The Semantics of Resource Sharing in Lexical-Functional Grammar

Andrew Kehler*
Aiken Computation Lab
33 Oxford Street
Harvard University
Cambridge, MA 02138

Mary Dalrymple†
John Lamping†
Vijay Saraswat†
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304

Abstract

We argue that the resource sharing that is commonly manifest in semantic accounts of coordination is instead appropriately handled in terms of structure-sharing in LFG f-structures. We provide an extension to the previous account of LFG semantics (Dalrymple et al., 1993a) according to which dependencies between f-structures are viewed as resources; as a result a one-to-one correspondence between uses of f-structures and meanings is maintained. The resulting system is sufficiently restricted in cases where other approaches overgenerate; the very property of resource-sensitivity for which resource sharing appears to be problematic actually provides explanatory advantages over systems that more freely replicate resources during derivation.

1 Introduction

The resource-based approach to semantic composition in Lexical-Functional Grammar (LFG) obtains the interpretation for a phrase via a logical deduction, beginning with the interpretations of its parts as premises (Dalrymple et al., 1993a). The resource-sensitive system of *linear logic* is used to compute meanings in accordance with relationships manifest in LFG f-structures. The properties of the system ensure that meanings are used exactly once, allowing *coherence* and *completeness* conditions on f-structures (Kaplan and Bresnan, 1982, pages 211–212) to be maintained.

However, there are cases where a single constituent appears to yield more than one contribution to the meaning of an utterance. This is most obvious in, but is not limited to, sentences involving coordination. In example (1), for instance, *NAFTA* is the object of two different verbs:

(1) Bill supported, and Hillary opposed, NAFTA.

Since the hallmark of the linear logic approach is to ensure that f-structure contributions are utilized exactly once in a derivation, such constructions would at first glance appear to be problematic for the approach.

We argue that the resource sharing that is commonly manifest in the treatment of coordination in other approaches is appropriately handled by exploiting the structure-sharing in LFG f-structures. We refine our previous analysis to account for cases where an f-structure is reached by multiple paths from an enclosing f-structure.

Dalrymple et al. (1993a) provides an account of LFG semantics that represents the meaning of lexical items with linear logic formulas. These formulas manipulate basic assertions of the form $f_\sigma \leadsto M$, for *f-structures* f and *meaning logic terms* M. Here σ is a mapping, the *semantic projection*, that relates f-structures to semantic structures. To distinguish between multiple paths entering an f-structure, we now take σ to map from sets of paths in f-structures to semantic structures. Further, the paths between f-structures are made available in the semantic space as resources. This makes it possible for the semantic formulas to exploit information about the multiple paths into an f-structure in order to account for the multiple uses of the f-structure's semantic contribution. The resulting system is sufficiently restricted in cases where other approaches overgenerate; the very property of resource-sensitivity for which resource sharing appears to be problematic actually provides explanatory advantages over systems that more freely replicate resources during derivation.

In Section 2, we review previous approaches to the semantics of coordination and argument shar-

*kehler@das.harvard.edu
†{dalrymple,lamping,saraswat}@parc.xerox.com

ing, and make note of some of their drawbacks. We describe the revised semantic framework in Section 3, and work through several examples of non-constituent coordination (specifically, right-node raising) in Section 4. We discuss examples involving intensional verbs in Section 5.

2 Previous Work

2.1 Combinatory Categorial Grammar

Steedman (1985; 1989; 1990), working in the framework of Combinatory Categorial Grammar (CCG), presents what is probably the most adequate analysis of non-constituent coordination to date. As noted by Steedman and discussed by Oehrle (1990), the addition of the rule of function composition to the inventory of syntactic rules in Categorial Grammar enables the formation of constituents with right-peripheral gaps, providing a basis for a clean treatment of cases of right node raising as exemplified by sentence (1). Such examples are handled by a coordination schema which allows like categories to be conjoined, shown in (2).

(2) Coordination: X CONJ X $\Rightarrow$ X

This schema gives rise to various actual rules whose semantics depends on the number of arguments that the shared material takes. For the cases of RNR considered here, the rule has the form shown in (3).

(3) (coordination)

X/Y:F CONJ:& X/Y:G $\Rightarrow$ X/Y:$\lambda x.(Fx \& Gx)$

The contraction from $\lambda x.Fx$ and $\lambda x.Gx$ to $\lambda x.(Fx \& Gx)$ in this rule allows for the single argument to be utilized twice.

As noted by Hudson (1976), however, not all examples of RNR involve coordinate structures:

(4) Citizens who support, paraded against politicians who oppose, two trade bills.

Obviously, such cases fall outside of the purview of the coordination schema. An analysis for this sentence is available in the CCG framework by the addition of the *xsubstitute* combinator (Steedman, p.c.), as defined in Steedman (1987).

(5) (<xsubstitute)

Y/Z:G (X\Y)/Z:F $\Rightarrow$ X/Z: $\lambda x.(Fx(Gx))$

The use of this combinator assimilates cases of noncoordinate RNR to cases involving parasitic gaps.

While this approach has some drawbacks,[1] we do not offer a competing analysis of the syntax of sentences like (4) here. Rather, we seek an analysis of RNR (and of resource sharing in general) that is uniform in the semantics; such a treatment isn't available in CCG because of its tight integration between syntax and semantics.

2.2 Partee and Rooth

Perhaps the most influential and widely-adopted semantic treatment of coordination is the approach of Partee and Rooth (1983). They propose a generalized conjunction scheme in which conjuncts of the same type can be combined. As is the case with Steedman's operators, contraction inherent in the schema allows for a single shared argument to be distributed as an argument of each conjunct. Type-lifting is allowed to produce like types when necessary; the combination of the coordination scheme and type-lifting can have the effect of 'copying' an argument of higher type, such as a quantifier in the case of coordinated intensional verbs. They propose a 'processing strategy' requiring that expressions are interpreted at the lowest possible type, with type-raising taking place only where necessary.

To illustrate, Partee and Rooth assume that extensional verbs such as *find* are entered in the lexicon with basic type $\langle e, \langle e, t \rangle \rangle$, whereas intensional verbs like *want*, which require a quantifier as an argument, have type $\langle \langle \langle e, t \rangle, t \rangle, \langle e, t \rangle \rangle$ (ignoring intensionality). Two extensional verbs such as *find* and *support* are coordinated at their basic types:

(6) *find and support* (type $\langle e, \langle e, t \rangle \rangle$):

$\lambda y.\lambda x.[find(x, y) \wedge support(x, y)]$

Two intensional verbs such as *want* and *seek* are also coordinated at their basic (higher) types:

(7) *want and seek* (type $\langle \langle \langle e, t \rangle, t \rangle, \langle e, t \rangle \rangle$):

$\lambda \mathcal{P}.\lambda x.[want(x, \mathcal{P}) \wedge seek(x, \mathcal{P})]$

The argument to this expression is a quantified NP. When an intensional and an extensional verb are coordinated, the extensional verb must be

[1] We find two problems with the approach as it stands. First, the intuition that one gap is 'parasitic' upon the other in cases like (4) is not strong, whereas the CCG analysis suggests an asymmetry between the two gaps. Second, the combinator appears to cause overgeneration. While it allows sentence (4), it also allows sentence (b), where *two trade bills* is analyzed as the object of both verbs:

(b) *Politicians who oppose, paraded against, two trade bills.

type-raised to promote it to the type of the intensional verb:

(8) *want and find* (type $\langle\langle\langle e,t\rangle,t\rangle,\langle e,t\rangle\rangle$):

$$\lambda\mathcal{P}.\lambda x.[want(x,\mathcal{P}) \wedge \mathcal{P}(\lambda y.find(x,y))]$$

Again, this leads to the desired result. However, an unwelcome consequence of this approach, which appears to have gone unnoticed in the literature, arises in cases in which more than two verbs are conjoined. If an intensional verb is coordinated with more than one extensional verb, a copy of the quantifier will be distributed to each verb in the coordinate structure. For instance, in (9), two extensional verbs and an intensional verb are coordinated.

(9) *want, find, and support*:

$$\begin{aligned}\lambda\mathcal{P}.\lambda x.[\ & want(x,\mathcal{P}) \\ & \wedge \mathcal{P}(\lambda y.find(x,y)) \\ & \wedge \mathcal{P}(\lambda y.support(x,y))\ ]\end{aligned}$$

Application of this expression to a quantifier results in two quantifiers being scoped separately over the extensional verbs. This is the wrong result; in a sentence such as *Hillary wanted, found, and supported two candidates*, the desired result is where *one* quantifier scopes over both extensional verbs (that is, Hillary found and supported the same two candidates), just as in the case where all the verbs are extensional. Further, there does not seem to be an obvious way to modify the Partee and Rooth proposal so as to produce the correct result, the problem being that the ability to copy quantifiers inherent in their schema is too unrestricted.

A second problem with the account is that, as with Steedman's coordination schema, Partee and Rooth's type-raising strategy only applies to coordinate structures. However, the need to type-raise extends to cases not involving coordination, as in sentence (10).

(10) Citizens who seek, paraded against politicians who have, a decent health insurance policy.

We will present an analysis that preserves the intuition underlying Partee and Rooth's processing strategy, but that predicts and generates the correct reading for cases such as (9). Furthermore, the account applies equally to examples not involving coordination, as is the case in sentence (10).

3 LFG and Linear Logic

LFG assumes two syntactic levels of representation: constituent structure (*c-structure*)[2] encodes phrasal dominance and precedence relations, and functional structure (*f-structure*) encodes syntactic predicate-argument structure. The f-structure for sentence (11) is given in (12):

(11) Bill supported NAFTA.

(12)
$$f:\begin{bmatrix}\text{PRED} & \text{`SUPPORT'} \\ \text{SUBJ} & g:\begin{bmatrix}\text{PRED} & \text{`BILL'}\end{bmatrix} \\ \text{OBJ} & h:\begin{bmatrix}\text{PRED} & \text{`NAFTA'}\end{bmatrix}\end{bmatrix}$$

Lexical entries specify syntactic constraints on f-structures as well as semantic information:

(13)
Bill	NP	$(\uparrow \text{PRED}) = \text{`BILL'}$ $\uparrow_\sigma \rightsquigarrow Bill$
supported	V	$(\uparrow \text{PRED}) = \text{`SUPPORT'}$ $\forall X,Y.\ \langle\uparrow \text{SUBJ}\rangle_\sigma \rightsquigarrow X$ $\otimes \langle\uparrow \text{OBJ}\rangle_\sigma \rightsquigarrow Y$ $\multimap \uparrow_\sigma \rightsquigarrow supported(X,Y)$
NAFTA	NP	$(\uparrow \text{PRED}) = \text{`NAFTA'}$ $\uparrow_\sigma \rightsquigarrow NAFTA$

Semantic information is expressed in (1) a *meaning language* and (2) a language for assembling meanings, or *glue language*. The meaning language could be that of any appropriate logic; for present purposes, higher-order logic will suffice. Expressions of the meaning language (such as *Bill*) appear on the right side of the meaning relation $\rightsquigarrow$.

The glue language is the *tensor* fragment of *linear logic* (Girard, 1987). The semantic contribution of each lexical entry, which we will refer to as a *meaning constructor*, is a linear-logic formula consisting of instructions in the glue language for combining the meanings of the lexical entry's syntactic arguments to obtain the meaning of the f-structure headed by the entry. For instance, the meaning constructor for the verb *supported* is a glue language formula paraphrasable as: "If my SUBJ means X and ($\otimes$) my OBJ means Y, then ($\multimap$) my sentence means $supported(X,Y)$".

In the system described in Dalrymple et al. (1993a), the $\rightsquigarrow$ relation associates expressions in the meaning language with f-structures. As a result, each f-structure contributed a single meaning constructor as a resource to be used in a derivation. Because linear logic does not have any form of logical contraction (as is inherent in

[2] For discussion of c-structure and its relation to f-structure, see, for example, Kaplan and Bresnan (1982).

the approaches discussed earlier), cases where resources are shared appear to be problematic in this framework. Intuitively, however, the need for the multiple use of an f-structure meaning results not from the appearance of a particular lexical item (e.g., a conjunction) or a particular syntactic construction (e.g., parasitic gap constructions), but instead results from multiple paths to it from within the f-structure that contains it, where structure sharing is motivated on syntactic grounds. We therefore revise the earlier framework to model what we will term *occurrences* of f-structures as resources explicitly in the logic.

F-structures can mathematically be regarded as (finite) functions from a set of attributes to a set of atomic values, semantic forms and (recursively) f-structures. We will identify an occurrence of an f-structure with a path (from the root) to that occurrence; sets of occurrences of an f-structure can therefore be identified with path sets in the f-structure. We take, then, the domain of the σ projection to be path sets in the root f-structure. Only those path sets S are considered which satisfy the property that the extensions of each path in S are identical. Therefore the f-structure reached by each of these paths is identical. Hence from a path set S, we can read off an f-structure S_f. In the examples discussed in Dalrymple et al. (1993a) there is a one-to-one correspondence between the set of path sets S and the set of f-structures S_f picked out by such path sets, so the two methods yield the same predictions for those cases.

Relations between path sets are represented explicitly as resources in the logic by *R-relations*. R-relations are represented as three-place predicates of the form $R(F,\ P,\ G)$ which indicate that (the path set) G appears at the end of a path P (of length 1) extending (the path set) F. That is, the f-structure G_f appears at the end of the singleton path P in the f-structure F_f. For example, the f-structure given in (12) results in two R-relations:

$$\begin{array}{ll} (i) & R(f, \textsc{subj},\ g) \\ (ii) & R(f, \textsc{obj},\ h) \end{array}$$

Because f and g represent path sets entering an f-structure that they label, R-relation (i) indicates that the set of paths $\langle f\ \textsc{subj}\rangle$ (which denotes the set of paths f concatenated with SUBJ) is a subset of the set of paths denoted by g. An axiom for interpretation provides the links between meanings of path sets related by R-relations.

$$\textbf{Axiom I}: \quad \begin{array}{l} !(\forall F, G, P, X.\ G_\sigma \rightsquigarrow X \\ \quad \multimap\ !(R(F,P,G) \multimap \langle F\ P\rangle_\sigma \rightsquigarrow X)) \end{array}$$

According to this axiom, if a set of paths G has meaning X, then for each R-relation $R(F,P,G)$ that has been introduced, a resource $\langle F\ P\rangle_\sigma \rightsquigarrow X$ can be produced. The linear logic operator '!' allows the conclusion $(R(F,P,G) \multimap \langle F\ P\rangle_\sigma \rightsquigarrow X)$ to be used as many times as necessary: once for each R-relation $R(F,P,G)$ introduced by the f-structure.

We show how a deduction can be performed to derive a meaning for example (11) using the meaning constructors in (13), R-relations (i) and (ii), and Axiom I. Instantiating the lexical entries for *Bill*, *NAFTA*, and *supported* according to the labels on the f-structure in (12), we obtain the following premises:

$$\begin{array}{ll} \textbf{bill}: & g_\sigma \rightsquigarrow \mathit{Bill} \\ \textbf{NAFTA}: & h_\sigma \rightsquigarrow \mathit{NAFTA} \\ \textbf{supported}: & \forall X, Y.\ \langle f\ \textsc{subj}\rangle_\sigma \rightsquigarrow \mathrm{X} \\ & \quad \otimes \langle f\ \textsc{obj}\rangle_\sigma \rightsquigarrow \mathrm{Y} \\ & \quad \multimap f_\sigma \rightsquigarrow \mathit{supported}(X,Y) \end{array}$$

First, combining Axiom I with the contribution for *Bill* yields:

(14) $!\,\forall F, P.\ R(F,\ P,\ g) \multimap \langle F\ P\rangle_\sigma \rightsquigarrow \mathit{Bill}$

This formula states that if a path set is R-related to the (path set corresponding to the) f-structure for *Bill*, then it receives *Bill* as its meaning. From R-relation (i) and formula (14), we derive (15), giving the meaning of the subject of f.

(15) $\langle f\ \textsc{subj}\rangle_\sigma \rightsquigarrow \mathit{Bill}$

The meaning constructor for *supported* combines with (15) to derive the formula for **bill-supported** shown in (16).

(16) $\forall Y.\ \langle f\ \textsc{obj}\rangle_\sigma \rightsquigarrow Y$
$\quad \multimap f_\sigma \rightsquigarrow \mathit{supported}(\mathit{Bill}, Y)$

Similarly, using the meaning of *NAFTA*, R-relation (ii), and Axiom I, we can derive the meaning shown in (17):

(17) $\langle f\ \textsc{obj}\rangle_\sigma \rightsquigarrow \mathit{NAFTA}$

and combine it with (16) to derive (18):

(18) $f_\sigma \rightsquigarrow \mathit{supported}(\mathit{Bill}, \mathit{NAFTA})$

At each step, universal instantiation and modus ponens are used. A second derivation is also possible, in which **supported** and **NAFTA** are combined first and the result is then combined with **Bill**.

The use of linear logic provides a flexible mechanism for deducing meanings of sentences based on their f-structure representations. Accounts of various linguistic phenomena have been developed within the framework on which our extension is based, including quantifiers and anaphora (Dalrymple et al., 1994a), intensional verbs (Dalrymple et al., 1994b), and complex predicates (Dalrymple et al., 1993b). The logic fits well with the 'resource-sensitivity' of natural language semantics: there is a one-to-one correspondence between f-structure relationships and meanings; the multiple use of resources arises from multiple paths to them in the f-structure. In the next section, we show how this system applies to several cases of right-node raising.

4 Examples

4.1 RNR with Coordination

First we consider the derivation of the basic case of right-node raising (RNR) illustrated in sentence (1), repeated in (19).

(19) Bill supported, and Hillary opposed, NAFTA.

The f-structure for example (19) is shown in (20).

(20)
$$f\colon \left\{ \begin{array}{l} f_1\colon \left[\begin{array}{ll} \textsc{pred} & \text{`SUPPORT'} \\ \textsc{subj} & g\colon\left[ \textsc{pred} \quad \text{`BILL'} \right] \\ \textsc{obj} & h\colon\left[\textsc{pred} \quad \text{`NAFTA'} \right] \end{array} \right] \\ \\ f_2\colon \left[ \begin{array}{ll} \textsc{pred} & \text{`OPPOSE'} \\ \textsc{subj} & i\colon\left[\textsc{pred} \quad \text{`HILLARY'} \right] \\ \textsc{obj} & \end{array} \right] \end{array} \right\}$$

The meaning constructors contributed by the lexical items are as follows:

$$\begin{array}{ll}
\textbf{Bill}: & g_\sigma \rightsquigarrow \mathit{Bill} \\
\textbf{Hillary}: & i_\sigma \rightsquigarrow \mathit{Hillary} \\
\textbf{supported}: & \forall X, Y.\ \langle f_1\ \textsc{subj}\rangle_\sigma \rightsquigarrow X \\
& \quad \otimes \langle f_1\ \textsc{obj}\rangle_\sigma \rightsquigarrow Y \\
& \quad \multimap f_{1\sigma} \rightsquigarrow \mathit{supported}(X, Y) \\
\textbf{opposed}: & \forall X, Y.\ \langle f_2\ \textsc{subj}\rangle_\sigma \rightsquigarrow X \\
& \quad \otimes \langle f_2\ \textsc{obj}\rangle_\sigma \rightsquigarrow Y \\
& \quad \multimap f_{2\sigma} \rightsquigarrow \mathit{opposed}(X, Y) \\
\textbf{and}: & \forall X, Y.\ \langle f\ \textsc{conj}\rangle_\sigma \rightsquigarrow X \\
& \quad \otimes \langle f\ \textsc{conj}\rangle_\sigma \rightsquigarrow Y \\
& \quad \multimap f_\sigma \rightsquigarrow \mathit{and}(X, Y) \\
\textbf{and2}: & !(\forall X, Y.\ \langle f\ \textsc{conj}\rangle_\sigma \rightsquigarrow X \\
& \quad \otimes f_\sigma \rightsquigarrow Y \\
& \quad \multimap f_\sigma \rightsquigarrow \mathit{and}(X, Y)) \\
\textbf{NAFTA}: & h_\sigma \rightsquigarrow \mathit{NAFTA}
\end{array}$$

Here, we treat *and* as a binary relation. This suffices for this example, but in general we will have to allow for cases where more than two constituents are conjoined. Therefore, a second meaning constructor **and2** is also contributed by the appearance of *and*, prefixed with the linear logic operator '!', so that it may be used as many times as necessary (and possibly not at all, as is the case in this example).

The R-relations resulting from the feature-value relationships manifest in the f-structure in (20) are:[3]

$$\begin{array}{ll}
(i) & R(f, \textsc{conj}, f_1) \\
(ii) & R(f, \textsc{conj}, f_2) \\
(iii) & R(f_1, \textsc{subj}, g) \\
(iv) & R(f_1, \textsc{obj}, h) \\
(v) & R(f_2, \textsc{subj}, i) \\
(vi) & R(f_2, \textsc{obj}, h)
\end{array}$$

There are several equivalent derivation orders; here we step through one.[4] Using the meanings for *Bill, supported, Hillary,* and *opposed*, R-relations (iii) and (v), and Axiom I, we can derive meanings for *Bill supported* and *Hillary opposed* in the fashion described in Section 3:

$$\begin{array}{ll}
\textbf{bill-supported}: & \forall Y.\ \langle f_1\ \textsc{obj}\rangle_\sigma \rightsquigarrow Y \\
& \quad \multimap f_{1\sigma} \rightsquigarrow \mathit{supported}(\mathit{Bill}, Y) \\
\textbf{hillary-opposed}: & \forall Z.\ \langle f_2\ \textsc{obj}\rangle_\sigma \rightsquigarrow Z \\
& \quad \multimap f_{2\sigma} \rightsquigarrow \mathit{opposed}(\mathit{Hillary}, Z)
\end{array}$$

We combine the antecedents and consequents of the foregoing formulae to yield:

$$\begin{array}{l}
\textbf{bill-supported} \otimes \textbf{hillary-opposed}: \\
\forall Y, Z.\ \langle f_1\ \textsc{obj}\rangle_\sigma \rightsquigarrow Y \otimes \langle f_2\ \textsc{obj}\rangle_\sigma \rightsquigarrow Z \\
\quad \multimap f_{1\sigma} \rightsquigarrow \mathit{supported}(\mathit{Bill}, Y) \\
\quad\quad \otimes f_{2\sigma} \rightsquigarrow \mathit{opposed}(\mathit{Hillary}, Z)
\end{array}$$

Consuming the meaning of *and* and R-relations (i) and (ii), and using Axiom I, we derive:

$$\begin{array}{l}
\textbf{bill-supported-and-hillary-opposed1}: \\
\forall Y, Z.\ \langle f_1\ \textsc{obj}\rangle_\sigma \rightsquigarrow Y \otimes \langle f_2\ \textsc{obj}\rangle_\sigma \rightsquigarrow Z \\
\quad \multimap f_\sigma \rightsquigarrow \mathit{and}(\mathit{supported}(\mathit{Bill}, Y), \\
\quad\quad\quad\quad \mathit{opposed}(\mathit{Hillary}, Z))
\end{array}$$

Using Axiom I and R-relations (iv) and (vi), the following implication can be derived:

$$\begin{array}{l}
\forall X.\ h_\sigma \rightsquigarrow X \\
\quad \multimap \langle f_1\ \textsc{obj}\rangle_\sigma \rightsquigarrow X \otimes \langle f_2\ \textsc{obj}\rangle_\sigma \rightsquigarrow X
\end{array}$$

Using these last two formulae, by transitivity we obtain:

$$\begin{array}{l}
\textbf{bill-supported-and-hillary-opposed2}: \\
\forall X.\ h_\sigma \rightsquigarrow X \\
\quad \multimap f_\sigma \rightsquigarrow \mathit{and}(\mathit{supported}(\mathit{Bill}, X), \\
\quad\quad\quad\quad \mathit{opposed}(\mathit{Hillary}, X))
\end{array}$$

[3] We treat the CONJ features as unordered, as they are in the f-structure set.

[4] In the interest of space, we will skip some intermediate steps in the derivation.

Finally, consuming the contribution of *NAFTA*, by universal instantiation and modus ponens we obtain a meaning for the whole sentence:

$$f_\sigma \rightsquigarrow and(supported(Bill, NAFTA),\\ opposed(Hillary, NAFTA))$$

At this stage, all accountable resources have been consumed, and the deduction is complete.

4.2 RNR with Coordination and Quantified NPs

We now consider sentence (21), where a quantified NP is shared.

(21) Bill supported, and Hillary opposed, two trade bills.

Partee and Rooth (1983) observe, and we agree, that the quantifier in such cases only scopes once, resulting in the reading where Bill supported and Hillary opposed the same two bills.[5] Our analysis predicts this fact in the same way as Partee and Rooth's analysis does.

The meanings contributed by the lexical items and f-structure dependencies are the same as in the previous example, except for that of the object NP. Following Dalrymple et al. (1994a), the meaning derived using the contributions from an f-structure h for *two trade bills* is:

two-trade-bills:
$$\forall H, S.\ (\forall x.\ h_\sigma \rightsquigarrow x \multimap H \rightsquigarrow S(x))\\ \multimap H \rightsquigarrow two(z, tradebill(z), S(z))$$

The derivation is just as before, up until the final step, where we have derived the formula labeled **bill-supported-and-hillary-opposed2**. This formula matches the antecedent of the quantified NP meaning, so by universal instantiation and modus ponens we derive:

$$f_\sigma \rightsquigarrow two(z, tradebill(z), and(supported(Bill, z),\\ opposed(Hillary, z)))$$

With this derivation, there is only one quantifier meaning which scopes over the meaning of the coordinated material. A result where the quantifier meaning appears twice, scoping over each conjunct separately, is not available with the rules we have given thus far; we return to this point in Section 5.

The analysis readily extends to cases of noncoordinate RNR such as example (4), repeated as example (22).

[5]We therefore disagree with Hendricks (1993), who claims that such sentences readily allow a reading involving four trade bills.

(22) Citizens who support, paraded against politicians who oppose, two trade bills.

In our analysis, the f-structure for *two trade bills* is resource-shared as the object of the two verbs, just as it is in the coordinated case.

Space limitations preclude our going through the derivation; however, it is straightforward given the semantic contributions of the lexical items and R-relations. The fact that there is no coordination involved has no bearing on the result, since the semantics of resource-sharing is distinct from that of coordination in our analysis. As previously noted, this separation is not possible in CCG because of the tight integration between syntax and semantics. In LFG, the syntax/semantics interface is more loosely coupled, affording the flexibility to handle coordinated and non-coordinated cases of RNR uniformly in the semantics. This also allows for our semantics of coordination not to require schemas nor entities of polymorphic type; our meaning of *and* is type $t \times t \to t$.

5 Intensional Verbs

We now return to consider cases involving intensional verbs. The preferred reading for sentence (23), in which only one quantifier scopes over the two extensional predicates, is shown below:

(23) Hillary wanted, found, and supported two candidates.

$$and(wanted(Hillary,\\ {}^\wedge\lambda Q.two(x, candidate(x), [{}^\vee Q](x))),\\ two(z, candidate(z),\\ and(found(Hillary, z),\\ supported(Hillary, z))))$$

The f-structure for example (23) is given in (24).

(24)
$$f: \left\{ \begin{array}{l} f_1: \begin{bmatrix} \text{PRED} & \text{`WANT'} \\ \text{SUBJ} & g: \begin{bmatrix} \text{PRED} & \text{`HILLARY'} \end{bmatrix} \\ \text{OBJ} & h: \begin{bmatrix} \text{PRED} & \text{`CANDIDATE'} \\ \text{SPEC} & \text{`TWO'} \end{bmatrix} \end{bmatrix} \\ f_2: \begin{bmatrix} \text{PRED} & \text{`FIND'} \\ \text{SUBJ} & \\ \text{OBJ} & \end{bmatrix} \\ f_3: \begin{bmatrix} \text{PRED} & \text{`SUPPORT'} \\ \text{SUBJ} & \\ \text{OBJ} & \end{bmatrix} \end{array} \right\}$$

The meaning constructors for the lexical items are given in Figure 1. Recall that a second meaning

Hillary:	$g_\sigma \rightsquigarrow \mathit{Hillary}$
wanted:	$\forall X, Y.\ \langle f_1\ \textsc{subj}\rangle_\sigma \rightsquigarrow X \otimes (\forall s, p.\ (\forall X.\ \langle f_1\ \textsc{subj}\rangle_\sigma \rightsquigarrow X \multimap s \rightsquigarrow p(X)) \multimap s \rightsquigarrow Y(\hat{\ }p)) \multimap f_{1\sigma} \rightsquigarrow \mathit{wanted}(X, \hat{\ }Y)$
found:	$\forall X, Y.\ \langle f_2\ \textsc{subj}\rangle_\sigma \rightsquigarrow X \otimes \langle f_2\ \textsc{obj}\rangle_\sigma \rightsquigarrow Y \multimap f_{2\sigma} \rightsquigarrow \mathit{found}(X, Y)$
supported:	$\forall X, Y.\ \langle f_3\ \textsc{subj}\rangle_\sigma \rightsquigarrow X \otimes \langle f_3\ \textsc{obj}\rangle_\sigma \rightsquigarrow Y \multimap f_{3\sigma} \rightsquigarrow \mathit{supported}(X, Y)$
and:	$\forall X, Y.\ \langle f\ \textsc{conj}\rangle_\sigma \rightsquigarrow X \otimes \langle f\ \textsc{conj}\rangle_\sigma \rightsquigarrow Y \multimap f_\sigma \rightsquigarrow \mathit{and}(X, Y)$
and2:	$!(\forall X, Y.\ \langle f\ \textsc{conj}\rangle_\sigma \rightsquigarrow X \otimes f_\sigma \rightsquigarrow Y \multimap f_\sigma \rightsquigarrow \mathit{and}(X, Y))$
two-candidates:	$\forall H, S.\ (\forall x.\ h_\sigma \rightsquigarrow X \multimap H \rightsquigarrow S(x)) \multimap H \rightsquigarrow \mathit{two}(z, \mathit{candidate}(z), S(z))$

Figure 1: Meaning constructors for sentence (23)

constructor **and2** is introduced by *and* in order to handle cases where there are more than two conjuncts; this contribution will be used once in the derivation of the meaning for sentence (23). The following R-relations result from the f-structural relationships:

(i)	$R(f, \textsc{conj},\ f_1)$
(ii)	$R(f, \textsc{conj},\ f_2)$
(iii)	$R(f, \textsc{conj},\ f_3)$
(iv)	$R(f_1, \textsc{subj},\ g)$
(v)	$R(f_2, \textsc{subj},\ g)$
(vi)	$R(f_3, \textsc{subj},\ g)$
(vii)	$R(f_1, \textsc{obj},\ h)$
$(viii)$	$R(f_2, \textsc{obj},\ h)$
(ix)	$R(f_3, \textsc{obj},\ h)$

Following the analysis given in Dalrymple et al. (1994b), the lexical entry for *want* takes a quantified NP as an argument. This requires that the quantified NP meaning be duplicated, since otherwise no readings result. We provide a special rule for duplicating quantified NPs when necessary:

(25) **QNP Duplication**:

$$\begin{array}{l} !(\forall F, Q. \\ \qquad [\forall H, S.\ (\forall x.\ F_\sigma \rightsquigarrow x \multimap H \rightsquigarrow S(x)) \\ \qquad\qquad \multimap H \rightsquigarrow Q(S)] \\ \multimap \qquad [\ \ [\forall H, S.\ (\forall x.\ F_\sigma \rightsquigarrow x \multimap H \rightsquigarrow S(x)) \\ \qquad\qquad \multimap H \rightsquigarrow Q(S)] \\ \qquad\quad \otimes [\forall H, S.\ (\forall x.\ F_\sigma \rightsquigarrow x \multimap H \rightsquigarrow S(x)) \\ \qquad\qquad \multimap H \rightsquigarrow Q(S)]\ ]) \end{array}$$

In the interest of space, again we only show a few steps of the derivation. Combining the meanings for *Hillary, found, supported,* and *and*, Axiom I, and R-relations (ii), (iii), (v), (vi), (viii), and (ix), we can derive:

$$\begin{array}{l} h_\sigma \rightsquigarrow x \multimap f_\sigma \rightsquigarrow \mathit{and}(\mathit{found}(\mathit{Hillary}, x), \\ \qquad\qquad\qquad \mathit{supported}(\mathit{Hillary}, x))) \end{array}$$

We duplicate the meaning of *two candidates* using QNP Duplication, and combine one copy with the foregoing formula to yield:

$$\begin{array}{l} f_\sigma \rightsquigarrow \mathit{two}(z, \mathit{candidate}(z), \\ \qquad\qquad \mathit{and}(\mathit{found}(\mathit{Hillary}, z), \\ \qquad\qquad\qquad \mathit{supported}(\mathit{Hillary}, z))) \end{array}$$

We then combine the other meaning of *two candidates* with the meanings of *Hillary* and *wanted*, and using Axiom I and R-relations (i), (iv), and (vii) we obtain:

$$\begin{array}{l} (f\ \textsc{conj})_\sigma \rightsquigarrow \\ \qquad \mathit{wanted}(\mathit{Hillary}, \\ \qquad\qquad \hat{\ }\lambda Q.\mathit{two}(z, \mathit{candidate}(z), [\check{\ }Q](z))) \end{array}$$

Finally, using *and2* with the two foregoing formulae, we deduce the desired result:

$$\begin{array}{l} f_\sigma \rightsquigarrow \mathit{and}(\mathit{wanted}(\mathit{Hillary}, \\ \qquad\qquad\qquad \hat{\ }\lambda Q.\mathit{two}(x, \mathit{candidate}(x), [\check{\ }Q](x))), \\ \qquad\quad \mathit{two}(z, \mathit{candidate}(z), \\ \qquad\qquad\quad \mathit{and}(\mathit{found}(\mathit{Hillary}, z), \\ \qquad\qquad\qquad \mathit{supported}(\mathit{Hillary}, z)))) \end{array}$$

We can now specify a Partee and Rooth style processing strategy, which is to prefer readings which require the least use of QNP duplication. This strategy predicts the readings generated for the examples in Section 4. It also predicts the desired reading for sentence (23), since that reading requires two quantifiers. While the reading generated by Partee and Rooth is derivable, it requires three quantifiers and thus uses QNP duplication twice, which is less preferred than the reading requiring two quantifiers which uses QNP duplication once. Also, it allows some flexibility in cases where pragmatics strongly suggests that quantifiers are copied and distributed for multiple extensional verbs; unlike the Partee and Rooth account, this would apply equally to the case where there are also intensional verbs and the case where there are not. Finally, our account readily applies to cases of intensional verbs without coordination as in example (10), since it applies more generally to cases of resource sharing.

6 Conclusions and Future Work

We have given an account of resource sharing in the syntax/semantics interface of LFG. The multiple use of semantic contributions results from viewing dependencies in f-structures as resources; in this way the one-to-one correspondence between f-structure relations and meanings is maintained. The resulting account does not suffer from overgeneration inherent in other approaches, and applies equally to cases of resource sharing that do not involve coordination. Furthermore, it lends itself readily to an extension for the intensional verb case that has advantages over the widely-assumed account of Partee and Rooth (1983).

Here we have separated the issue of arriving at the appropriate f-structure in the syntax from the issue of deriving the correct semantics from the f-structure. We have argued that this is the correct distinction to make, and have given a treatment of the second issue. A treatment of the first issue will be articulated in a future forum.

Acknowledgements

We would like to thank Sam Bayer, John Maxwell, Fernando Pereira, Dick Oehrle, Stuart Shieber, and especially Ron Kaplan for helpful discussion and comments. The first author was supported in part by National Science Foundation Grant IRI-9009018, National Science Foundation Grant IRI-9350192, and a grant from the Xerox Corporation.

References

Mary Dalrymple, John Lamping, and Vijay Saraswat. 1993a. LFG semantics via constraints. In *Proceedings of the Sixth Meeting of the European ACL*, University of Utrecht, April. European Chapter of the Association for Computational Linguistics.

Mary Dalrymple, Angie Hinrichs, John Lamping, and Vijay Saraswat. 1993b. The resource logic of complex predicate interpretation. In *Proceedings of the 1993 Republic of China Computational Linguistics Conference (ROCLING)*, Hsitou National Park, Taiwan, September. Computational Linguistics Society of R.O.C.

Mary Dalrymple, John Lamping, Fernando C. N. Pereira, and Vijay Saraswat. 1994a. A deductive account of quantification in LFG. In Makoto Kanazawa, Christopher J. Piñón, and Henriette de Swart, editors, *Quantifiers, Deduction, and Context*. Center for the Study of Language and Information, Stanford, California. To appear.

Mary Dalrymple, John Lamping, Fernando C. N. Pereira, and Vijay Saraswat. 1994b. Intensional verbs without type-raising or lexical ambiguity. In *Conference on Information-Oriented Approaches to Logic, Language and Computation*, Moraga, California. Saint Mary's College.

J.-Y. Girard. 1987. Linear logic. *Theoretical Computer Science*, 45:1–102.

Herman Hendriks. 1993. *Studied Flexibility: Categories and Types in Syntax and Semantics.* ILLC dissertation series 1993—5, University of Amsterdam, Amsterdam, Holland.

Richard A Hudson. 1976. Conjunction reduction, gapping, and right-node raising. *Language*, 52(3):535–562.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press, Cambridge, MA.

Richard T. Oehrle. 1990. Categorial frameworks, coordination, and extraction. In Aaron Halpern, editor, *Proceedings of the Ninth West Coast Conference on Formal Linguistics*, pages 411–425.

Barbara Partee and Mats Rooth. 1983. Generalized conjunction and type ambiguity. In Rainer Bauerle, Christoph Schwarze, and Arnim von Stechow, editors, *Meaning, Use, and Interpretation of Language*, pages 361–383. de Gruyter, Berlin.

Mark J. Steedman. 1985. Dependency and coordination in the grammar of Dutch and English. *Language*, 61:523–568.

Mark J. Steedman. 1987. Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory*, 5:403–439.

Mark J. Steedman. 1989. Constituency and coordination in a combinatory grammar. In Mark Baltin and Anthony Kroch, editors, *Alternative Conceptions of Phrase Structure*, pages 201–231. Chicago University Press.

Mark J. Steedman. 1990. Gapping as constituent coordination. *Linguistics and Philosophy*, 13(2):207–263.