

# MONOTONIC SEMANTIC INTERPRETATION\*

Hiyan Alshawi and Richard Crouch  
SRI International  
Cambridge Computer Science Research Centre  
23 Millers Yard  
Cambridge CB2 1RQ, U.K.  
hiyan@cam.sri.com rc@cam.sri.com

## ABSTRACT

Aspects of semantic interpretation, such as quantifier scoping and reference resolution, are often realised computationally by non-monotonic operations involving loss of information and destructive manipulation of semantic representations. The paper describes how monotonic reference resolution and scoping can be carried out using a revised Quasi Logical Form (QLF) representation. Semantics for QLF are presented in which the denotations of formulas are extended monotonically as QLF expressions are resolved.

## 1. INTRODUCTION

The monotonicity property of unification based grammar formalisms is perhaps the most important factor in their widespread use for grammatical description and parsing. Monotonicity guarantees that the grammatical analysis of a sentence can proceed incrementally by combining information from rules and lexical entries in a nondestructive way. By contrast, aspects of semantic interpretation, such as reference and quantifier scope resolution, are often realised by non-monotonic operations involving loss of information and destructive manipulation of semantic representations. A ‘two-level’ approach to semantic interpretation tends to result (Bronneberg *et al.* 1980), where an initial, underspecified representation is transformed into a separate, specified, representation.

The goal of the work described here is to provide a model for semantic interpretation that is fully monotonic in both linguistic and contextual aspects of interpretation, and which employs just one level of semantic representation — Quasi Logical Form (QLF). Contextual resolution of under-

specified QLF expressions involves the instantiation of QLF meta-variables. The semantics for the QLF formalism makes the denotation of a QLF formula a partial function to truth-values, with resolution leading to a monotonic extension of the denotation function. We believe that there are several advantages to the approach taken, including:

- Order independence of resolution operations
- Production of partial interpretations
- Simpler interactions between phenomena
- Reversibility for synthesis/generation

The QLF formalism is a development of Alshawi 1990. As before, underspecified QLFs are produced on the basis of a unification grammar. Previously, QLF resolution was only partially monotonic; full monotonicity required changes to the original QLF formalism and the resolution and scoping processes. These changes have been implemented in a further development of the Core Language Engine (Alshawi 1992), although we will ignore most implementation issues in the present paper.

The paper is organized as follows. Section 2 provides the syntax of the QLF language and Section 3 gives some illustrative examples of monotonic QLF resolution. Sections 4 and 5 present the semantics of the QLF formalism. Section 6 discusses the relationship between monotonic interpretation, Pereira’s categorial semantics (Pereira 1990), and context change approaches to semantics. Section 7 mentions some benefits of using QLF-like representations in implementing natural language systems.

## 2. SYNTAX OF MONOTONIC QLF

We give here a syntactic description of the QLF constructs for terms and formulas<sup>1</sup>.

<sup>1</sup>The notation we use in implementations is slightly different but equivalent to that presented here.

---

\*This work on the Core Language Engine was carried out under CLARE, a collaborative project involving BP Research, British Aerospace, British Telecom, Cambridge University, SRI International and the UK Defence Research Agency. The project is funded in part by the UK Department of Trade and Industry.

A QLF term must be one of the following

- a term variable: *X*, *Y*, ...
- a term index: +*i*, +*j*, ...
- a constant term: 7, *mary1*, ...
- an expressions of the form:  
`term(Idx,Cat,Restr,Quant,Reft)`

The term index, *Idx*, uniquely identifies the term expression. *Cat* is a list of feature-value equations, for example `<type=pro,num=sing,...>`. *Restr* is a first-order, one-place predicate. For a resolved term, *Quant* will be a generalized quantifier (a cardinality predicate holding of two properties) and *Reft*, the term's 'referent', will be a constant or term index. For an 'unresolved' term, *Quant* and *Reft* may be meta-variables (*\_x*, *\_y*, ...). (QLF terms may also be functional applications, though we will ignore these here).

A QLF formula must be one of the following

- the application of a predicate to arguments:  
`Predicate(Argument1,...,Argumentn)`
- an expression of the form:  
`form(Category,Restriction,Resolution)`
- a formula with scoping constraints:  
`Scope:Formula`

*Predicate* is a first or higher-order predicate, including the usual logical operators *and*, *not*, etc. An argument may be a term, a formula or a lambda abstract. Lambda abstracts take the form *Var*<sup>Body</sup> where *Body* is a formula or an abstract and *Var* is a variable ranging over individuals or relations. *Restriction* is a higher-order predicate. *Resolution* is a formula (the 'referent' of the form expression), or is a meta-variable if the form expression is unresolved. *Scope* is either a meta-variable when scoping information is underspecified or a (possibly empty) list of term indices e.g. [+*i*,+*j*] if term +*i* outscopes +*j*. The terms identified by the indices must occur within *Formula*.

The degree to which a QLF is unresolved corresponds approximately to the extent to which meta-variables (appearing above as *Quant*, *Reft*, *Scope*, and *Resolution*) are instantiated to the appropriate kind of object level expressions (though see Section 5 for an explicit characterization of unresolved QLFs and partial interpretations.)

### 3. EXAMPLE QLF RESOLUTIONS

Resolution of QLFs through the instantiation of meta-variables has been applied to a wide range

of phenomena. These include pronouns, definite descriptions, implicit or vague relations, ellipsis and temporal relations (see Alshawi 1990 for an account of some kinds of reference resolution in an earlier QLF formalism). For concreteness, we present a few illustrative examples of monotonic QLF resolution<sup>2</sup>. We do not attempt to describe the mechanism by which the resolutions are chosen.

It will become evident that the notation is closer to (the syntactic structure of) natural language than is the case for traditional logical formalisms. For example, terms usually correspond to noun phrases, with information about whether e.g. they are pronominal, quantified or proper names included in the term's category. This makes the QLF representation easier to read than it might seem at first, once its initial unfamiliarity is overcome.

**Quantification:** *Every boy met a tall girl* illustrates the representation of quantification. The basic QLF analysis might be (ignoring tense):

```
_s:meet(term(+b,<type=q,lex=every>,boy,_q,_x),
        term(+g,<type=q,lex=a>,
            Y^and(girl(Y),tall(Y)),_r,_y)).
```

A resolved structure could be obtained by instantiating the quantifier meta-variables *\_q* and *\_r* to *forall* and *exists*<sup>3</sup>, and the scoping meta-variable *\_s* to [+*b*,+*g*] for the '∀∃' reading:

```
[+b,+g]:
meet(term(+b,<type=q,lex=every>,
        boy,forall,+b),
      term(+g,<type=q,lex=a>,
          Y^and(girl(Y),tall(Y)),exists,+g)).
```

In a restriction-body notation for generalized quantifiers, the truth conditional content of this resolved expression corresponds to

```
forall(B,boy(B),
      exists(G,and(girl(G),tall(G)),
              meet(B,G))).
```

**Anaphora:** *Every boy claims he met her* illustrates the treatment of anaphora (in a context

<sup>2</sup>Although the QLF framework can support a variety of alternative semantic analyses for specific phenomena, to provide concrete illustrations one or other analysis needs to be chosen. In the following examples, it should be possible to separate particular analyses from the general points we wish to make about monotonic interpretation.

<sup>3</sup>The benefits of being able to resolve determiners to quantifiers are discussed in Alshawi 1990. For example, determiners like *some* (plural) could be resolved to collective or distributive quantifiers, *three* could be interpreted as meaning either 'exactly three' or 'at least three', and if need be, bare plurals like *dogs* could be variously interpreted as meaning 'some dogs', 'all dogs' or 'most dogs'.

where Mary is assumed to be salient)<sup>4</sup>

Unresolved:

```
_s1:claim(
  term(+b,<type=q,lex=every>,boy,_q1,_x),
  _s2:meet(term(+h1,<type=pro,lex=he>,
    male,_q2,_y),
    term(+h2,<type=pro,lex=her>,
    female,_q3,_z)))
```

Resolved:

```
[+b]:claim(
  term(+b,<type=q,lex=every>,
    boy,forall,+b),
  [+h1]:meet(term(+h1,<type=pro,lex=he>,
    male,exists,+b),
    term(+h2,<type=pro,lex=her>,
    female,exists,mary)))
```

The pronominal term for *her* is resolved so that it existentially quantifies over female objects identical to *mary*. The ‘bound variable’ pronoun *he* has a referent coindexed with its antecedent, *+b*. The scope of *+h2* is left unspecified, since exactly the same truth conditions arise if it is given wide or narrow scope with respect to *every boy* or *he*.

**Vague Relations:** An unresolved QLF expression representing the noun phrase *a woman on a bus* might be a term containing a form that arises from the the prepositional phrase modification:

```
term(+w,<lex=a,...>,
  X^and(woman(X),
    form(<type=prep,lex=on>,
      R^R(+w,term(+b,<lex=a,...>,
        bus,_q2,_b)),
      _f)),
  _q1,_w).
```

Informally, the form is resolved by applying its restriction,  $R^R(\dots)$  to an appropriate salient predicate, and instantiating the form’s meta-variable, *\_f*, with the result. In this case, the appropriate predicate might be *inside*, so that *\_f* is instantiated to

```
inside(+w,term(+b,<lex=a,...>,bus,_q2,_b)).
```

**Tense:** One way of treating tense is by means of a temporal relation form in the restriction of an event term. For *John slept* we might have:

```
_s:sleep(term(+e,<type=event>,
  E^form(<type=trel,tense=past>,
    R^and(event(E),R(E)),
    _t),
  _q1,_e),
  term(+j,<type=name>,
  J^name(J,'John'),_q2,_j)).
```

<sup>4</sup>Here we simplify the issues arising out of the semantics of intensional, sentential complement verbs like *claim*.

Since the tense on the temporal relation category is past, the resolution says that the event occurred before a particular speech time, *t7*:

```
[+e]:
sleep(
  term(+e,<type=event>,
    E^form(<type=trel,tense=past>,
      R^and(event(E),R(E)),
      and(event(E),precede(E,t7))),
    exists,+e),
  term(+j,<type=name>,
  J^name(J,'John'),exists,john1)).
```

The resolution  $\text{and}(\text{event}(E),\text{precede}(E,t7))$  is the result of applying the form’s restriction  $R^{\text{and}(\text{event}(E),R(E))}$  to a contextually derived predicate, in this case  $E1^{\text{precede}(E1,t7)}$ .

QLF is not committed to an event based treatment of tense. An alternative that has also been implemented is to treat the verbal predication *sleep(...)* as a temporal form, whose category specifies tense and aspect information.

**Ellipsis:** A more complex example, involving ellipsis and quantification, is provided by

*Each boy claimed he was clever, and so did John.*

A partially resolved QLF, but one in which the ellipsis is still unresolved, might be as follows (ignoring tense and event variables):

```
and(
  claim(term(+b,<lex=every>,
    boy,exists,+b),
    clever(term(+h,<lex=he>,
      male,exists,+b))),
  form(<type=vpellipsis>,
    P^P(term(+j,<type=name>,J^name(J,'John'),
      exists,john)),
    _e)).
```

This is a conjunction of the QLF for the antecedent clause (*Each boy claimed he was clever* under a bound pronoun reading) with a form expression for the verb phrase ellipsis. Solutions for instantiating the meta-variable *\_e* for the ellipsis are the result of applying a property *P1*, derived from the antecedent clause, to the term with index *+j*. The sentence has two readings: a sloppy reading where John claims that he is clever, and a strict one where John claims that each of the boys is clever. The choice between a strict or sloppy reading depends on how the term *he* is reinterpreted in the ellipsis resolution. Intuitively, strict identity involves referring to the same object as before, whereas sloppy identity involves referring to a relevantly similar object.

In QLF, a strict reading results from reinterpreting the ellipsis pronoun as co-indexed with the original, i.e. taking *P1* to be:

$X \text{ claim}(X, \text{clever}(+h))$ .

Constraints on legitimate scoping (Section 5) force  $+b$  and  $+h$  to take wide scope over both the antecedent and ellipsis. The sloppy reading results from re-indexing the ellipsis pronoun so that it has the same restriction and category as the original, but is resolved to  $+j$  and has a new index  $+h1$ . This corresponds to taking  $P1$  to be:

$X \text{ claim}(X, \text{clever}(\text{term}(+h1, \langle \text{lex}=\text{he} \rangle, \text{male}, \text{exists}, +j)))$ .

More generally, in Crouch and Alshawi 1992 we explore the claim that solutions to verb phrase ellipsis have the general form:

$P1 = X1 \dots Xi \text{ S}[X1/s1, \dots, Xi/si, \dots, tn/sn]$ .

That is,  $P1$  is formed out of an antecedent clause QLF  $S$  by abstracting over the ‘parallel elements’  $s1 \dots si$ , perhaps with some additional substitutions for terms  $si+1 \dots sn$  in  $S$  ( $E[a/b]$  is the expression  $E$  with  $a$  substituted for  $b$ ). This seems to be sufficient to cover the range of examples treated by Dalrymple, Shieber and Pereira (1991), but that is a specific linguistic claim about verb phrase ellipsis in English and not central to the present paper.

## 4. SEMANTICS FOR QLF

In this section we outline the semantics of the QLF language in a way that is as close as possible to classical approaches that provide the semantics in terms of a function from models to truth values. The main difference is that denotation functions will be partial functions for some unresolved QLF formulas, reflecting the intuition that these are ‘partial interpretations’. The denotation of a QLF expression will be extended monotonically as it is further resolved, a fully resolved formula receiving a total function as its denotation. The semantics is not intended to describe the resolution process.

Before giving evaluation rules for the QLF language, we first present a simplified version of the semantics for fully instantiated QLF expressions. This is for expository purposes only; the full QLF semantics does not depend on the simplified version.

### 4.1 SIMPLIFIED SEMANTICS

We will use the notation  $[[E]]m$  for the truth value of an expression  $E$  with respect to a model  $m$  (but will leave  $m$  implicit).  $m$  includes an interpretation function  $I$  for mapping constants and predicates into domain individuals and relations. Also left implicit is a function assigning values to variables, which is required for the evaluation of lambda abstracts as characteristic functions.

Constructs in the ‘standard’ predicate logic subset of QLF receive their semantics with the usual evaluation rules, for example:

- $[[P(a1, \dots, an)]] = 1$  iff  $I(a1) \dots I(an)$  are in the relation  $I(P)$ , and 0 otherwise.
- $[[\text{and}(F1, F2)]] = 1$  iff  $[[F1]] = 1$  and  $[[F2]] = 1$ , and 0 otherwise.

The evaluation rule for a formula  $F$  with a scoping variable instantiated to  $[I, J, \dots]$  and containing a term  $T = \text{term}(I, C, R, Q, A)$  is as follows:

- $[[[I, J, \dots] : F]] = 1$  iff  $[[Q(R', F')]] = 1$ , and 0 otherwise, where  
 $R'$  is  $X \text{ (and}(R(X), X=A)) [X/I]$ , and  
 $F'$  is  $X \text{ (}[J, \dots] : \text{and}(F, X=A)) [X/T, X/I]$

This evaluation rule states that a formula with a scoping constraint list may be evaluated by ‘discharging’ the term for the first index on the list with respect to a formula with a reduced scoping constraint. The rule discharges the term by abstracting over occurrences of the term and its index, and applying the generalized quantifier  $Q$  to the term’s restriction and the abstract derived from the formula. In Section 5 we will say more about the ramifications of adopting this type of quantifier evaluation rule. Note that this rule is also applicable to resolved terms such as pronouns for which  $Q$  has been resolved to **exists** and  $T$  is a constant or a scoped variable.

The denotation assigned to a resolved formula  $\text{form}(C, R, F')$  in which the resolution variable has been instantiated to a formula  $F'$  is simply:

- $[[\text{form}(C, R, F')]] = 1$  iff  $[[F']] = 1$ , and 0 otherwise.

### 4.2 QLF SEMANTICS

As mentioned earlier, the denotation of a formula  $F$  in the QLF language will be a possibly partial function ( $[[[ \dots ]]]$ ) from models to truth values. Again we use the notation  $[[F]]m$  for the truth value of a formula  $F$  with respect to a model  $m$  (explicit reference to a variable assignment function is again suppressed). For interpretation to be monotonic, we want  $[[G]]$  to be an extension of  $[[F]]$  whenever  $G$  is a more resolved version of  $F$ , and in particular for  $[[G]]$  to be total if  $G$  is fully resolved.

We will define  $[[[ \dots ]]]$  for QLFs in terms of a relation  $W$  between formulas, models and truth values. Evaluation rules will be given for  $W(F, m, v)$ , but since more than one rule may apply (or a rule may apply in more than one way),  $W$  will in general be a relation. The relationship between  $[[[ \dots ]]]$  and  $W$  for a formula  $F$  is as follows:

- $[[F]]m=1$  iff  $W(F,m,1)$  but not  $W(F,m,0)$ ;
- $[[F]]m=0$  iff  $W(F,m,0)$  but not  $W(F,m,1)$ ;
- $[[F]]m$  undefined iff  $W(F,m,1)$  and  $W(F,m,0)$ .

Henceforth we will leave the model argument  $m$  implicit. The evaluation rules for  $W$  will generally take the form

$$W(F,v) \text{ if } W(F',v)$$

where  $F'$  contains one fewer unresolved expression than  $F$  (so that it is possible for the process of rule application to terminate). The use of *if* rather than *iff* in these rules means that it is possible for rules producing more than one value  $v$  to apply and hence for  $[[F]]$  to be partial.

The model provides an interpretation function  $I$  mapping constants and predicates to individual and relations. We will also need to assume a relation  $S(C,H)$  (for 'salient') between QLF categories  $C$  and QLF expressions  $H$  standing for individuals, quantifiers, or predicates, but the precise nature of the salience relation and the way it changes during a discourse are not important for the evaluation rules for QLF given here. The intuitive motivation for  $S$  is that the category in an unresolved QLF expression restricts the set of possible referents for that expression.  $S$  is discussed further in Section 5. We are now in position to present the evaluation rules, which we number Q1, Q2, etc.

For standard connectives we have the obvious evaluation rules, for example,

$$Q1 \ W(\text{and}(F,G),1) \text{ if } W(F,1) \text{ and } W(G,1).$$

$$Q2 \ W(\text{and}(F,G),0) \text{ if } W(F,0) \text{ or } W(G,0).$$

$$Q3 \ W(\text{not}(F),1) \text{ if } W(F,0).$$

$$Q4 \ W(\text{not}(F),0) \text{ if } W(F,1).$$

Two rules applicable to a formula  $F$  containing a term with uninstantiated referent and quantifier meta-variables:

$$Q5 \ W(F,v) \text{ if } W(F[\text{exists}/\_q,A/\_r],v) \text{ and } W(R(A),1),$$

where:

$F$  is a formula containing the term  
 $T = \text{term}(I, C, R, \_q, \_r)$ , and  
 $A$  is term such that  $S(C,A)$ .

$$Q6 \ W(F,v) \text{ if } W(F[Q/\_q, I/\_r],v),$$

where:

$F$  is a formula containing the term  
 $T = \text{term}(I, C, R, \_q, \_r)$ , and  
 $Q$  is a quantifier such that  $S(C,Q)$ .

(The substitutions for the meta-variables  $\_r$  and  $\_q$  are to be read as part of the evaluation rule.)

A rule applicable to a formula  $F$  in which a (possibly unscoped) quantified term occurs:

$$Q7 \ W(F,v) \text{ if } W(Q(R',F'),v),$$

where:

$F$  is a formula containing the term  
 $T = \text{term}(I, C, R, Q, A)$ ,  
 $R'$  is  $X \wedge (\text{and}(R(X), X=A))[X/I]$ , and  
 $F'$  is  $X \wedge (\text{and}(F, X=A))[X/T, X/I]$ .

A rule applicable to a formula with an instantiated scoping constraint

$$Q8 \ W([I, J, \dots]:F,v) \text{ if } W(Q(R',F'),v),$$

where:

$F$  is a formula containing the term  
 $T = \text{term}(I, C, R, Q, A)$ ,  
 $R'$  is  $X \wedge (\text{and}(R(X), X=A))[X/I]$ , and  
 $F'$  is  $X \wedge ([J, \dots]:\text{and}(F, X=A))[X/T, X/I]$ .

We also need a trivial rule for a formula with an uninstantiated scoping constraint so that it reduces to application of other rules:

$$Q9 \ W(\_s:F,v) \text{ if } W(F,v).$$

Two rules are applicable to **form** expressions, corresponding to the cases of an uninstantiated or instantiated resolution meta-variable:

$$Q10 \ W(F,v) \text{ if } W(F[R(P)/\_r],v)$$

where:

$F$  is a formula **form**( $C, R, \_r$ )  
 $P$  is a predicate such that  $S(C,P)$ .

$$Q11 \ W(\text{form}(C, R, F'),v) \text{ if } W(F',v)$$

where  $F'$  is a QLF formula.

In a more complete description of the semantics we would also have to state that the evaluation rules provided give the only way of determining membership of the relation  $W$ .

## 5. NOTES ON THE SEMANTICS

**Monotonicity:** In this paper we are using monotonicity in two senses which (by design) turn out to be consistent. The first is a syntactic notion for QLF representations (instantiation rather than destructive manipulation), while the second is semantic:

1.  $F1$  is a more resolved version of  $F2$  if  $F1$  can be obtained by instantiating zero or more meta-variables in  $F2$ .
2.  $F1$  is a less partial interpretation than  $F2$  if  $[[F1]]$  is an extension of  $[[F2]]$ .

The claim of monotonicity for QLF is that for formulas  $F1$  and  $F2$ , if  $F1$  is a more resolved version of  $F2$  then  $F1$  is a less partial interpretation than  $F2$ .

**Scoping Constraints:** The quantification rules, (Q7) and (Q8), (i) select a term from a formula, (ii) discharge all occurrences of the term and its index in the formula and the term's restriction, replacing them by a variable, and (iii) apply the term's quantifier to the discharged restriction and formula. The difference between (Q7) and (Q8) is simply that the latter also discharges the head of the scoping list, in this case by removing it rather than by replacing it. (Keep in mind that the discharge and replacement operations take place at the level of the evaluation rules for QLF; they are not applied to the QLF expressions representing natural language meanings themselves).

As with Lewin's scoping algorithm, (Lewin 1990), there are no constraints built explicitly into the QLF semantics on where a quantification rule for a term may be applied, or indeed on the number of times it may be applied. However, several constraints arise out of (a) the *absence* of any semantic rules for evaluating isolated terms, term indices or scope lists, and (b) the requirement that a term be selected from a formula so that its quantifier is known.

The emergent conditions on legitimate scoping are

1. No term may be quantified-in more than once: The first application of the quantifier rule discharges the term. Subsequent applications of the rule lower down in the evaluation would fail to select an undischarged term.
2. When a term's index occurs in a scope list, the quantifier rule for the term must be applied at that point: It must be applied to discharge the head of the scope list, and by (1) above cannot additionally be applied anywhere else.
3. All occurrences of a term's index must occur within the scope of the application of the term's quantifier rule: The quantification rule will only discharge indices within the formula to which it is applied. Any occurrences of the index outside the formula will be undischarged, and hence unevaluable.
4. If a term R occurs within the restriction of a term H, and R is to be given wide scope over the restriction, then R must also be given wide scope over H: Otherwise, suppose H is given wide scope over R. Term H will first be discharged, replacing the term, and with it its restriction, in the formula to which the rule is applied. Then the quantification rule for R needs to be applied to the discharged formula, but the formula will not contain an occurrence of the term R, making the rule inapplicable.

The last two constraints have often been attributed to restrictions on free variables and vacu-

ous quantification. The attribution is problematic since open formulas and vacuously quantified formulas are both logically well defined, and without suspect appeal to the syntax of the logical formalism they cannot be ruled out as linguistically ill-formed. By contrast, QLF makes these violations semantically unevaluable.

**Unscoped Terms:** When a term's index is not mentioned in any scope list, the term may be quantified in at any point within the formula. For anaphoric terms whose referent has been resolved to some individual constant, it does matter where the quantification rule is applied; since the term existentially quantifies over things identical to a single object, the scope of the quantification is immaterial. It is thus convenient to leave anaphoric terms like this unscoped in QLF. Although this makes the QLF look (syntactically) as though it is not fully resolved, semantically it is. For other unscoped terms, alternative applications of the quantifier rule may well lead to distinct truth conditions, and in these cases the QLF is genuinely unresolved.

**Context Dependence:** Fully resolved QLFs are context-independent in the same sense that holds for closed formulas in traditional predicate logic (i.e. if the interpretation of the constant symbols in the language is fixed). Unresolved QLFs behave more like open formulas, and there is an analogy between assignments to unbound variables in predicate logic and possible resolutions of meta-variables admitted by the salience relation  $S$ .  $S(C,H)$  should be thought of as providing QLF expressions whose denotations are possible referents for unresolved expressions with category  $C$ . (It would have been possible to define  $S$  as a direct relation between categories and referents, but this complicates the statement of its role in resolution and in the semantic definitions.) We used  $S$  above in the definition of QLF semantics, but it is also central to NL processing: being able to compute  $S$  can clearly play an important role in the process of reference resolution during NL interpretation and in the process of building descriptions during NL synthesis. (The computational analogue of  $S$  was implemented as a collection of 'resolution rules' in Alshawi 1990.)

An important question is what to allow as possible expressions in the range of  $S$ . One observation is that as the range is widened, more NL resolution phenomena are covered. A rough summary is as follows:

- constants: intersentential pronouns
- predicate constants: compound nouns, prepositions

- quantifiers: vague determiners
- indices: bound variable, intrasentential pronouns
- predicates built from NP restrictions: one-anaphora
- predicates built from previous QLFs: intersentential ellipsis
- predicates built from current QLF: intrasentential ellipsis

## 6. RELATED APPROACHES

Viewed from a slightly different perspective, monotonic interpretation has a number of points of contact with Pereira's categorial semantics (Pereira 1990). Put briefly, in categorial semantics, semantic evaluation is represented as deduction in a functional calculus that derives the meanings of sentences from the meanings of their parts. Considerable emphasis is placed on the nature of these semantic derivations, as well as on the final results of the derivations (the 'logical forms' of sentences).

One significant advantage of this approach is that constraints on legitimate scoping emerge naturally from a consideration of permissible derivations of sentence meaning, rather than arising artificially from syntactic constraints imposed on logical forms. Derivations involving quantified terms first introduce an assumption that allows one to derive a simple term from a quantified term. This assumption is later discharged by the application of a quantifier. Conditions on the appropriate introduction and discharge of assumptions in natural deduction systems impose restrictions on the way that quantifiers may legitimately be applied. For example, a quantifier assumption may not be discharged if it depends on further assumptions that have not themselves been discharged. This prevents the occurrence of free variables in logical form, but without appeal to the syntax of logical form.

The discharge of terms and term indices when evaluating QLF closely parallels the discharge of quantifier assumptions in categorial semantics. Indeed, the terms and the indices are precisely the assumptions introduced by quantified expressions, and which need to be discharged. Furthermore, the different orders in which quantifier assumptions may be discharged in categorial derivation correspond to the choices that the quantifier rules permit for discharging quantified terms.

Where monotonic interpretation and categorial semantics part company is on the degree of explicitness with which semantic derivations are represented. In categorial semantics, derivation is a

background process that builds up logical forms, but is not explicitly represented in the semantic formalism. By contrast, the annotation of QLFs with scope lists provides an extra level of information about how the derivations proceed. In particular, they indicate which evaluation rules should be applied where.

QLF thus provides a (usually partial) specification of a semantic derivation, showing (a) what the initial 'premises' are (roughly, lexical meanings, although these too may only be partially specified), and (b) the rules by which the 'premises' are combined. QLF resolution amounts to further instantiating this specification. This view of QLF can be contrasted with Logical Form as it is normally understood, which represents the results of carrying out a semantic derivation.

The difference between specifying a derivation and carrying it out is what makes resolution order independent in monotonic interpretation. Making a resolution to QLF only specifies when and how an expression should be evaluated during semantic derivation; it does not carry out that part of the derivation. Where no distinction is drawn between making a resolution and carrying out the corresponding step of the derivation, the order of resolution can be important. Thus, for Dalrymple, Shieber and Pereira (1991), where this distinction is not drawn, the precise interleaving of scope and ellipsis resolution determines the interpretation of the sentence. In QLF, resolutions dictate the order in which various steps of the derivation are carried out, but the resolution order does not reflect the derivation order.

Distinguishing between specifying and performing a derivation also means that a monotonic treatment of ellipsis resolution does not need to resort to higher-order unification. Dalrymple, Shieber and Pereira use higher-order unification to 'unpick' the composition of constituent meanings obtained in the semantic derivation from the ellipsis antecedent. Some of these meanings are then put back together to produce a predicate that can be applied to the ellipsis arguments. Since monotonic resolution does not carry out the final composition of meanings, but merely sets out conditions on how it is to take place, there is no need to unpick composed meanings and put them back together again.

It is worth pointing out that monotonic interpretation is compatible with approaches to meaning as a transition between contexts or information states, and where the order in which transitions are made is significant (e.g. Veltman 1991). In such a framework, monotonic interpretation would amount to making decisions about which transitions to take when, but would not involve putting those decisions into action. The monotonicity in

monotonic interpretation thus refers to the way in which alternative derivations of sentence meanings may be chosen, but not to the semantic effects of those sentence meanings.

## 7. IMPLEMENTATION BENEFITS

A description of the language processing mechanisms to which we have applied the monotonic semantics model is beyond the scope of this paper. However, we believe that the QLF representation presented here brings significant advantages to implementing mechanisms for reference resolution, scoping, preference and generation.

**Reference and Scoping:** The order independence of resolution operations allows for a variety of control structures in implementing a resolution mechanism. We find it convenient to make a bottom up pass through QLFs making reference resolutions, followed by a stage of scoping resolution, and to iterate over this should any of the resolutions introduce further unresolved expressions.

The salience relation  $S$  can be implemented as procedures that search for properties, objects or indices in context. Scoping proceeds simply by the non-deterministic instantiation of scoping constraints, subject to the restrictions imposed on evaluable QLFs (Section 5), plus techniques for ignoring logically equivalent scopings, as for example described by Moran (1988).

**Preference and Disambiguation:** A resolved QLF preserves all the information in the original unresolved QLF, and also records the correspondence between resolved and unresolved expressions. This makes it possible to define preference metrics that can be used for ranking alternative interpretations independently of the search strategies used to derive them. For example, in the case of scoping, these metrics can combine information about how far a quantifier was 'raised' with information about the surface form of its determiner. Preference ranking over alternative resolutions facilitates automatic disambiguation of input. Interactive disambiguation can make use of generation from resolved QLFs for confirmation by a user.

**Generation:** There is a strong connection between monotonicity and reversibility in language processing systems. Monotonicity of unification means that algorithms such as head-driven generation (Shieber *et al* 1990) can be applied to grammars developed for analysis. We use a variant of this algorithm for generating from QLFs, and the monotonicity of semantic interpretation means

that the grammar used for generating from unresolved QLFs (the normal 'output' of the grammar) can also be used for generation from resolved QLFs.

In parallel to the distinction between grammatical analysis (of NL into unresolved QLFs) and interpretation, we make the distinction between grammatical synthesis (of NL from QLFs) and *description*. Description is the process of deriving a QLF from which synthesis proceeds by taking a fact (e.g. a database assertion) as input. We hope to report on our approach to description elsewhere. However, one of the principles of QLF-based description is that while interpretation instantiates referent fields in underspecified QLFs, description involves instantiating category and restriction fields for QLFs in which referent fields are already instantiated. The preference metrics applied to rank alternative interpretations can be applied equally well to ranking resolved QLFs produced by a nondeterministic description process, so there is a sense in which the preference mechanism can also be made reversible.

## REFERENCES

- Alshawi, H. 1990. "Resolving Quasi Logical Forms". *Computational Linguistics* 16:133-144.
- Alshawi, H., ed. 1992 (in press). *The Core Language Engine*. Cambridge, Massachusetts: The MIT Press.
- Bronneberg, W.J.H.J., H.C. Bunt, S.P.J. Landsbergen, R.J.H. Scha, W.J. Schoenmakers and E.P.C. van Utteren. 1980. "The Question Answering System PHLIQA1". In L. Bolc (ed.), *Natural Language Question Answering Systems*. Macmillan.
- Crouch, R. and H. Alshawi. 1992. "Ellipsis and Distributivity in Monotonic Interpretation", Technical Report, SRI International, Cambridge, UK.
- Dalrymple, M., S. M. Shieber, and F. C. N. Pereira. 1991. "Ellipsis and Higher-Order Unification". *Linguistics and Philosophy*, 14:399-452.
- Lewin, I. 1990. "A Quantifier Scoping Algorithm without a Free Variable Constraint", Proceedings of COLING 1990.
- Moran, D. B. 1988. "Quantifier Scoping in the SRI Core Language Engine". Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics, 33-40.
- Pereira, F. C. N. 1990. "Categorial Semantics and Scoping", *Computational Linguistics* 16:1-10.
- Shieber, S. M., G. van Noord, F. C. N. Pereira, and R. C. Moore. 1990. "Semantic-Head-Driven Generation". *Computational Linguistics* 16:30-43.
- Veltman, F. 1990. "Defaults in Update Semantics", in H. Kamp (ed), *Conditionals, Defaults and Belief Revision*, DYANA deliverable R2.5.A.