] as the scope of the quantification over X. Further examples of resolving pronouns to variables are given in Sections 4.4, 4.10, and 4.13.

In the antecedents list, the item −X indicates that the variable X comes from a singular noun phrase antecedent (it would have been +X for a plural antecedent). This information is included in the reference category of the a_term rather than in its restriction, because it is relevant, for intra-sentential reference, to syntactic agreement as opposed to constraining what the logical form variable can be bound to. This can be seen by comparing the sentences *Every bishop admires himself* and *All bishops admire themselves*. In both cases the logical variable ranges over individual bishops, while there is number agreement between the the pronoun and the singular and plural subject noun phrases respectively.[2]

4.4 DEFINITE DESCRIPTIONS

Definite descriptions are represented as q_terms in QLF, and as such are scoped by the CLE quantifier scoping mechanism in the same way as other quantified noun phrases. The scoping of definite descriptions is motivated by the possibility of their resolution to quantifiers (Section

4.7). The scoped QLF and unscoped QLFs for *the small dog slept* are therefore, respectively:

```
[sleep,
  q_term(⟨t=ref,p=def,l=the,n=sing,a=⟨⟩⟩,
      X,
      [and,[dog,X],[small,X]])].

quant(⟨t=ref,p=def,l=the,n=sing,a=⟨⟩⟩,
    X,
    [and,[dog,X],[small,X]],
    [sleep,X]).
```

If the referent of the definite description here is fido, say, then the resolution of the description is effected by replacing the above quant formula by its body after replacing occurrences of the variable X in the body with fido. In simple cases such as this, the same effect could have been achieved by representing the definite description by an a_term to be replaced by fido in the resolved LF.

The rules for handling such referential readings of definite descriptions include methods for proposing referents from the external application context as well as the CLE context model:

```
refrule(q_term(⟨t=ref,p=def,l=the,n=sing⟩,_,_),
        salient_satisfying_restriction)
refrule(q-term(⟨t=ref,p=def,l=the,n=sing⟩,_,_),
        reference_candidate_applic).
```

In the second rule, the resolution method is an application dependent method for singular definite descriptions. Typically, such a method will attempt to check that candidates it proposes satisfy the restriction (though not necessarily the body) of the definite quantifier by simple database lookup or inference from a knowledge base. For example, it might propose a salient entity john as the referent for *the sailor that Mary loves* if it managed to prove that the LF

```
[and,[sailor,john],[loves,mary,john]]
```

holds in the knowledge base. We will return to definite descriptions when discussing plurals and attributive readings in the following sections.

4.5 DETERMINERS AND QUANTIFIERS

Resolving QLFs covers phenomena that are often handled by the semantic analysis and quantifier scoping components of language processing systems (including earlier versions of the CLE). Some of these, such as the correspondence between determiners and quantifiers and the derivation of collective and attributive readings of noun phrases, are handled as "quantifier resolution." The basic idea will be given in this section; other cases are discussed in Sections 4.6 and 4.7.

As mentioned earlier, the QLF representation of a quantified noun phrase or definite description is a q_term for which the scoping algorithm proposes possible scopes (body expressions). This process takes into account scoping preferences, in particular those associated with the determiner appearing in the category of the q_term. In the resulting scoped QLF, there will be a quant formula for each q_term as follows:

```
q_term(Category,Variable,Restriction)
quant(Category,Variable,Restriction,Body)
```

In earlier versions of the CLE scoping algorithm, the q_termwas replaced with its variable. Currently this information is only removed from the body when a final LF is extracted from an RQLF, so that it can be taken into account by the constraints applied to RQLF interpretations.

After scoping, reference resolution fixes the interpretation of determiner categories as quantifiers, yielding

```
quant (Quantifier, Variable,Restriction,Body).
```

The mapping of determiner categories to quantifiers depends on lexical reference declarations. For example, the following two declarations allow resolution of the determiners in *some dog* and *all dogs* to exists and forall, respectively:

```
reflex(some,⟨t=quant,p=det,n=sing,l=some⟩,exists)
reflex(all,⟨t=quant,p=det,l=all⟩,forall)
```

Quantifiers (in the distributive case) are taken to be predicates on two cardinalities, the number of entities, R, satisfying the restriction, and the number of entities, I, satisfying the conjunction of the restriction and the body. Abbreviations for these predicates are used for common cases, so, for example, forall abbreviates $\lambda r \lambda i.r = i$, which in our notation is R ^ I ^ [eq,R,I]. Other examples of the declarations for distributive quantifiers are

```
reflex(several,⟨t=quant,p=det,n=plur,lex=several⟩,
  R^I^[geq,I,3])⟩
reflex(N,⟨t=quant,p=det,n=number(N),lex=N⟩,R^I^[geq,I,N])⟩
reflex(bare,⟨t=quant,n=plur,l=bare⟩,forall)
reflex(bare,⟨t=quant,n=plur,l=bare⟩,R^I^[geq,I,2])
reflex(bare,⟨t=quant,n=plur,l=bare⟩,most).
```

The first of these is for numeral determiners, so this declaration treats *3* as *at least 3* (geq is the ≥ relation).[3] The last three declarations allow "empty determiners" for bare plurals like *dogs* to be resolved as a universal, as in *dogs are animals,* or some number greater than one, as in *she could hear dogs barking,* or most as in *dogs eat meat.*

4.6 COLLECTIVE READINGS

Our treatment of quantifier resolution covers distributive/collective distinctions. In the LF language, quantifiers can be of the form set(Q), where again Q is a predicate on two numbers. For example, the LF representation of the collective reading of *Two boys carried John* is[4]

```
quant(set(R^I^[geq,I,2]),
  B,
  [boy,B],
  quant(exists,E,[event,E],[carry,E,B,john]))
```

The interpretation of a collective set quantification is as follows. Set quantification variables range over sets of individuals. For such a quantificaticn to be true, Q must hold of the cardinality of the union of sets satisfying the restriction, and the cardinality of the maximal subset of this union that satisfies the body.[5]

The QLF for the sentence, before scoping, is

```
[carry,
  q_term(⟨t=quant,n=sing,l=ex⟩,E,[event,E]),
  q_term(⟨t=quant,p=det,n=number(2),l=2⟩,B,[boy,B]),
  john],
```

in which the determiner is resolved to the set quantifier according to the first of the following declarations:

```
reflex(N,⟨t=quant,p=det,n=number(N),l=N⟩,set(R^I^[geq,I,N]))
reflex(several,⟨t=quant,p=det,n=plur,l=several⟩,
  set(R^I^[geq,I,3])).
```

The interaction of scoping alternatives with the collective/distributive readings allowed by the declarations matching several and 2 gives several possible LFs for a sentence like *2 girls met several boys*, of which the following involves two (or more) girls, a meeting event for each girl, each event involving three or more boys:

```
quant(R^I^[geq,I,2],G,[girl,G],
  quant(exists,E,[event,E],
    quant(set(R^I^[geq,I,3]),B,[boy,B],
      [meet,E,G,B]))).
```

4.7 ATTRIBUTIVE DESCRIPTIONS

As well as the referential readings of definite descriptions discussed in Section 4.4, attributive readings expressed as quantified LF expressions are also proposed during QLF resolution. Since definite descriptions are represented in QLF as q_terms, the analysis of *Mary met the minister from France* is

```
[meet,
  mary,
  q_term(⟨t=ref,p=def,l=the,n=sing,a=⟨mary⟩⟩,
    M,[and,[minister,M],[from,M,france]])].
```

After scoping, determiner resolution (Section 4.5) takes place according to a lexical declaration:

```
reflex(the,⟨t=ref,p=def,l=the,n=sing⟩,exists),
```

producing the following LF:

```
quant(exists,M,
  [and,[minister,M],[from,M,france]],
  [meet,mary,M]).
```

If uniqueness of the description referent is required, for example, if presuppositions are being derived from the sentence, a more appropriate lexical declaration might be one giving the quantifier predicate R^I^[eq,I,1] (exactly one).

In the case of parametrized definite descriptions, only attributive readings are allowed. An example of a description parameterized by a free variable can arise in the resolution of a sentence such as *Every dog buried the bone that it found* for which the QLF, shown here after scoping, is

```
quant(⟨t=quant,p=det,l=every⟩,
  X,[dog,X],
  quant(⟨t=ref,p=def,l=the,n=sing,a=⟨-X⟩⟩,Y,
    [and,[bone,Y],
      [find,
        a_term(⟨t=ref,p=pro,l=it,
                n=sing,a=⟨-Y,-X⟩⟩,
          W,[impersonal,W]),
        Y]],
    [bury,X,Y]).
```

One of the possibilities for resolving the pronoun *it* in this sentence is to treat it as a bound variable anaphor, i.e., to replace the a_term with X, the variable ranging over dogs. (This resolution is proposed because X is on the a_term antecedents list; see Section 4.1. Y is also proposed, but this possibility is eventually ruled out by sortal constraints). If this resolution is made, then the definite description variable, Y, cannot be resolved to a constant because the variable X is free in its restriction. This constraint is checked along with other binding constraints on resolution (Section 6). After resolution of the pronoun, determiner resolution takes place, giving

```
quant(forall,X,[dog,X],
  quant(exists,Y,
    [and,[bone,Y],[find,X,Y]],
    [bury,X,Y])).
```

4.8 PLURAL PRONOUNS

Plural pronouns that are bound variable anaphora are treated in the CLE in the same way as singular cases, but taking number agreement into account, as was illustrated in Section 4.3 above with the example *All bishops admire themselves*. This gives us a collective or distributive reading depending on whether the variable replacing the a_term is bound by a set quantifier or a normal quantifier (Section 4.6).

Plural pronouns, including plural reflexives, can also be resolved to collections (i.e. sets of entities) taken from the antecedents list in reference categories or from the context model (Section 5.1). These referents may be proposed by either intra-sentential or inter-sentential resolution methods, including application specific methods. At the QLF level, a plural pronoun such as *they* is represented as a q_term for quantifying over subsets S of a collection corresponding to an a_term:

```
q_term(⟨t=quant,l=all⟩,
  S,
  [subset,S,
    a_term(⟨t=ref,p=pro,l=they,n=plur⟩,
      X,[entity,X])]).
```

This QLF representation covers the possibilities of a distributive or collective interpretation depending on whether the q_term is resolved to a normal or set quantifier. In other words, the use of the two QLF constructs in the pronoun analysis factors out the distributivity and reference aspects of its interpretation.

As an example, the QLF representation for *John and Mary met after they graduated* is

```
[and,
  [meet,
    q_term(⟨t=quant,n=sing,l=ex⟩,E,[event,E]),
    {john,mary}],
      [after,E,
      [graduate,
    q_term(⟨t=quant,n=sing,l=ex⟩,F,[event,F]),
    q_term(⟨t=quant,l=all⟩,S,
      [subset,S,
        a_term(⟨t=ref,p=pro,l=they,n=plur,
              a=⟨{john,mary},mary,john⟩⟩,
          X,[entity,X])])]]].
```

In this example, the collection {john,mary} is the subject of the (collective) meeting event, and it also appears on the antecedents list for *they*. If the pronoun a_term is resolved to this collection and its q_term is resolved to the normal (distributive) forall, we get the following LF:

```
quant(exists,E,[event,E],
  [and,
    [meet,E,{john,mary}],
    [after,E,
      quant(forall,S,
      [subset,S,{john,mary}],
      quant(exists,F,[event,F],
        [graduate,F,S]))]]).
```

Since the variable S is interpreted distributively, the restriction will bind it to singleton subsets of {john,mary}, ensuring that there are graduation events for each of John and Mary.

4.9 PLURAL DEFINITE DESCRIPTIONS

Plural definite descriptions are handled in a parallel fashion to their singular counterparts, except that the restriction employs subset in a similar way to plural pronouns. The QLF representation for a noun phrase like *the three dogs* is thus

```
q_term(⟨t=quant,n=plur,l=all⟩,
  S,
  [subset,S,
    q_term(⟨t=ref,p=def,l=the,n=number(3)⟩,
      X,[dog,X])]).
```

If the lower q_term is resolved to an explicit set, this gives a referential reading. Otherwise the determiner can be resolved to a quantifier with the declaration:

```
reflex(the,⟨t=ref,p=def,n=number(N)⟩,set(R ^ I ^ [eq,I,N]))
```

giving an attributive reading. The distribution/collective distinction arises from the resolution of the quantifier for S to forall or set(forall), as explained in the case of plural pronouns (Section 4.8).

The formation of discourse entities for plural referents is a major problem for language processing systems. This is reflected in the CLE in that the current coverage of English plural reference is far from complete. In particular, none of the current built-in resolution methods in the CLE produces new collections from the set of entities present in the context model, which is required for handling discourses such as *John left; Bill left; they were angry*. However, the CLE external context interface (Section 5.2) allows collections, such as the set of answers to a database query, to be included in the model.

4.10 UNBOUND ANAPHORIC TERMS

When an argument position in a QLF predication must co-refer with a pronoun or definite description, 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*:

```
[want,
  a_term(⟨t=ref,p=pro,l=he,n=sing,a=⟨⟩⟩,X,[male,X]),
  [apply,
    y ^ [swim,Y],
    a_index(X)]].
```

(The unreduced lambda application comes from the infinitive; Section 6.4.) If the a_index variable is subsequently resolved to a quantified variable or a constant, then the a_index operator becomes redundant and is not included in the final LF. However, in special cases, such as the so-called "donkey sentences" (Geach 1962; Kamp 1981), an anaphoric term may be resolved to a quantified variable V outside the scope of the quantifier that binds V. An example is one reading of *Every farmer who owns a dog loves it*, in which the "unbound dependency" is indicated by retaining the a_index operator in the final structure:

```
quant(forall,X,
  [and,
    [farmer,X],
    quant(exists,Y,
      [dog,Y],
      [own,X,Y]),
  [love,X,a_index(Y)]]).
```

Such structures, which in some sense are not fully resolved LFs, cannot be given an interpretation using normal variable binding. However, an interpretation model similar to the one advocated by Groenendijk and Stokhof (1987) for their "dynamic logic" may be appropriate. The more immediate question of how we recognize that the out-of-scope reference is permitted is treated in our framework as a constraint on binding (Section 6).

4.11 IMPLICIT RELATIONS

English 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, R, introduced implicitly by a possessive phrase, holds between John and the house. Since possessive constructions are treated as definite descriptions, the scoped QLF for *John's house leaks* is as follows:

```
quant(⟨t=ref,p=def,...⟩,X,
  a_form(⟨t=pred,p=poss⟩,R,
    [and,[house,X],[R,john,X]]),
  [leak,X]).
```

The implicit relation R can then be determined by the reference resolver and instantiated, to owns or lives_in say, in the resolved LF. (The plausibility phase would then be expected to rule out incorrect candidates giving different relations in the two sentences *Where was John's car hired from?* and *John's car broke down three days after he bought it.*)

Relations implicit in (nonlexicalized) compound nouns, possessives, and *have* constructions can often only be determined from detailed knowledge of the domain of discourse (Alshawi 1987). Since this knowledge is not available to the CLE, the CLE depends on application-specific rules such as:

```
refrule(a_form(⟨t=pred,p=nn⟩,_,_),
        relation_candidate_applic)
```

for proposing resolutions for these constructions.

4.12 LEXICALIZED RELATIONS

These are treated as a special case of implicit relations and are applicable to relational nouns and lexicalized compound nouns with rules such as:

```
refrule(a_form(⟨t=pred,p=poss⟩,_,_),
        relational_sense)
refrule(a_form(⟨t=pred,p=nn⟩,_,_),
        lexicalized_compound).
```

At the QLF level, the representation of *John's mother* has an a_form that is exactly parallel to the one for *John's house*:

```
a_form(⟨t=pred,p=poss⟩,R,[and,[mother,X],[R,john,X]]).
```

Lexical reference declarations associated with noun senses in the CLE lexicon can specify a relation to be used to resolve the a_form analysis of a noun phrase with a relational noun head. The declarations give the phrase type (e.g. poss for possessive constructions and genit for the genitive *of* construction) for which resolution to the relational sense is applicable. Given such a declaration,

```
reflex(mother,⟨p=poss⟩,mother_of),
```

the resolution method relational_sense will propose a resolution of the above a_form in which R is instantiated to the predicate mother_of:

```
[and,[mother,X],[mother_of,john,X]].
```

Lexicalized compound nominals like *boat train* are treated similarly in that the method lexicalized_compound in the resolution rule given above proposes predicates specified in lexical declarations for the relevant noun senses.

4.13 POSSESSIVE PRONOUNS

Noun phrases with possessive pronouns acting as determiners are covered by the rules for pronouns, implicit relations, and definite descriptions. For example, the noun phrase *his mother* is analyzed in QLF using a combination of an a_term for the pronoun, a definite description quantifier for the whole noun phrase, and an a_form for the relation between the noun phrase referent and the pronoun. The scoped QLF representation for *John met his mother* is

```
quant(⟨t=ref,p=def,l=the,n=sing,a=⟨john⟩⟩,X,
  a_form(⟨t=pred,p=poss⟩,R,
    [and,[mother,X],
      [R,
        a_term(⟨t=ref,p=pro,l=he,n=sing,a=⟨john⟩⟩,Y,
          [male,Y]),
        X]]),
  [meet,john,X]).
```

Since resolution proceeds recursively, the a_term is resolved first, to john say, then the a_form to the relational noun predicate mother_of, and finally the definite description is resolved to janet, say, giving [meet,john,janet]. If no appropriate referent corresponding to John's mother is proposed, then an attributive reading will be produced by resolving the definite description to an existential quantification (Section 4.7) giving the following LF:

```
quant(exists,X,
  [and,[mother,X],[mother_of,john,X]],
  [meet,john,X]).
```

4.14 ONE-ANAPHORA

The so-called "one-anaphora" (Webber 1979), which can be complete noun phrases, as in *Mary knitted one,* or modified nominals, as in *John knitted a grey one,* are taken to refer to restrictions from a preceding noun phrase, for example X^[jumper,X] (i.e. $\lambda x.Jumper(x)$) from the sentence *Is it easy to knit a jumper?* It is therefore appropriate to represent such an anaphor in QLF with an a_form rather than an a_term, so that it matches the rule:

```
refrule(a_form(⟨t=pred,p=one⟩,_,_),
        previous_description).
```

Unlike the relation variables used in the representation of implicit relations, the variable bound by the a_form has the type of a one-place predicate. For example, the scoped

QLF for *John knitted a grey one* is

```
quant(⟨t=quant,p=det,l=a,n=sing⟩,X,
   [and,a_form(⟨t=pred,p=one⟩,P,[P,X]),
      [grey,X]],
   [knit,john,X]).
```

The resolution process involves replacing the body of the a_form with a resolved conjunct, in this case the one obtained by instantiating P to jumper, giving the following LF for the sentence:

```
quant(exists,X,
   [and,[jumper,X],[grey,X]],
   [knit,john,X]).
```

In the more general case, the resolution must be made to some but not all conjuncts of the preceding noun phrase restriction. For example, after a sentence like *John found a female cat,* the resolved restriction in *Mary wanted a male one* cannot include the contradictory conjunct [female,X]. The resolution method previous_description proposes each of the predicates X^[and,[cat,X],[female,X]], X^[cat,X], and X^[female,X] as resolution candidates, but only the second of these results in an LF satisfying sortal constraints.

The current CLE one-anaphora resolution method only proposes restrictions present in the context model that were derived from noun phrases in previous sentences. However, it would be possible to extend the treatment to handling intra-sentential cases by including another method that proposed restrictions of noun phrases whose variables are threaded as possible pronoun antecedents.

5 Context Model

5.1 Discourse Entities and Salience

The CLE maintains a simple model of context, consisting of a set of entities with associated salience weights. These entities, ordered by salience, are then passed to any reference resolution methods called; methods may or may not make use of this information when proposing referents. Entities in the salience model are represented as LF expressions. They are typically constants corresponding to individuals or collections, but they also include descriptions (predicates on one argument represented as a lambda abstraction) that are used in one-anaphora resolution. RQLFs corresponding to the interpretation of previous sentences are also included in the context model for the purpose of ellipsis resolution.

Updating the context model involves changing the set of items in the model and updating their salience weights. New items enter the model when an RQLF expression that is taken to be the correct interpretation of a sentence is derived. The addition of items to the context model in response to producing an RQLF interpretation is in many respects a simplification of Webber's approach to the introduction of discourse entities (Webber 1979), so we will not discuss it further here. One difference is that our treatment of bound variable anaphora operates directly on QLFs, so we avoid the need to generate temporary descriptions of entities for resolving such anaphora.

Updating salience weights is very simple in the CLE, weights being decremented as new sentences are processed. It is possible to extend the salience weight mechanism so that it can take into account the influence of syntactic, semantic, and nonlinguistic factors contributing to contextual salience as shown in Alshawi (1987), where it is also argued that notions of global and local focus (Grosz 1977; Sidner 1979) can be viewed as corresponding to salience thresholds.

5.2 External Context Interface

When the CLE is used as an interface to application software, the application becomes part of the context within which the interpretation of user utterances takes place. In order that the application may influence the interpretation process, the CLE provides facilities for interfacing the application to CLE processing during the pragmatics phases; that is, reference resolution and plausibility checking. This is done through sortal restrictions (Section 6.2) and through the use of "interface procedures."

CLE interface procedures relevant to reference resolution and plausibility are concerned with proposing referents for noun phrase anaphora and implicit relations and for checking the plausibility of interpretations in the application context. It is easy to add reference rules with application-specific methods. For example, a rule could be added matching the a_term for the deictic pronoun *here,* the method in the rule being a procedure that returned the coordinates of a pointing device. This approach allows simpler or more comprehensive interfaces to CLE pragmatics, according to the effort available for application development.

The CLE external context interface also allows applications to send back constants, such as answers or sets of answers to database queries, for inclusion in the model after logical form evaluation has taken place. This makes it possible for the user to refer to such entities with anaphoric expressions in later exchanges. Sortal information for the new constants can also be provided through this interface.

6 Constraints on Resolved LFs

6.1 Applying the Constraints

Resolved logical forms proposed by the reference resolution component are passed to the plausibility checking component. The main role of this component is to apply pragmatic constraints on plausibility in the form of sortal restrictions, and any application-specific constraints defined through the external context interface. Linguistic constraints on reference are also applied during this phase, because they also act as filters on the output of the reference resolution component.

Linguistically motivated preferences, encoded implicitly in the CLE by ordering lexical entries and rules, and explicitly as quantifier scoping rules, are reflected in the order in which logical forms are passed to the plausibility component. The normal mode of operation of the CLE is therefore to assume that the first proposed logical form to pass the plausibility constraints is the intended interpretation. In future developments of our model, we hope that the plausibility component will produce numerical (e.g. probabilistic) estimates of plausibility. In the meantime, the interface between the plausibility component and an application system is such that it is possible for the application to compare the first five candidate interpretations, say, against each other, selecting one of them.

As explained in Section 2.2, the RQLF passed to the plausibility phase includes the set of resolutions made, enabling the plausibility of these resolutions to be evaluated, rather than simply the plausibility of the corresponding final LF. This is particularly relevant to applying the linguistic constraints on pronoun reference, but it also gives the application a chance to veto resolutions proposed by the CLE. In general, the reason for passing the RQLF to the plausibility component is that plausibility, as we view it, is really a function of the interpretation of an utterance in context rather than simply a function of its truth conditions, as illustrated in the rest of this section.

6.2 SORTAL CONSTRAINTS

The sortal restriction mechanism used in the CLE associates sorts with objects, properties, and relations according to a declaratively specified sort hierarchy (Alshawi et al. 1988) that can be extended or redefined in order to reflect constraints that are specific to a particular domain of discourse. Sorts are coded into terms that unify if and only if they do not violate the restrictions (cf. Mellish 1988), giving an efficient way of applying plausibility constraints on domain predicates. The coding technique used is by now well known, so we will not discuss it further. In the CLE, checking the sortal constraints involves unifying logical form variables with sorted variables of the form V;S, where S is a coded sort term.

It should be possible to extend the sortal restriction mechanism to cover constraints on whether predicates corresponding to word senses take individuals or collections as their arguments in order to rule out distributive or collective readings proposed during QLF resolution, but this has not yet been tried in the context of the CLE.

As with the other constraints, sortal constraints are applied to the RQLF rather than the final LF. This is necessary because QLF subexpressions that do not appear in the final LF, such as expressions for relative clauses in referential definite descriptions, need to be taken into account, and also because reference resolution introduces new predicates and arguments as referents. Applying the sortal check to the RQLF ensures that the constraints arising from the QLF expressions and the resolutions are checked for compatibility.

6.3 BINDING CONSTRAINTS

Variable binding constraints on resolved logical forms have already been mentioned in the discussion of definite descriptions and unbound anaphoric terms. Three constraints on variable binding are applied:

1. All variables, other than a_index variables, must occur within the scope of their binders.
2. A definite description whose restriction includes a variable bound outside the restriction must be resolved as a quantified expression.
3. An a_index variable must be "accessible" from a QLF q_term for the variable.

The first of these is the familiar constraint that disallows free variables (Montague 1973; May 1985) and covers variables bound by lambda abstraction as well as quantifiers. The second constraint was discussed with an example in Section 4.7. Clearly this constraint must be applied to an RQLF before its conversion into an LF, as the restriction of the definite description may not appear in the final LF representation.

The third constraint is the one relevant to unbound anaphoric terms (Section 4.10). Various characterizations of the required accessibility conditions relevant to this constraint have been studied, especially in terms of subordination of discourse representation structures in DRT (Kamp 1981). The accessibility condition in the CLE is rather simplistic and is currently restricted to making variables from an indefinite noun phrase in the restriction of a quantification accessible from its body (or one in the antecedent of an implication accessible in its consequent). This predicts the following for the acceptability of resolving the pronoun in the examples below to the dog variable (assuming narrow scoping of this variable):

Every farmer who owns a dog loves it
*Every farmer who owns no dog loves it
*Every farmer who owns it loves a dog
If a farmer owns a dog then s/he loves its

6.4 CONFIGURATIONAL CONSTRAINTS

The configurational constraints we consider here are basically linguistic constraints on the co-reference of reflexive and nonreflexive pronouns with other noun phrases in a sentence. In the CLE, the constraints are expressed at the RQLF level in contrast to some approaches in the linguistics literature (Lasnik 1976; Reinhart 1983), where they are expressed at the level of syntax trees.

Some constraints are implicit in the way semantic analysis rules thread items on the antecedents list for pronouns (Section 4.1). In this section we will describe the constraint that applies to the distribution of reflexive and nonreflexive pronouns. A simplified version of the constraint is explained first: a reflexive pronoun whose a_term is an argument to a logical form predicate must co-refer with another argument of that predicate, whereas a nonreflexive pronoun cannot co-refer with such an argument. This rules out the starred sentences below, under the resolutions shown in parentheses, while accepting the others:

Mary shot her (Janet)
Mary shot her (Mary)*
Mary shot herself (Janet)*
Mary shot herself (Mary)
Every bishop admires him (John)
Every bishop admires him (that same bishop)*
John's father loves himself (John)*
Bill said that he (Bill) had shot him (John)
Bill said that he (John) had shot himself (John)
Bill said that he (John) had shot himself (Bill)*

Other possible resolutions of these sentences are also accepted or rejected by the constraint as expected. When applying the constraint, only the RQLF needs to be examined, since this will include the predicate-argument structure, the reference category for a_terms showing p=pro or p=refl, and the proposed resolutions. In the first and last of the above sentences, the constraint applies to the argument sets of the predicates shoot, and say and shoot respectively:

```
[shoot,
  mary,
  a_term(⟨t=ref,p=pro,l=her,n=sing,a=⟨mary⟩⟩,janet,
    [female,janet])]

[say,
  bill,
  [shoot,
    a_term(⟨t=ref,p=pro,l=he,n=sing,a=⟨bill⟩⟩,
      john,[male,john]),
    a_term(⟨t=ref,p=refl,l=him,
      n=sing,a=⟨ – john,bill⟩⟩,
      bill,[male,bill])]].
```

The constraint gives also the expected behavior for comparatives such as:

John is taller than him (Bill)
John is as tall as him (John)*

because it applies to the arguments of a predicate more used in the LF representation of comparative constructions (Alshawi and van Eijck 1989), the final LF for the first of the examples above being

```
[more,E^F^[tall_degree,E,F],john,bill].
```

The actual constraint that is applied is slightly more complicated than the one stated above because it takes into account (and was part of the motivation for) propositional arguments that are unreduced lambda applications. These are used in the representation of infinitival complements, so, for example, *John expected him (Bill) to shoot himself (Bill)* has the following resolved LF:

```
[expect,
  john,
  [apply,
    X^[shoot,X,bill],
    bill]].
```

The full constraint is as stated before except that it demands that when a predicate P has a lambda application argument

```
[P, ...,[apply,X^Body,A],...],
```

then the argument A for the application is treated as though it were also an argument of P, and furthermore, the constraint is enforced on the lambda body after performing the reduction with a resolved argument. Informally then, the full constraint says how the simpler one should be applied at both clause levels. It accepts or rejects resolutions as shown for the following sentences:

John expected to shoot himself (John)
John expected to shoot himself (Bill)*
John believed him (Bill) to be nice
John expected him (Bill) to shoot himself (Bill)
John expected him (John) to shoot himself (Bill)*
John expected him (Bill) to shoot himself (John)*
John expected him (Bill) to shoot him (Bill)*
John expected him (Bill) to shoot him (John)

Since the constraint is applied at the level of logical form, it can cover sentences with "picture nouns" to give

John saw a picture of himself (John)
John saw a friend of himself (John)*

The implementation requires declarations specifying that relations such as picture_of are "representational," and then treats the meaning of the noun phrase *a picture of himself* as though it were co-referential with *John* for the purposes of checking the reflexive constraint.

One weakness in the way the constraint on reflexives is currently applied in the CLE is that it is sensitive to whether prepositional phrase modifiers have been subcategorized for or whether they are optional. Optional arguments do not appear in the predication for the verb, but rather as additional predications involving the event variable. This means that *John spoke to himself,* for example, will be rejected by the constraint as implemented unless the *to* prepositional phrase is subcategorized for. Fortunately, for cases like *Mary wrapped a jumper round herself* there is a strong argument that the prepositional phrase is not optional: *Mary wrapped a jumper* means something quite different.

7 Further Research

We are currently developing our model for QLF resolution within the framework of a combined language processing and contextual reasoning system called CLARE, of which the CLE is the language component. This includes work on the treatment of ellipsis as QLF resolution. Some simple cases are already handled, such as the representation of an utterance consisting solely of a noun phrase as a_form (...,P,[P,*NP*]) and resolving it by substituting the noun phrase meaning into the RQLF for a preceding sentence. We hope to extend this treatment to comparatives and other elliptical expressions whose resolutions are partially governed by syntax. Another significant aspect of QLF resolution that we have not yet addressed is determining the temporal indices implicit in QLF tense and aspect operators.

Perhaps the most challenging problem in automatic language processing, that of simulating the role of plausibility in understanding, remains the most difficult part of QLF resolution. Our framework isolates this problem into the more concrete one of assigning plausibility ratings to RQLF interpretations. We will be looking at ways of exploiting inferential and probabilistic reasoning with domain knowledge in developing solutions to this problem.

Acknowledgments

The research reported in this paper was carried out as part of the CLE development effort to which the following people have also contributed: David Carter, Jan van Eijck, Bob Moore, Doug Moran, Barney Pell, Fernando Pereira, Steve Pulman, and Arnold Smith. Development of the CLE was funded by an Alvey grant and by members of the NATTIE consortium (British Aerospace, British Telecom, Hewlett Packard, ICL, Olivetti, Philips, Shell Research, and SRI International). Steve Pulman, David Carter, and Jan van Eijck provided helpful comments on drafts of the text.

References

Alshawi, H. 1987 *Memory and Context for Language Interpretation.* Cambridge University Press, Cambridge, England.

Alshawi, H. and van Eijck, J. 1989 Logical Forms in the Core Language Engine. *27th Annual Meeting of the Association for Computational Linguistics,* Vancouver, British Columbia.

Alshawi, H., Carter, D. M., van Eijck, J., Moore, R. C., Moran, D. B., Pereira, F. C. N., Pulman, S. G. and Smith, A. G. 1988 *Interim Report on the SRI Core Language Engine.* Technical Report CCSRC-5, SRI International, Cambridge Research Centre, Cambridge, England.

Carter, D. M. 1987 *Interpreting Anaphors in Natural Language Texts.* Ellis Horwood, Chichester, England.

Geach, P. 1962 *Reference and Generality.* Cornell University Press, Ithaca, NY.

Groenendijk, J. and Stokhof, M. 1987 Dynamic Predicate Logic. Faculteit der Wijsbegeerte, University of Amsterdam, Amsterdam, The Netherlands.

Grosz, B. J. 1977 The Representation and Use of Focus in Dialogue Understanding. (Ph.D. thesis), Technical Note No. 151, SRI International, Menlo Park, CA.

Hobbs, J. R. 1976 Pronoun Resolution. Technical Report 76-1, Department of Computer Science, City College, City University of New York, NY.

Kamp, H. 1981 A Theory of Truth and Semantic Representation. In: Groenendijk, J. A. G., Janssen, T. M. V., and Stokhof, M. B. J., eds., *Formal Methods in the Study of Language.* Mathematisch Centrum, Amsterdam, The Netherlands.

Lasnik, H. 1976 Remarks on Coreference. *Linguistic Analysis* 2:1–22.

McDonald, D. B. 1982 Understanding Noun Compounds. Ph.D. thesis, Carnegie–Mellon University, PA.

May, R. 1985 *Logical Form.* Linguistic Inquiry Monographs, MIT Press, Cambridge, MA.

Mellish, C. S. 1988 Implementing Systemic Classification by Unification. *Computational Linguistics* 14:40–51.

Montague, R. 1984 The Proper Treatment of Quantification in Ordinary English. In: Thomason, R., ed., *Formal Philosophy: Selected Papers of Richard Montague.* Yale University Press, New Haven, CT.

Moore, R. C. 1981 Problems in Logical Form. *19th Annual Meeting of the Association for Computational Linguistics.* 117–124.

Moran, D. B. 1988 Quantifier Scoping in the SRI Core Language Engine. *26th Annual Meeting of the Association for Computational Linguistics.* 33–40.

Partee, B. 1978 Bound Variables and Other Anaphors. In: Waltz, D., ed., *Proceedings of TINLAP-2.* University of Illinois-Urbana. 79–83.

Pereira, F. C. N. 1981 Extraposition Grammars. *Computational Linguistics* 7:243–256.

Pollack, M. E. and Pereira, F. C. N. 1988 An Integrated Framework for Semantic and Pragmatic Interpretation. *26th Annual Meeting of the Association for Computational Linguistics.* 75–86.

Reinhart, T. 1983 *Anaphora and Semantic Interpretation.* Croom Helm, London, U.K.

Scha, R. J. H. 1981 Distributive, Collective and Cumulative Quantification. In: Groenendijk, J. A. G., Janssen, T. M. V., and Stokhof, M. B. J., eds., *Formal Methods in the Study of Language.* Mathematisch Centrum, Amsterdam, The Netherlands.

Sidner, C. L. 1979 Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse. (Ph.D. thesis), Technical Report AI-TR-537, Artificial Intelligence Laboratory, MIT, Cambridge, MA.

Walker, M. A. 1989 Evaluating Discourse Processing Algorithms. *27th Annual Meeting of the Association for Computational Linguistics,* 251–261.

Webber, B. 1979 *A Formal Approach to Discourse Anaphora.* Garland, New York, NY.

Woods, W. A. 1978 Semantics and Quantification in Natural Language Question Answering. In: Yovits, M. C., ed., *Advances in Computers.* Academic Press, New York, NY.

Notes

1. CLE logical forms use square brackets for the application of predicates and operators to their arguments. For expository purposes, most formulae are shown simplified in this paper. In particular, we will often omit the tense operators and Davidsonian style event variables that appear in CLE sentence representations.
2. It would perhaps be more consistent with our overall framework not to represent this agreement information on antecedents lists, but to apply an agreement constraint together with the other constraints on pronomial reference to be discussed later.
3. Another possibility is to treat *3* as *exactly 3.* We take the contextual dependence of the plausibility of these interpretation possibilities to support the "quantifier resolution" approach.
4. In discussing collective and distributive readings, we explicitly show the quantification over event variables to make clear the number of events involved.
5. Strictly speaking, all LF variables are interpreted as ranging over sets, with those for normal quantifiers ranging over individuals which are singleton sets. (cf. Scha 1981).