Ellipsis and Quantification: A Substitutional Approach

Richard Crouch
SRI International, Cambridge Computer Science Research Centre
23 Millers Yard, Mill Lane,
Cambridge, CB2 1RQ, UK
rc@cam.sri.com

Abstract

The paper describes a substitutional approach to ellipsis resolution giving comparable results to (Dalrymple et al., 1991), but without the need for order-sensitive interleaving of quantifier scoping and ellipsis resolution. It is argued that the order-independence results from viewing semantic interpretation as building a description of a semantic composition, instead of the more common view of interpretation as actually performing the composition.

1 Introduction

Dalrymple, Shieber and Pereira (1991) (henceforth, DSP) give an equational treatment of ellipsis via higher-order unification which, amongst other things, provides an insightful analysis of the interactions between ellipsis and quantification. But it suffers a number of drawbacks, especially when viewed from a computational perspective.

First, the precise order in which quantifiers are scoped and ellipses resolved determines the final interpretation of elliptical sentences. It is hard to see how DSP's analysis could be implemented within a system employing a pipelined architecture that, say, separates quantifier scoping out from other reference resolution operations—this would seem to preclude the generation of some legitimate readings. Yet many systems, for good practical reasons, employ this kind of architecture.

Second, without additional constraints, DSP slightly overgenerate readings for sentences like

(1) John revised his paper before the teacher did, and so did Bill.

Kehler (1993a) has convincingly argued that this problem arises because DSP do not distinguish between merely co-referential and co-indexed (in his terminology, role-linked) expressions.

Third, though perhaps less importantly, higher-order unification going beyond second-order matching is required for resolving ellipses involving quantification. This increases the computational complexity of the ellipsis resolution task.

This paper presents a treatment of ellipsis which avoids these difficulties, while having essentially the same coverage as DSP. The treatment is easily implementable, and forms the basis of the ellipsis resolution component currently used within the Core Language Engine (Alshawi et al., 1992).

Ellipsis interpretations are represented as simple sets of substitutions on semantic representations of the antecedent. The substitutions can be built up in an order-independent way (i.e. before, after or during scoping), and without recourse to higher-order unification. The treatment is similar to the discourse copying analysis of (Kehler, 1993a), and to the substitutional treatment suggested by Kamp within Discourse Representation Theory, described in (Gawron and Peters, 1990). However, we extend the notion of strict and sloppy identity to deal with more than just pronouns. In doing so, we readily deal with phenomena like scope parallelism.

While the treatment of ellipsis is hopefully of some value in its own right, a more general conclusion can be drawn concerning the requirements for a *computational* theory of semantics. Briefly, the standard view within formal semantics, which DSP inherit, identifies semantic interpretation with composition: interpretation is the process of taking the meanings of various constituents and composing them together to form the meaning of the whole. This makes semantic interpretation a highly order-dependent affair; e.g. the order in which a functor is composed with its arguments can substantially affect the resulting meaning. This is reflected in the order-sensitive interleaving of scope and ellipsis resolution in DSP's account. In addition, composition is only sensitive to the *meanings* of its components. Typically there is a many-one mapping from compositions onto meanings. So, for example, whether two terms with identical meanings are merely co-referential or are co-indexed is the kind of information that may get lost: the difference amounts to two ways of composing the same meaning.

The alternative proposed here is to view semantic interpretation as a process of building a (possibly partial) ***description*** of the intended semantic composition; i.e. (partial) descriptions of what the meanings of various constituents are, and how they should be composed together.[1] While the order in which composition operations are performed can radically affect the outcome, the order in which descriptions are built up is unimportant. In the case of ellipsis, this extra layer of descriptive indirection permits an equational treatment of ellipsis that (i) is order-independent, (ii) can take account compositional distinctions that do not result in meaning differences, and also (iii) does not require the use of higher-order unification for dealing with quantifiers.

The paper is organised as follows. Section 2 describes the substitutional treatment of ellipsis by way of a few examples presented in a simplified version of Quasi Logical Form (QLF) (Alshawi and Crouch, 1992; Alshawi et al., 1992). Section 3 gives the semantics for the notation, and argues that QLF is best understood as providing descriptions of semantic compositions. Section 4 raises some open questions concerning the determination of parallelism between ellipsis and antecedent, and other issues. Section 5 concludes.

2 Ellipsis Substitutions

This section illustrates the substitutional treatment of ellipsis through a small number of examples. For presentation purposes we only sketch the intended semantics of the simplified QLF notation used, and a more detailed discussion is deferred until section 3.

2.1 Simple VP Ellipsis

A simple, uninteresting example to fix some notation:

(2) John slept. So did Mary

We represent the first sentence, ignoring tense, as a (resolved) QLF

(3) $[+j] : \text{sleep}(\ \mathtt{term}(+j,\ \exists,\ \lambda y.\text{name}(y, \text{'John'}),\ \lambda y.y = \text{j\_smith}))$

The noun phrase ***John*** gives rise to an existentially quantified term, uniquely identified by the index +j. The `term` expression has four arguments: an index, a determiner/quantifier, an explicit restriction, and an additional contextually derived restriction. In this case, the quantifier ranges over objects that are named 'John' and are further restricted to be identical to some (contextually salient) individual, denoted by j_smith. Prior to reference resolution, the contextual restriction on the term would be an uninstantiated meta-variable; resolution consists of instantiating meta-variables to contextually appropriate values. The scope of the term is indicated by the scope node [+j] : prefixing the formula sleep(`term`(+j,...)). Again, prior to resolution this scope node would be an uninstantiated meta-variable.

A generalized quantifier representation equivalent to the above is

(4) $\exists(\lambda y.(\text{name}(y, \text{'John'}) \wedge y = \text{j\_smith}),\ \lambda x.\text{sleep}(x))$

The index in the scope node means that to semantically evaluate the QLF, you get hold of the quantifier, restriction and contextual restriction of the corresponding term. This forms a (generalized) quantifier expression, whose body is obtained by discharging all occurrences of the term and it index to a variable, and abstracting over the variable. Terms and indices not dischargeable in this manner lead to uninterpretable QLFs (Alshawi and Crouch, 1992).

We represent the elliptical sentence, again abbreviated, as a (partially resolved) QLF:

(5) $?P(\ \mathtt{term}(+m,\ \exists,\ \lambda y.\text{name}(y, \text{'Mary'}),\ \lambda y.y = \text{m\_jones}))$

$?P$ is an unresolved meta-variable. To resolve the ellipsis, it needs to be instantiated to some contextually salient predicate.

Along similar lines to DSP, we can set up an equation to determine possible values for $?P$[2]:

(6) $?P(\mathtt{term}(+j,\ldots)) = [+j]\text{:sleep}(\mathtt{term}(+j,\ldots))$

That is, we are looking for a predicate that when applied to the subject term of the ellipsis antecedent returns the antecedent. The interpretation of the ellipsis is then given by applying this predicate to the subject of the ellipsis.

The equation (6) is solved by setting $?P$ to something that takes a term T as an argument and substitutes T for `term`(+j,...) and the index of T for +j throughout the ellipsis antecedent (the RHS of (6)):

(7) $?P = T\hat{}([+j]\text{:sleep}(\mathtt{term}(+j,\ldots))\ |\ \{\mathtt{term}(+j,\ldots)/T, +j/\text{idx\_of}(T)\})$

Here $T\hat{}(\ldots)$ is a form of abstraction; for now it will do no harm view it as a form of λ-abstraction, though this is not strictly accurate. The substitutions are represented using the notation $|\ \{old/new, \ldots\}$.

Applying this value for $?P$ in the ellipsis (5), we get

(8) $[+j]\text{:sleep}(\mathtt{term}(+j,\ldots))\ |\ \{\mathtt{term}(+j,\ldots)/\mathtt{term}(+m,\ldots), +j/+m\}$

Ellipsis resolution thus amounts to selecting an antecedent and determining a set of substitutions to apply to it. For reasons that will be explained shortly, it is important that resolution does not actually carry out the application of the substitutions.

[1] This is similar to Nerbonne's (1991) constraint-based semantics, except that he builds descriptions of logical forms, not semantic compositions.

[2] Terms shown abbreviated, i.e. `term`(+j,...) instead of `term`(+j, $\exists$, λy.name(y,'John'), $\lambda y.y$ = j_smith).

However, were we to do this in this particular case, where the antecedent (3) is fully resolved, we would successfully capture the intended interpretation of the ellipsis, namely:

(9) $[+m]$: sleep(**term**(+m, $\exists$, λy.name(y, 'Mary'), $\lambda y.y =$ m_jones))

Note that the substitutions are not applied in the conventional order; viz. first replace +j by +m throughout (3) and then replace **term**(+j,...) by **term**(+m,...). The first substitution would ensure that there was no **term**(+j,...) for the second substitution to replace. The order in which substitutions apply instead depends on the order in which the expressions occur when making a top down pass through (3), such as one would do when applying semantic evaluation rules to the formula.

Note also that the term index substitution applies to the scope node, so that [+j] : is replaced by [+m] :. This ensures that the term for Mary in the ellipsis gets a parallel scope to the term for John in the antecedent. Scope parallelism may not be significant where proper names are concerned, but is important when it comes to more obviously quantificational terms (section 2.3).

2.2 Evaluative Substitutions

The meaning of an ellipsis is composed in essentially the same way, and from the same components, as the meaning of its antecedent. However, some changes need to be made in order to accommodate new material introduced by the ellipsis. The substitutions specify what these changes are. In the example discussed above, the meaning of the ellipsis is built up in the same way as for the antecedent, except that whenever you encounter a term corresponding to 'John' or something dependent/co-indexed with it, you it is treated as though it were the term for 'Mary' or dependent/co-indexed with it.

This means that the substitutions act as directives controlling the way in which QLF expressions within their scope are evaluated. They are not syntactic operations on QLF expressions — they are part of the QLF object language.

The reason that substitutions are not 'applied' immediately upon ellipsis resolution is as follows. At the time of deciding on the ellipsis substitutions, the precise composition of the antecedent may not yet have been determined. (For instance the scopes of quantifiers or the contextual restrictions on pronouns in the antecedent may not have been resolved; this will correspond to the presence of uninstantiated meta-variables in the antecedent QLF.) The ellipsis should follow, modulo the substitutions, the same composition as the antecedent, ***whatever that composition is eventually determined to be.*** It makes no sense to apply the substitutions before the antecedent is fully resolved, though it does make sense to decide what the appropriate substitutions should be.

In practical terms what this amounts to is exploiting re-entrancy in QLFs. The elliptical QLF will contain a predicate formed from the antecedent QLF plus substitutions. Any uninstantiated meta-variables in the antecedent are thus re-entrant in the ellipsis. Consequently, any further resolutions to the antecedent are automatically imposed on the ellipsis. This would not be the case if the substitutions were treated as syntactic operations on QLF to be applied immediately: some re-entrant meta-variables would be substituted out of the ellipsis, and those remaining would not be subject to the substitutions (which would have already been applied) when they were eventually instantiated.

2.3 Scope Parallelism

It was noted above that substitutions on term indices in scope nodes ensures scope parallelism. This is now illustrated with a more interesting example (adapted from Hirshbühler as cited by DSP).

(10) A Canadian flag hung in front of every house, and an American flag did too.

The antecedent has two possible scopings: a single Canadian flag in front of all the houses, or each house with its own flag. Whichever scoping is given to the antecedent, a parallel scoping should be given to the ellipsis.

A simplified QLF for (10) is

(11) $?S1$: and($?S2$: hang(**term**(+c, $\exists$,...), **term**(+h, $\forall$,...)), $?P$(**term**(+a, $\exists$,...)))

where the indices +c, +a and +h are mnemonic for Canadian flag, American flag and house. Taking the first conjunct as the antecedent, we can set up an equation

(12) $?S2$:hang(**term**(+c...), **term**(+h...)) $= ?P$(**term**(+c...))

the solution to which is[3]

(13) $?P = T$^($?S2$:hang(**term**(+c...), **term**(+h...)) | {**term**(+c...)/T, +c/idx_of(T)}

This make the elliptical conjunct equivalent to

(14) $?S2$:hang(**term**(+c...), **term**(+h...)) | {**term**(+c...)/**term**(+a,...), +c/+a}

The scope node, $?S2$ can be resolved to [+h, +c] ('every house' takes wide scope), or [+c, +h] ('a Canadian flag' takes wide scope). Whichever resolution is made, the substitution of +a for +c ensures parallel scoping in the ellipsis for 'an American flag'. Cashing out the substitutions for the first case, we have

[3] In the next section we place some extra constraints on possible solutions, but these aren't strictly relevant here.

(15) $[]$:and([+h, +c]:hang(**term**(+c,∃,...),
term(+h,∀,...)),
[+h, +a]:hang(**term**(+a,∃,...),
term(+h,∀,...)))

There is another scoping option which instantiates ?$S1$ to [+h], i.e. gives 'every house' wide scope over both antecedent and ellipsis. In this case the two terms, **term**(+h...) in ellipsis and antecedent are both discharged (i.e. bound) at the scope node ?$S1$, rather than being separately bound at the two copies of ?$S2$

(16) [+h]:and([+c]:hang(**term**(+c,∃,...),
term(+h,∀,...)),
[+a]:hang(**term**(+a,∃,...),
term(+h,∀,...)))

(This has equivalent truth-conditions to (15)).[4]

Besides illustrating scope parallelism, this is an example where DSP have to resort to higher-order unification beyond second-order matching. But no such increase in complexity is required under the present treatment.

2.4 Strict and Sloppy Identity

The notion of strict and sloppy identity is usually confined to pronominal items occurring in antecedents and (implicitly) in ellipses.[5] A standard example is

(17) John loves his mother, and Simon does too.

On the strict reading, Simon and John both love John's mother. The implicit pronoun has been strictly identified with the pronoun in the antecedent to pick out the same referent, John. On the sloppy reading Simon loves Simon's mother. The implicit pronoun has been sloppily identified with its antecedent to refer to something matching a similar description, i.e. the subject or agent of the loving relation, Simon.

The sentence

(18) John read a book he owned, and so did Simon.

has three readings: John and Simon read the same book; John and Simon both read a book belonging to John, though not necessarily the same one; John reads one of John's books and Simon reads one of Simon's books.

Intuitively, the first reading arises from strictly identifying the elliptical book with the antecedent book. The second arises from strictly identifying the pronouns, while sloppily identifying the books. The third from sloppily identifying both the books and the pronouns. In the literature, the first reading would not be viewed as a case of strict identity. But this view emerges naturally from our treatment of substitutions, and is arguably a more natural characterisation of the phenomena.

We need to distinguish between parallel and non-parallel terms in ellipsis antecedents. Parallel terms, like *John* in the example above, are those that correspond terms appearing explicitly in the ellipsis. Non-parallel terms are those that do not have an explicit parallel in the ellipsis. (Determining which terms are parallel/non-parallel is touched on in section 4.)

For parallel terms, we have no choice about the ellipsis substitution. We replace both the term and its index by the corresponding term and index from the ellipsis. But for *all* non-parallel terms we have a choice between a strict or a sloppy substitution.[6]

A sloppy substitution involves substituting a new term index for the old one. This has the effect of reindexing the version of the term occurring in the ellipsis, so that it refers to the same kind of thing as the antecedent term but is not otherwise linked to it.

A strict substitution substitutes the term by its index. In this way, the version of the term occurring in the ellipsis is directly linked to antecedent term.

To illustrate, an abbreviated QLF for the antecedent *John read a book he owned* is

(19) ?S:
read(
term(+j...)
term(+b,∃,
λy.book(y)∧
own(**term**(+h...rft(+j)), y),
...))

Here, we have left the scope node as an uninstantiated meta-variable ?S. The pronominal term +h occurs in the restriction of the book term +b. The pronoun has been resolved to have a contextual restriction, rft(+j), that co-indexes it with the subject term. Here, 'rft(.)' is a function that when applied to an entity-denoting expression (e.g. a variable or constant) returns the property of being identical to that entity; when it applies to a term index, it returns an E-type property contextually linked to the term.

The ellipsis can be represented as

(20) ?P(**term**(+s, ∃, λy.name(y,'Simon'),...))

which is conjoined with the antecedent.

The three readings of (18) are illustrated below, listing substitutions to be applied to the antecedent

[4] If +c is given wide scope over antecedent and ellipsis, the QLF is rendered uninterpretable, which is as required. As detailed in section 3, scoping +c discharges the term and its index by substituting a variable for it. But the ellipsis substitution overrides this, substituting a new term and index, +a. But there is no way of discharging them.

[5] Also to pronouns of laziness.

[6] This is true of the non-parallel **term**(+h,...) in example (11); but this added complication does not affect the basic account of scope parallelism given earlier.

and cashing out the results of their application, though omitting scope.

(21) *Strict book*

$\{$+j/+s, **term**(+j,...)/**term**(+s,...),
term(+b,...)/+b, ...$\}$

read(**term**(+s,...), +b)

(a) Since all reference to the term +h is removed by the strict substitution on the term in which it occurs, it makes no difference whether the pronoun is given a strict or a sloppy substitution. (b) Strict substitution for the book leaves behind an occurrence of the index +b in the ellipsis. For the QLF to be interpretable, it is necessary to give the antecedent book term wide scope over the ellipsis in order to discharge the index.

(22) *Sloppy book, strict pronoun*

$\{$+j/+s, **term**(+j,...)/**term**(+s,...),
+b/+b1, **term**(+h,...)/+h$\}$

```
read(
  term(+s...)
  term(+b1,∃,
        λy.book(y)∧
           own(+h, y)
        ...))
```

As above, the antecedent pronoun is constrained to be given wide scope over the ellipsis, on pain of the index +h being undischargeable. (Pronouns, like proper names, are treated as contextually restricted quantifiers, where the contextual restriction may limit the domain of quantification to one individual.)

(23) *Sloppy book, sloppy pronoun*

$\{$+j/+s, **term**(+j,...)/**term**(+s,...),
+b/+b1, +h/+h1$\}$

```
read(
  term(+s...)
  term(+b1,∃,
        λy.book(y)∧
           own(term(+h1...rft(+s)), y),
        ...))
```

The index substitution from the primary term re-indexes the contextual restriction of the pronoun. It becomes coindexed with +s instead of +j.

DSP's account of the first reading of (18) is significantly different from their account of the last two readings. The first reading involves scoping the book quantifier before ellipsis resolution. The other two readings only scope the quantifier after resolution, and differ in giving the pronoun a strict or a sloppy interpretation. In our account the choice of strict or sloppy substitutions for secondary terms can constrain permissible quantifier scopings.[7] But the making of these choices does not have to be interleaved in a precise order with the scoping of quantifiers.

Moreover, the difference between strict and sloppy readings does not depend on somehow being able to distinguish between primary and secondary occurrences of terms with the same meaning. In DSP's representation of the antecedent of (18), both NPs 'John' and 'he' give rise to two occurrences of the same term (a constant, j). The QLF representation is able to distinguish between the primary and the secondary, pronominal, reference to John.

2.5 Other Phenomena

Space precludes illustrating the substitutional approach through further examples, though more are discussed in (Alshawi et al., 1992; Cooper et al., 1994b). The coverage is basically the same as DSP's:

Antecedent Contained Deletion: A sloppy substitution for *every person that Simon did* in the sentence *John greeted every person that Simon did* results in re-introducing the ellipsis in its own resolution. This leads to an uninterpretable cyclic QLF in much the same way that DSP obtain a violation of the occurs check on sound unification.

Cascaded Ellipsis: The number of readings obtained for *John revised his paper before the teacher did, and then Simon did* was used as a benchmark by DSP. The approach here gets the four readings identified by them as most plausible. With slight modification, it gets a fifth reading of marginal plausibility. The modification is to allow (strict) substitutions on terms not explicitly appearing in the ellipsis antecedent — i.e. the implicit *his paper* in the second ellipsis when resolving the third ellipsis.

We do not get a sixth, implausible reading, provided that in the first clause *his* is resolved as being coindexed with the **term** for *John*; i.e. that *John* and *his* do not both *independently* refer to the same individual. Kehler blocks this reading in a similar manner. DSP block the reading by a more artificial restriction on the depth of embedding of expressions in logical forms; they lack the means for distinguishing between coindexed and merely co-referential expressions.

Multiple VP Ellipsis Multiple VP ellipsis (Gardent, 1993) poses problems at the level of determining which VP is the antecedent of which ellipsis. But at the level of incorporating elliptical material once

[7] The converse also holds. Giving an antecedent term wide scope over the ellipsis renders the choice of a strict or a sloppy substitution for it in the ellipsis immaterial. During semantic evaluation of the QLF, discharging the antecedent through scoping will substitute out all occurrence of the term and its index before ellipsis substitutions are applied. Note though that this order dependence applies at the level of evaluating QLFs, not constructing and resolving them.

the antecedents have been determined, it appears to offer no special problems.

Other Forms of Ellipsis: Other forms of ellipsis, besides VP-ellipsis can be handled substitutionally. For example, NP-ellipsis (e.g. *Who slept? John.*) is straightforwardly accommodated. PP-ellipsis (e.g. *Who left on Tuesday? And on Wednesday?*) requires substitutions for `form` constructions in QLF (not described here) representing prepositional phrases.

2.6 Comparisons

The use of terms and indices has parallels to proposals due to Kehler and Kamp (Kehler, 1993a; Gawron and Peters, 1990). Kehler adopts an analysis where (referential) arguments to verbs are represented as related to a Davidsonian event via thematic role functions, e.g. *agent(e)=john)*. Pronouns typically refer to these functions, e.g. *he=agent(e)*. In VP ellipsis, strict identity corresponds to copying the entire role assignment from the antecedent. Sloppy identity corresponds to copying the function, but applying it to the event of the ellided clause.

For Kamp, strict identity involves copying the discourse referent of the antecedent and identifying it with that of the ellided pronoun. Sloppy identity copies the conditions on the antecedent discourse referent, and applies them to the discourse referent of the ellided pronoun.

Neither Kamp nor Kehler extend their copying/substitution mechanism to anything besides pronouns, as we have done. In Kehler's case, it is hard to see how his role assignment functions can be extended to deal with non-referential terms in the desired manner. DRT's use of discourse referents to indicate scope suggests that Kamp's treatment may be more readily extended in this manner; lists of discourse referents at the top of DRS boxes are highly reminiscent of the index lists in scope nodes.

3 Semantic Evaluation

Figure 1 defines a valuation relation for the QLF fragment used above, derived from (Alshawi and Crouch, 1992; Cooper et al., 1994a). If a QLF expression contains uninstantiated meta-variables, the valuation relation can associate more than one value with the expression. In the case of formulas, they may be given both the values true and false, corresponding to the formula being true under one possible resolution and false under another. A subsumption ordering over QLFS, $\sqsupseteq$, is employed in the evaluation rules, in effect to propose possible instantiations for meta-variables (the rule fragment only allows for scope meta-variables, but (Cooper et al., 1994a) describes the more general case where other kinds of meta-variable are permitted). A partially instantiated QLF therefore effectively specifies a set of possible evaluations (or semantic compositions).

Definition of $\mathcal{V}(QLF, M, g, Subs, v)$

where QLF is a QLF expression
- M is a model, $\langle O, F\rangle$
- g is an assignment of values to variables
- $Subs$ is a set of substitutions
- v is a value assigned to the QLF expression

1. Constant symbols, c: $\mathcal{V}(c, M, g, Subs, v)$ iff $F(c) = v$ (where F is the interpretation function for non-logical constants provided by M)
2. Variables, x: $\mathcal{V}(x, M, g, Subs, v)$ iff $g(x) = v$
3. Reinterpretation:
$\mathcal{V}(QLF_1, M, g, Subs, v)$ iff $\mathcal{V}(QLF_2, M, g, Subs, v)$ where $QLF_1/QLF_2 \in Subs$
4. Merging reinterpretations:
$\mathcal{V}(QLF|Subs_1, M, g, Subs_2, v)$ if
$\mathcal{V}(QLF, M, g, Subs_1 \uplus Subs_2, v)$
5. Abstraction:
$\mathcal{V}(\lambda x.\phi, M, g, Subs, h)$ if
$\phi \sqsupseteq \phi'$ and h is such that $\mathcal{V}(\phi', M, g^x_k, Subs, v)$ iff $h(k, v)$
6. Application:
$\mathcal{V}(p(a_1, \ldots, a_n), M, g, Subs, P(A_1, \ldots, A_n))$ if
$p(a_1, \ldots, a_n) \sqsupseteq p'(a'_1, \ldots, a'_n)$,
$\mathcal{V}(p', M, g, Subs, P)$,
$\mathcal{V}(a'_1, M, g, Subs, A_1)$,
$\ldots$, and
$\mathcal{V}(a'_n, M, g, Subs, A_n)$
7. ^-Application:
$\mathcal{V}(X\hat{}\phi\ (T),\ M, g, Subs, v)$ if
$\mathcal{V}(\phi \mid \{X/T\},\ M, g, Subs, v)$
8. Scoped formula:
$\mathcal{V}(Scope{:}\phi, M, g, Subs, v)$ if
$\mathcal{V}(Q'(R', \phi'), M, g, Subs, v)$
where
a) ϕ is a formula containing the term, T_0, $\mathtt{term}(I_0, Q_0, R_0, P_0)$
b) $newexpr(T_0, Subs) = T = \mathtt{term}(I, C, R, Q, P)$
c) $Scope \sqsupseteq [I, \ldots]$
d) R' is $\lambda x.(\mathrm{and}(R(x), P'(x)) \mid \{I/x\})$
e) ϕ' is $\lambda x.([\ldots]{:}\phi \mid \{T/x, I/x\})$

Operations on substitutions:

- $Subs_1 \uplus Subs_2$ combines two sets of substitutions. This is like set union, except that where $Subs_1$ and $Subs_2$ both substitute for a particular item, the substitution from $Subs_1$ is retained and not that in $Subs_2$.
- $newexpr(Old, Subs)$ returns New if $Old/New \in Subs$ and otherwise Old.

Figure 1: QLF Evaluation Rules

As the QLF becomes more instantiated, the set of possible evaluations narrows towards a singleton.

It is also possible for a QLF to be uninterpretable; to specify no possible evaluation. Thus, no rules are given for evaluating terms or their indices in isolation. They must first be discharged by the scoping rule, which substitutes the terms and indices by λ-bound variables. Inappropriate scoping may leave undischarged and hence uninterpretable terms and indices (which accounts for the so-called free-variable and vacuous quantification constraints on scope (Alshawi and Crouch, 1992)). The substitutions employed by the evaluation rule for scoping achieve a similar effect to the introduction and discharging of quantifier assumptions in Pereira's (1990) categorial semantics.

The non-deterministic nature of evaluation and the role of substitutions draws us to conclude that ellipsis substitutions operate on (descriptions of) the semantic compositions, not the results of such compositions.

4 Parallelism and Inference

Selecting ellipsis antecedents and parallel elements within them is an open problem (Prüst, 1992; Prüst et al., 1994; Kehler, 1993b; Grover et al., 1994). Our approach to parallelism is perhaps heavy-handed, but in the absence of a clear solutions, possibly more flexible. The QLFs shown above omitted category information present in `terms` and `forms`.[8] Categories are sets of feature value equations containing syntactic information relevant to determining how uninstantiated meta-variables can be resolved.

Tense in VP-ellipsis illustrates how categories can be put to work. In

(24) I enjoyed it. And so will you

the ellipsis is contained within a form expression whose category is

```
vp_ellipsis[tense=inf,modal=will,perfect=_,
            progressive=_,pol=pos,...]
```

This states the syntactic tense, aspect and polarity marked on the ellipsis (underscores indicate lack of specification). The category constrains resolution to look for verb phrase/sentence sources, which come wrapped in `forms` with categories like

```
vp[tense=past,modal=no,perfect=no,
   progressive=no,pol=pos,...]
```

Heuristics similar to those described by Hardt (1992) may be used for this. The category also says that, for this kind of VP match[9], the term in the antecedent whose category identifies it as being the subject should be treated as parallel to the explicit term in the ellipsis.

As this example illustrates, tense and aspect on ellipsis and antecedent do not have to agree. When this is so, the antecedent and ellipsis categories are both used to determine what `form` should be substituted for the antecedent `form`. This comprises the restriction of the antecedent `form` and a new category constructed by taking the features of the antecedent category, unless overridden by those on the ellipsis—a kind of (monotonic) priority union (Grover et al., 1994) except using skeptical as opposed to credulous default unification (Carpenter, 1993). When a new category is constructed for the antecedent, any tense resolutions also need to be undone, since the original ones may no longer be appropriate for the revised category. One thus merges the category information from source and antecedent to determine what verb phrase form should be substituted for the original. In this case, it will have a category

```
vp[tense=inf,modal=will,perfect=no,
   progressive=no,pol=neg,...]
```

A more general question is whether all ellipses involve recompositions, with variants, of linguistic antecedents. There are cases where a degree of inference seems to be required:

(25) We spent six weeks living in France, eating French food and speaking French, as we did in Austria the year before.

(one must apply the knowledge that Austrians speak German to correctly interpret the ellipsis). Pulman's (1994) equational treatment of context-dependency suggests one method of dealing with such cases. But it remains to be seen how readily the equations used for ellipsis here can be integrated into Pulman's framework.

5 Conclusions: Interpretation as Description

The substitutional treatment of ellipsis presented here has broadly the same coverage as DSP's higher-order unification treatment, but has the computational advantages of (i) not requiring order-sensitive interleaving of different resolution operations, and (ii) not requiring greater than second-order matching for dealing with quantifiers. In addition, it cures a slight overgeneration problem in DSP's account.

It has been claimed that these advantages arise from viewing semantic interpretation as a process of building descriptions of semantic compositions. To conclude, a few further arguments for this view, that are independent of any particular proposals for dealing with ellipsis.

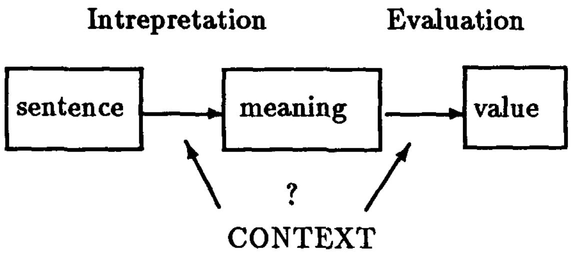
Order-Independence: One of the reasons for the computational success of unification-based syntactic formalisms is the order-independence of parser/generator operations they permit. If one looks at the order-sensitive nature of the operations of semantic compositions, they provide a poor starting point for a treatment of semantics enjoying similar computational success. But semantic interpreta-

[8] `forms` are described in (Alshawi and Crouch, 1992).

[9] Not all VP ellipses have VP antecedents.

tion, viewed as building a description of the intended composition, is a better prospect.

Context-Sensitivity: The truth values of many (all?) sentences undeniably depend on context. Context-dependence may enter either at the interpretive mapping from sentence to meaning and/or the evaluative mapping from meaning (and the world) to truth-values.



The more that context-dependence enters into the interpretive mapping (so that meanings are correspondingly more context-independent), the harder it is to maintain a principle of strict compositionality in interpretation. The syntactic structure underspecifies the intended composition, so that the meanings of some constituents (e.g. pronouns) and the mode of combination of other (e.g. quantifiers) are not fully specified. Further contextual information is required to fill the gaps. Again, interpretation seen as description building sits easily with this.

Preserving Information: Focusing exclusively on the results of semantic composition, i.e. meanings, can ignore differences in how those meanings were derived that can be linguistically significant (e.g. co-referential vs co-indexed terms). If this information is not to be lost, some way of referring to the structure of the compositions, as well as to their results, seems to be required.

Acknowledgements

The initial implementation of the work described here was carried out as part of the CLARE project, DTI IED4/1/1165. The writing of this paper was supported in part by the FraCaS project, LRE 62-051. I would especially like to thank Hiyan Alshawi and Steve Pulman for help and advice on topics relating to this paper. I have also benefited from conversations with Claire Grover, Ian Lewin and Massimo Poesio.

References

Hiyan Alshawi and Richard Crouch. 1992. Monotonic semantic interpretation. In *Proceedings 30th Annual Meeting of the Association for Computational Linguistics*, pages 32–38.

Hiyan Alshawi, David Carter, Richard Crouch, Stephen Pulman, Manny Rayner, and Arnold Smith. 1992. Clare: A contextual reasoning and cooperative response framework for the core language engine. Technical Report CRC-028, SRI International, Cambridge Computer Science Research Centre. Available by anonymous ftp from ftp.ai.sri.com, pub/sri-cam/reports/crc028.ps.Z; also cmp-lg.

Bob Carpenter. 1993. Skeptical and credulous default unification with applications to templates and inheritance. In E. Briscoe, V. de Paiva, and A. Copestake, editors, *Inheritance, Defaults and the Lexicon*, pages 13–37. Cambridge University Press.

Robin Cooper, Richard Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspars, Hans Kamp, Manfred Pinkal, Massimo Poesio, and Stephen Pulman. 1994a. Describing the approaches. FraCaS deliverable, D8. Available by anonymous ftp from ftp.cogsci.ed.ac.uk, pub/FRACAS/del8.ps.gz.

Robin Cooper et al. 1994b. Evaluating the descriptive capabilities of semantic theories. FraCaS deliverable, D9. Available by anonymous ftp from ftp.cogsci.ed.ac.uk, pub/FRACAS/del9.ps.gz.

Mary Dalrymple, Stuart M. Shieber, and Fernando C. N. Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14:399–452.

Claire Gardent. 1993. A unification-based approach to multiple vp ellipsis resolution. In *Proceedings 6th European ACL*, pages 139–148.

Mark Gawron and Stanley Peters. 1990. *Anaphora and Quantification in Situation Semantics*. Number 19 in CSLI Lecture Notes. CSLI, Stanford, CA.

Claire Grover, Chris Brew, Suresh Manandhar, and Marc Moens. 1994. Priority union and generalization in discourse grammars. In *Proceedings 32nd Annual Meeting of the Association for Computational Linguistics*, pages 17–24.

Daniel Hardt. 1992. An algorithm for vp ellipsis. In *Proceedings 30th Annual Meeting of the Association for Computational Linguistics*, pages 9–14.

Andrew Kehler. 1993a. A discourse copying algorithm for ellipsis and anaphora resolution. In *Proceedings 6th European ACL*, pages 203–212.

Andrew Kehler. 1993b. The effect of establishing coherence in ellipsis and anaphora resolution. In *Proceedings 31st Annual Meeting of the Association for Computational Linguistics*, pages 62–69.

John Nerbonne. 1991. Constraint-based semantics. In *Proceedings 8th Amsterdam Colloquium*, pages 425–443.

Fernando C. N. Pereira. 1990. Categorial semantics and scoping. *Computational Linguistics*, 16:1–10.

Hub Prüst, Remko Scha, and Martin van den Berg. 1994. Discourse grammar and verb phrase anaphora. *Linguistics and Philosophy*, 17:261–327.

Hub Prüst. 1992. *On Discourse Structure, VP Anaphora and Gapping*. Ph.D. thesis, University of Amsterdam.

Stephen Pulman. 1994. A computational theory of context dependency. In *Proceedings of the International Workshop on Computational Semantics*, pages 161–171, Tilburg.