Coordination and Parallelism in Glue Semantics: Integrating Discourse Cohesion and the Element Constraint

Ash Asudeh and Richard Crouch
Stanford University and PARC

Proceedings of the LFG02 Conference
National Technical University of Athens, Athens
Miriam Butt and Tracy Holloway King (Editors)

2002
CSLI Publications
`http://csli-publications.stanford.edu/`

Abstract We present initial work on a theory of coordination and parallelism in Glue Semantics (GLUE; Dalrymple 1999, 2001). We will explore points of convergence and divergence between our approach to coordination and similar Categorial Grammar (CG) approaches. We also compare our approach to a previous GLUE approach to coordination (Kehler et al. 1995, 1999) and argue that our approach is superior on the grounds that it preserves a very strong notion of resource-sensitivity (Dalrymple et al. 1993). We conclude by discussing parallelism in connection with the Coordinate Structure Constraint (CSC; Ross 1967). The CSC is a putatively robust condition on extraction which has been argued to be a feature of the CG approach to coordination and of other related approaches. It is standardly assumed to have two parts, the Conjunct Constraint and the Element Constraint (Grosu 1973). The Conjunct Constraint is quite robust, but the Element Constraint has been challenged repeatedly, most recently by Kehler (2002), who argues that the CSC is not a syntactic condition, but rather follows from conditions on discourse coherence and parallelism. We discuss a constraint language on the structure of GLUE derivations, and show how Kehler's theory of discourse cohesion can be related to parallelism in such derivations.

1 Introduction[1]

This paper presents an account of the semantics of coordination, framed within the theory of Glue Semantics (GLUE Dalrymple 1999, 2001). We compare this account to related work in Categorial Grammar (CG) (Steedman 1985, Emms 1990, Carpenter 1997), and an earlier GLUE approach to coordination (Kehler et al. 1995, 1999). We discuss parallelism in connection with the Coordinate Structure Constraint (CSC; Ross 1967), which is standardly assumed to have two parts: the Conjunct Constraint and the Element Constraint (Grosu 1973). Kehler (2002) discusses a set of exceptions to the Element Constraint and argues that it is not a syntactic condition, but follows from conditions on discourse coherence and parallelism. By treating GLUE derivations as first-class semantic representations on which interesting parallelism relations can be defined (Crouch 1999, Asudeh and Crouch to appear), we show how our account of coordination is able to deal with violations of the Element Constraint.

2 A Brief Overview of Glue Semantics

GLUE embodies a treatment of "semantic interpretation as deduction", similar to "parsing as deduction" in Categorial Grammar. GLUE identifies two separate logics in semantic interpretation: a meaning logic for expressing the target semantic representation; and a GLUE logic which specifies how chunks of meaning are deductively assembled. A variety of options are open for the meaning logic (IL, DRT, etc.). But the natural choice for the GLUE logic is a restricted fragment of propositional Linear Logic (Girard 1987). First, the resource-sensitivity of linear logic closely reflects that of natural language (Dalrymple et al. 1993). Second, the existence of a Curry-Howard isomorphism for the GLUE fragment of linear logic both renders it suitable for driving the deductive assembly of meanings, and also provides GLUE derivations with non-trivial identity criteria. We enlarge on these points below.

2.1 A Brief Overview of Linear Logic

Linear logic is resource-sensitive. Unlike traditional logics, linear logic derivations literally consume their premises in order to draw conclusions. This can be illustrated by the following contrastive patterns of inference ($\multimap$ is linear implication, and $\otimes$ is multiplicative conjunction):

---

[1] We would like to acknowledge Mary Dalrymple, Chris Potts, Mark Steedman and Ida Toivonen for comments on and criticism and discussion of various incarnations of these ideas. We would also like to acknowledge ourselves for all remaining errors. Asudeh is funded in part by SSHRC 752-98-0424.

	Traditional	Linear
Duplication	$a, a \to b \vdash b$ $a, a \to b \vdash b \wedge a$ *(a duplicated)*	$a, a \multimap b \vdash b$ $a, a \multimap b \not\vdash b \otimes a$ *(No duplication of a)*
Deletion	$a, b \vdash a \wedge b$ $a, b \vdash b$ *(a deleted)*	$a, b \vdash a \otimes b$ $a, b \not\vdash b$ *(No deletion of a)*

This ensures that each premise has to be used once, and exactly once, in a derivation; premises can be neither deleted nor duplicated.[2]

The same pattern of strict resource accounting is to be found in natural language: the contribution of each word and phrase must be used once and exactly once in the analysis of a sentence (Dalrymple et al. 1993, 1999b, Asudeh in progress). One cannot freely delete or duplicate the contributions that words make. As we will see in section 3, coordination provides a *prima facie* counterexample to this strict resource accounting. However, we will show how a strictly resourced account can be given.

The Curry-Howard Isomorphism (Howard 1980) pairs logical derivations with terms in the lambda-calculus. In particular, the proof rule of Implication Elimination (or *modus ponens*) corresponds to an operation of function application, while Implication Introduction (or hypothetical reasoning) corresonds to λ-abstraction:

(1) **Implication Elimination** **Implication Introduction**

$$\frac{P : a \multimap b \qquad Q : a}{P(Q) : b}\multimap_{\mathcal{E}} \qquad\qquad \frac{\begin{array}{c}[X : a]^i \\ \vdots \\ \phi : b\end{array}}{\lambda X.\phi : a \multimap b}\multimap_{\mathcal{I},i}$$

For Implication Elimination, if P is the λ-term labelling the derivation of $a \multimap b$, and Q is the term labelling the derivation of a, then $P(Q)$ is the term labelling the resultant derivation of b. The implication $a \multimap b$ is thus a function, P, which when applied to an argument Q of type a returns a result $P(Q)$ of type b. For Implication Introduction, suppose that assuming an arbitrary a, with a variable X labelling its unknown derivation, allows us to obtain a derivation ϕ of b. We can then discharge the assumption to get an implication from arguments of type a to results of type b, where the function corresponding to the implication is $\lambda X.\phi$.

Within the setting of GLUE, the λ-terms labelling linear logic formulas will be expressions from the meaning logic. Derivations will assemble these meanings by means of the function application and λ-abstraction operations defined by the Curry-Howard Isomorphism.

The following two derivations of $a, a \multimap b \vdash b$ show the interaction of the proof rules and λ-terms:

(2) (a) $\dfrac{A : a \quad P : a \multimap b}{P(A) : b}\multimap_{\mathcal{E}}$ (b) $\dfrac{A : a \qquad \dfrac{\dfrac{[X : a]^1 \qquad P : a \multimap b}{P(X) : b}\multimap_{\mathcal{E}}}{\lambda X.P(X) : a \multimap b}\multimap_{\mathcal{I},1}}{(\lambda X.P(X))(A) : b}\multimap_{\mathcal{E}}$

Derivation (2b) introduces an unnecessary detour. By applying $a \multimap b$ to an assumption of a and then immediately discharging the assumption, we rather pointlessly derive $a \multimap b$ from $a \multimap b$. We then apply this to the premise a to conclude b. Derivation (2a) is a more straightforward way of achieving the same result. Interestingly, the λ-terms of the two derivations are equivalent given η/β-reduction: $(\lambda X.P(X))(A) \Rightarrow P(A)$. This is non-accidental. The equivalence of the terms shows that the two derivations in (2a) and (2b) correspond to the same underlying proof.

The Curry-Howard Isomorphism thus induces non-trivial identity criteria for proofs, such that (2a) and (2b) denote the same proof, despite their surface differences. These identity criteria also match

---

[2] This strict resource accounting can locally be turned off by means of linear logic's ! modality, where $!a$ means that a is no longer resourced, and can be duplicated or deleted at will. However, we do not include ! in our GLUE fragment of linear logic; see section 3.4 below.

those induced by proof normalization (Prawitz 1965), which provides a set of rules (3) for expunging unnecessary detours from derivations

(3)
$$\dfrac{A \qquad \dfrac{\begin{matrix}[A]^i\\ \vdots\\ B\end{matrix}}{A \multimap B}\multimap_{\mathcal{I},i}}{B} \qquad \overset{\beta}{\Longrightarrow} \qquad \begin{matrix}A\\ \vdots\\ B\end{matrix}$$

The presence of such identity criteria (following Quine's dictum of "no entity without identity") allows us to regard proofs as first-class objects. In particular, normal form derivations (with all detours expunged) can be viewed as canonical representations for their underlying proofs. This opens the possibility of viewing normal form GLUE derivations as first-class levels of representation in semantic theory. In section 5 we will argue that GLUE proofs form an important level of representation in guaging semantic parallelism.

2.2 Examples of GLUE Derivations

In GLUE, meaning constructors for semantic composition are obtained from lexical items instantiated in particular syntactic structures. Each constructor has the form $\mathcal{M} : G$, where $\mathcal{M}$ is a term from some meaning language (e.g., IL, DRT, etc.) and G is a formula of propositional linear logic (Dalrymple et al. 1999a). The goal of a GLUE derivation is to consume all the lexical premises to produce a single conclusion stating the meaning of the sentence. Semantic ambiguity (e.g., scope ambiguity) results when there are alternative derivations from the same set of premises. The Curry-Howard Isomorphism combines lexical meanings in parallel with the structure of the linear logic deduction to build meaning terms.

In this paper we will assume an LFG syntax (see Dalrymple 2001, among others), with one important caveat to which we return in section 5.2, and a very generic predicate calculus, for the sake of exposition. Given these, consider (4) and its lexical items (5):

(4) John saw Fred.

(5)

John N	*Fred* N	*saw* V
$(\uparrow \text{PRED}) =$ 'John'	$(\uparrow \text{PRED}) =$ 'Fred'	$(\uparrow \text{PRED}) =$ 'see'
$\mathit{john} : \uparrow_{\sigma_e}$	$\mathit{fred} : \uparrow_{\sigma_e}$	$\mathit{see} : (\uparrow \text{OBJ})_{\sigma_e} \multimap (\uparrow \text{SUBJ})_{\sigma_e} \multimap \uparrow_{\sigma_t}$

The second line of each entry is its GLUE meaning constructor. The σ subscripts in the GLUE constructors are functions that map syntactic phrases onto their corresponding semantic resources. These resources are typed: e for entity, t for truth-value. The resources are denoted by atomic linear logic propositions. (We will suppress the σ and type subscripts on the linear logic atoms where convenient).

Parsing (4) both constructs the f(unctional)-structure (6) and instantiates the lexical entries so that the $\uparrow$ metavariables refer to nodes within (6), instantiating the lexical premises as in (7).

(6)
$$s\begin{bmatrix}\text{PRED} & \text{'see'}\\ \text{SUBJ} & j\begin{bmatrix}\text{PRED} & \text{'John'}\end{bmatrix}\\ \text{OBJ} & f\begin{bmatrix}\text{PRED} & \text{'Fred'}\end{bmatrix}\end{bmatrix}$$

(7)
$\mathit{john} : j_{\sigma_e}$
$\mathit{fred} : f_{\sigma_e}$
$\mathit{see} : f_{\sigma_e} \multimap j_{\sigma_e} \multimap s_{\sigma_t}$

The formulas in (7) are used as premises in a linear logic derivation. This must consume all the premises to produce a single conclusion stating the meaning paired with the head resource of the sentence (s_σ). In this case the derivation is straightforward: the three premises combine through two instance of implication elimination, which is functional application in the meaning language:

(8)
$$\dfrac{\dfrac{\dfrac{\mathit{fred} : f_e \qquad \mathit{see} : f_e \multimap j_e \multimap s_t}{\mathit{see}(\mathit{fred}) : j_e \multimap s_t}\multimap_{\mathcal{E}} \qquad \mathit{john} : j_e}{\mathit{see}(\mathit{fred})(\mathit{john}) : s_t}\multimap_{\mathcal{E}}}{\mathit{see}(\mathit{john},\ \mathit{fred}) : s_t}\ \mathit{Notational\ convention}$$

The second example will illustrate quantification. It also shows that quantifier scope ambiguity is handled in the GLUE derivations, without positing an ambiguous syntactic representation. Consider the following sentence, f-structure, and instantiated lexical items:

(9) Everyone found at least one gremlin.

(10)
$$f\left[\begin{array}{ll}\textsc{pred} & \text{‘find’}\\ \textsc{subj} & e\left[\begin{array}{ll}\textsc{pred} & \text{‘everyone’}\end{array}\right]\\ \textsc{obj} & g\left[\begin{array}{ll}\textsc{pred} & \text{‘gremlin’}\\ \textsc{spec} & \left[\begin{array}{ll}\textsc{pred} & \text{‘at least one’}\end{array}\right]\end{array}\right]\end{array}\right]$$

(11)
$$\begin{array}{l}\lambda P.\mathit{every}(\mathit{person},\ P) : (e_e \multimap X_t) \multimap X_t\\ \lambda u, v.\mathit{find}(v, u) : g_e \multimap e_e \multimap f_t\\ \lambda Q.\textsc{alo}(\mathit{gremlin},\ Q) : (g_e \multimap Y_t) \multimap Y_t\end{array}$$

The meaning terms $\lambda P.\mathit{every}(\mathit{person},\ P)$ and $\lambda Q.\textsc{alo}(\mathit{gremlin},\ Q)$ are standard generalized quantifier expressions, which we will henceforth abbreviate as EO and ALOG. Reading the types from the linear logic formulas, it can be see that both are of the (familiar) semantic type $\langle\langle e, t\rangle, t\rangle$. The upper case variables, X_t and Y_t, range over arbitrary type t atomic resources that the quantifiers could take as their scope. Essentially, the two quantifiers can apply to any type t clause that depends on the meaning of the subject e_e or the object g_e, and discharge this dependency by scoping the quantitifier.

From the three premises in (11), there are two distinct derivations f_t. Both have the same initial derivation (12), producing the semantic resource f_t dependent on both e_e and g_e. The derivations then fork, depending on which of these dependencies are discharged first via scoping a quantifier. ((13) for surface scope and (14) for inverse scope).

(12)
$$\dfrac{[x : e]^2 \qquad \dfrac{[y : g]^1 \qquad \lambda u, v.\mathit{find}(v, u) : g \multimap e \multimap f}{\lambda v.\mathit{find}(v, y) : e \multimap f}\multimap_{\mathcal{E}}}{\mathit{find}(x, y) : f}\multimap_{\mathcal{E}}$$

(13)
$$\dfrac{\textsc{eo} : (e \multimap X) \multimap X \qquad \dfrac{\dfrac{\textsc{alog} : (g \multimap Y) \multimap Y \qquad \dfrac{\mathit{find}(x, y) : f}{\lambda y.\mathit{find}(x, y) : g \multimap f}\multimap_{\mathcal{I},1}}{\textsc{alog}(\lambda y.\mathit{find}(x, y)) : f}\multimap_{\mathcal{E}Y=f}}{\lambda x.\textsc{alog}(\lambda y.\mathit{speak}(x, y)) : e \multimap f}\multimap_{\mathcal{I},2}}{\textsc{eo}(\lambda x.\textsc{alog}(\lambda y.\mathit{find}(x, y))) : f}\multimap_{\mathcal{E}X=f}$$

(14)
$$\dfrac{\textsc{alog} : (g \multimap Y) \multimap Y \qquad \dfrac{\dfrac{\textsc{eo} : (e \multimap X) \multimap X \qquad \dfrac{\mathit{find}(x, y) : f}{\lambda x.\mathit{find}(x, y) : e \multimap f}\multimap_{\mathcal{I},2}}{\textsc{eo}(\lambda x.\mathit{find}(x, y)) : f}\multimap_{\mathcal{E}X=f}}{\lambda y.\textsc{eo}(\lambda x.\mathit{find}(x, y)) : e \multimap s}\multimap_{\mathcal{I},1}}{\textsc{alog}(\lambda y.\textsc{eo}(\lambda x.\mathit{find}(x, y))) : f}\multimap_{\mathcal{E}Y=f}$$

Note that the two scopings for the sentence arise solely from alternative linear logic derivations from premises to conclusion, using standard rules of inference. No syntactic ambiguity needs to be posited. Moreover, no special assumptions need to be made about the meaning terms. Glue Semantics enforces a modular separation between the GLUE language and the target meaning language, so that the range of possible GLUE derivations is determined solely by the linear logic formulas expressing relations between semantic resources, and is entirely independent of the meaning terms associated with these resources. This modularity will be important when we come to define a level of semantic parallelism that abstracts away from differences in meaning (section 4.2).

3 Coordination in Glue Semantics

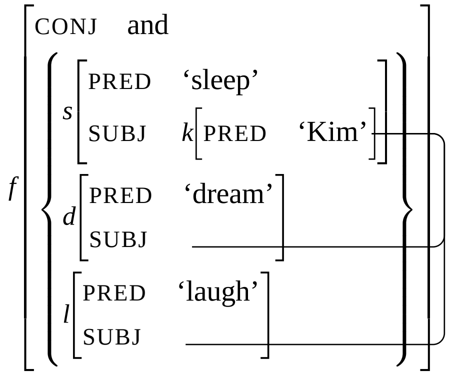
In this section we will look at two approaches to coordination in Glue Semantics (see also Dalrymple 2001:361–387). First we present our approach, which treats coordination essentially as modification and bears similarities to coordination in Montague Grammar, Generalized Phrase Structure Grammar (GPSG), Combinatory Categorial Grammar, and Type-Logical Grammar (Montague 1973, Gazdar 1980, 1981, 1982, Partee and Rooth 1983, Gazdar et al. 1985, Keenan and Faltz 1985, Steedman 1985, 1989, 1990, 2000, Emms 1990, Carpenter 1997). Second, we present and contrast a GLUE approach pursued by Kehler et al. (1995, 1999). Anticipating somewhat, the key difference between the two GLUE approaches is that the coordination as modification approach employs a resource management strategy to handle apparently conflicting resource consumption requirements, while the latter approach instead chooses to relax resource sensitivity.

3.1 An Example of VP Coordination

Let us start by considering a simple VP-coordination with three conjuncts:[3]

(15) Kim slept, dreamt and laughed.

(16)



(17) Target semantics = $sleep(kim) \wedge dream(kim) \wedge laugh(kim)$

The target semantic representation suggests that coordination poses a problem for the strict resource accounting inherent in GLUE. The single word *Kim* appears to make a threefold semantic contribution, as the subject argument to each of the three coordinated VPs. Spelling this out in terms of instantiated GLUE constructors, we would expect to obtain three ordinary intransitive verb premises but just one subject premise

(18)

kim	$kim : k$
slept	$sleep : k \multimap s$
dreamt	$dream : k \multimap d$
laughed	$laugh : k \multimap l$

How can a GLUE derivation match the one producer of the semantic resource k with the three consumers of k?

Our solution to this apparent resource mismatch is a treatment of coordination that consumes multiple dependencies on shared arguments to produce single dependencies on the shared arguments. Let

---

[3] We follow the usual convention of referring to the elements that are coordinated as conjuncts, even though the logical coordination relation is not limited to conjunction.

us add the following additional constructors, contributed by the coordination itself (For expository purposes, we here show these additional constructors in a simplified form. Their full form, (23), and how they are obtained is discussed below):

(19)
$$\begin{array}{l} \lambda P.P : (k \multimap l) \multimap (k \multimap f) \\ \lambda P, Q, x.P(x) \wedge Q(x) : (k \multimap d) \multimap (k \multimap f) \multimap (k \multimap f) \\ \lambda P, Q, x.P(x) \wedge Q(x) : (k \multimap s) \multimap (k \multimap f) \multimap (k \multimap f) \end{array}$$

The first constructor consumes the final conjunct VP ($k \multimap l$), and turns it into the 'seed' meaning for the coordinated VP ($k \multimap f$). This consumes one dependency on the argument k to produce another. The other two constructors each consume one of the remaining conjuncts, and turn them into modifiers of the seed VP meaning ($(k \multimap f) \multimap (k \multimap f)$). Each consumes one dependency on the shared argument k to modify the existing seed dependency on k. The GLUE derivation for the sentence proceeds as follows

(20)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{l} \mathit{laugh} : \\ k \multimap l \end{array} \quad \begin{array}{l} \lambda P.P : \\ (k \multimap l) \multimap (k \multimap f) \end{array}}{\begin{array}{l} \mathit{laugh} : \\ k \multimap f \end{array}} \quad \dfrac{\begin{array}{l} \mathit{dream} : \\ k \multimap d \end{array} \quad \begin{array}{l} \lambda P \lambda Q \lambda x.[P(x) \wedge Q(x)] : \\ (k \multimap d) \multimap ((k \multimap f) \multimap (k \multimap f)) \end{array}}{\begin{array}{l} \lambda Q \lambda x.[\mathit{dream}(x) \wedge Q(x)] : \\ (k \multimap f) \multimap (k \multimap f) \end{array}}}{\begin{array}{l} \lambda x.[\mathit{dream}(x) \wedge \mathit{laugh}(x)] : \\ k \multimap f \end{array}} \quad \dfrac{\begin{array}{l} \mathit{sleep} : \\ k \multimap s \end{array} \quad \begin{array}{l} \lambda P \lambda Q \lambda x.[P(x) \wedge Q(x)] : \\ (k \multimap s) \multimap ((k \multimap f) \multimap (k \multimap f)) \end{array}}{\begin{array}{l} \lambda Q \lambda x.[\mathit{sleep}(x) \wedge Q(x)] : \\ (k \multimap f) \multimap (k \multimap f) \end{array}}}{\begin{array}{l} \lambda x.[\mathit{sleep}(x) \wedge (\lambda x.[\mathit{dream}(x) \wedge \mathit{laugh}(x)])(x)] : \\ k \multimap f \end{array}}}{\begin{array}{l} \lambda x.[\mathit{sleep}(x) \wedge [\mathit{dream}(x) \wedge \mathit{laugh}(x)]] : \\ k \multimap f \end{array}} \quad \begin{array}{l} \mathit{kim} : \\ k \end{array}}{\begin{array}{l} [\mathit{sleep}(\mathit{kim}) \wedge [\mathit{dream}(\mathit{kim}) \wedge \mathit{laugh}(\mathit{kim})]] : \\ f \end{array}}$$

The derivation first consumes the three $k \multimap l, k \multimap s$ and $k \multimap d$ VPs to produce one $k \multimap f$ coordinate VP. Only then is the single k resource consumed.

We now address the question of where the additional GLUE constructors in (19) come from. In doing so, we will also generalize the constructors in (19) so that they apply to coordinators besides boolean conjunction.

In addition to purely lexical GLUE premises, we assume that c-structure rules can sometimes also introduce non-lexical, constructional GLUE premises (see also Dalrymple 2001). Such is the case with the rule for VP coordination (21), with the lexical entry for the conjunct *and* shown in (22).[4]

(21)
$$\mathrm{VP} \rightarrow \begin{array}{c} \mathrm{VP}^{+} \\ \downarrow \in \uparrow \\ \begin{array}{l} \lambda P, Q, C, x.\ C(P(x), Q(C,x)) : \\ \ [((\in \uparrow)_\sigma \textsc{crel}) \multimap (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma] \\ \ \ \multimap [((\in \uparrow)_\sigma \textsc{crel}) \multimap (\uparrow \textsc{subj})_\sigma \multimap (\in \uparrow)_\sigma] \\ \ \ \ \multimap [((\in \uparrow)_\sigma \textsc{crel}) \multimap (\uparrow \textsc{subj})_\sigma \multimap (\in \uparrow)_\sigma] \end{array} \end{array} \quad \begin{array}{c} \mathrm{Conj} \\ \uparrow = \downarrow \end{array} \quad \begin{array}{c} \mathrm{VP} \\ \downarrow \in \uparrow \\ \begin{array}{l} \lambda P, C.\ P : \\ \ [(\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma \\ \ \ \multimap [((\in \uparrow)_\sigma \textsc{crel}) \multimap (\uparrow \textsc{subj})_\sigma \multimap (\in \uparrow)_\sigma \end{array} \end{array}$$

(22) Lexical entry: *and* Conj
(↑ CONJ) = 'and'
$\mathit{and} : (\uparrow_\sigma \textsc{coord-rel})$

The c-structure rules assigns an additional GLUE premise to each VP conjunct. The final VP is the seed for the coordination. The meaning of the final VP, $(\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$, is consumed to produce an initial meaning for the coordinate VP, $(\uparrow \textsc{subj})_\sigma \multimap (\in \uparrow)_\sigma$.[5] This initial meaning is also *vacuously* dependent on the meaning of the coordinator, $((\in \uparrow)_\sigma \textsc{crel})$, provided by the lexical entry for the word *and*. Here, CREL (an abbreviation for COORDINATION-RELATION) is a feature in s(emantic)-structure, similar to other s-structure features like VAR(IABLE) and RESTR(ICTION) (for more details see Dalrymple 2001).

Non-final VP conjuncts each induce a constructor that consumes the conjunct VP meaning to produce a modifier of coordinate VP meaning. The meaning term for the constructor, $\lambda P, Q, C, x.\ C(P(x), Q(C, x)$ perhaps needs explanation. P represents the meaning of the conjunct

---

[4] We follow standard LFG practice in assuming that the Kleene plus (+) applies not just to the category, but also to its annotations.

[5] The coordinate f-structure is refered to by means of $(\in \uparrow)$, since the conjunct $\uparrow$ is contained within a set argument of the coordinate f-structure (Kaplan and Maxwell 1988).

VP, and Q the meaning of the seed, coordinate VP. C represents the meaning of the conjunction, and x the meaning of the shared subject. The constructor abstracts over all of these meanings, which is to say we are creating a function that takes them as arguments. The result of the function applies P to the subject x, and Q to both x and the conjunction C. The application of Q to C ensures that the value of the conjunction is threaded down throughout the coordinate VP. Finally, both the conjunct clause meaning, $P(x)$, and the coordinate clause meaning, $Q(C, x)$, are conjoined by the conjunction C.

Applying these rules in the setting of (16) (where c abbreviates the semantic resource $(f_\sigma$ CREL$)$ and f abbreviates the semantic resource for the entire coordinate structure), we obtain the following premises:

(23) 1. kim $\mathit{kim} : k$
2. slept $\mathit{sleep} : k \multimap s$
3. dreamt $\mathit{dream} : k \multimap d$
4. and $\mathit{and} : c$
5. laughed $\mathit{laugh} : k \multimap l$
6. $\lambda P, C.P : (k \multimap l) \multimap (c \multimap k \multimap f)$
7. $\lambda P, Q, C, x.\ C(P(x), Q(C, x)) : (k \multimap d) \multimap (c \multimap k \multimap f) \multimap (c \multimap k \multimap f)$
8. $\lambda P, Q, C, x.\ C(P(x), Q(C, x)) : (k \multimap s) \multimap (c \multimap k \multimap f) \multimap (c \multimap k \multimap f)$

Going through the derivation step by step, we first apply 6 to 5 to get the seed meaning for the coordination, 9.

(24) 9. $\lambda C.\ \mathit{laugh} : (c \multimap k \multimap f)$

We also apply 7 to 3 to obtain the seed modifier 10:

(25) 10. $\lambda Q, C, x.\ C(\mathit{dream}(x), Q(C, x)) : (c \multimap k \multimap f) \multimap (c \multimap k \multimap f)$

We then modify 9 by 10 to give 11

(26) 11. $\lambda C, x.\ C(\mathit{dream}(x),\ \mathit{laugh}(x)) : (c \multimap k \multimap f)$

Applying 8 to 2 gives another seed modifier, 12

(27) 12. $\lambda Q, C, x.\ C(\mathit{sleep}(x), Q(C, x)) : (c \multimap k \multimap f) \multimap (c \multimap k \multimap f)$

Applying 12 to 11 gives

(28) 13. $\lambda C, x.\ c(\mathit{sleep}(x),\ C(\mathit{dream}(x),\ \mathit{laugh}(x)) : (c \multimap k \multimap f)$

Applying 13 to 4 and then 1 finally yields the conclusion

(29) $\mathit{and}(\mathit{sleep}(\mathit{kim}),\ \mathit{and}(\mathit{dream}(\mathit{kim}),\ \mathit{laugh}(\mathit{kim})) : f$

Note how both the meaning of the subject and the meaning of the conjunction appear at several places in the final meaning term, without duplication of the corresponding GLUE resources.

In this section, we have deliberately chosen an example with three conjuncts. The recursive nature of our analysis (turn the rightmost conjunct into a seed meaning for the coordination, and all other conjuncts into modifiers of the coordination) ensures that we can deal with coordinations with any number of conjuncts. This addresses a criticism by Dalrymple (2001:379) that a previous version of the coordination-as-modification approach was insufficiently general.

Another criticism made by Dalrymple (2001) is that there are non-coordination cases where resource sharing is an issue, such as the following example, which she cites from Hudson (1976):

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

Although there is no coordination in this right-node raising case, the resource sharing issue is similar to the VP-coordination case we have been examining, as the verbs *support* and *paraded* share an object, *two trade bills*. The rule for right-node raising would handle the resource management of the VPs in essentially the same manner as the coordination rule, but without making the semantic contribution of coordination.

3.2 A General Semantic Schema for Coordination?

Our method for dealing with VP coordination readily generalizes to the coordination of any phrases with zero or more shared arguments (Kaplan and Maxwell 1988). It is tempting, therefore, to attempt a general schema for the semantics of coordination, along the lines of

(31)

$$\mathrm{X} \rightarrow \quad \begin{array}{c} \mathrm{X}^{+} \\ \downarrow\ \in\ \uparrow \\ \begin{array}{l} \lambda P, Q, C, \vec{x}.\ C(P(\vec{x}), Q(C, \vec{x})) : \\ [((\in\uparrow)_\sigma\ \mathrm{CREL}) \multimap \langle \overrightarrow{\alpha_{\multimap}} \rangle_n \multimap \uparrow_\sigma] \\ \quad \multimap [((\in\uparrow)_\sigma\ \mathrm{CREL}) \multimap \langle \overrightarrow{\alpha_{\multimap}} \rangle_n \multimap (\in\uparrow)_\sigma] \\ \quad \multimap [((\in\uparrow)_\sigma\ \mathrm{CREL}) \multimap \langle \overrightarrow{\alpha_{\multimap}} \rangle_n \multimap (\in\uparrow)_\sigma] \end{array} \end{array} \qquad \begin{array}{c} \mathrm{Conj} \\ \uparrow = \downarrow \end{array} \qquad \begin{array}{c} \mathrm{X} \\ \downarrow\ \in\ \uparrow \\ \begin{array}{l} \lambda P, C.\ P : \\ [\langle \overrightarrow{\alpha_{\multimap}} \rangle_n \multimap \uparrow_\sigma \\ \quad \multimap [((\in\uparrow)_\sigma\ \mathrm{CREL}) \multimap \langle \overrightarrow{\alpha_{\multimap}} \rangle_n \multimap (\in\uparrow)_\sigma] \end{array} \end{array}$$

where X specifies the category level of the coordination, $\langle \overrightarrow{\alpha_{\multimap}} \rangle_n$ represent a sequence of n implications over the shared argument resources of the conjuncts, and $\vec{x}$ is the corresponding sequence of meaning variables. The value of n is set by the level of the coordination: for sentential coordination, where there are no shared arguments, $n = 0$; for VP coordination $n = 1$, etc.

While is may be possible to define such a schematic c-structure rule and semantics for coordination, it is unclear that this is desirable. Let us leave aside the question of whether a single schema is syntactically possible. It is implausible to demand that all levels of coordination must have exactly the same semantics. For example, purely boolean coordination, of the kind outlined above, may be what is required for VPs. But for NPs, we might wish to adopt a semantics where the seed NP conjunct additionally introduces a group entity, and where the conjunct NPs quantify over elements of the group, e.g.

(32) Kim and/or a dog left
$\exists X.group(X) \wedge$
$and/or(member(kim, X),\ \exists d.\ dog(d) \wedge member(d, X))$
$\wedge\ leave(X)$

This is not a purely boolean coordination. However, the conjunctive word *and/or* does have a purely boolean semantics within the coordination. One would not be able to combine these two styles of semantics for different coordination levels under a single c-structure schema for coordination.

Instead, it is preferable to assume one c-structure coordination rule per category. The syntactic portion of each c-structure rule will indeed always have the form of the syntactic portion of (31), as shown in (33), where *X* can be any category of the language, including the categories for partial constituents, such as *x*-VP, defined by Maxwell and Manning (1996) in their treatment of nonconstituent coordination and coordination of unlikes.

(33)

$$\mathrm{X} \rightarrow \quad \begin{array}{ccc} \mathrm{X}^{+} & \mathrm{CONJ} & \mathrm{X} \\ \downarrow\ \in\ \uparrow & \uparrow = \downarrow & \downarrow\ \in\ \uparrow \end{array}$$

However, the semantics of the each rule can be tuned to suit the particular category in question.

Significant linguistic generalizations can be achieved by means of macros[6] encoding common patterns of analysis. Thus one can encode our semantic analysis of shared arguments in one macro, and invoke this in slightly different settings in various specific c-structure coordination rules.

3.3 A Brief Comparison to Coordination in Categorial Grammar

Our approach bears a lot of similarity to various Categorial Grammar (CG) approaches to coordination, including approaches in both Combinatory Categorial Grammar (CCG) and Type-Logical Grammar (TLG) (Steedman 1985, Emms 1990, Carpenter 1997). The boolean coordinator we introduced in (22)

---

[6] A macro is a device widely used in programming, and in computational grammars, where repeatedly used chunks of code/rules are written once, possibly parameterized, in a single place. Macro calls in the grammar are expanded out to be replaced by the rule chunks. If (33) were defined as a macro, parameterized by the category X, then multiple calls to the macro for different categories (S, VP, NP, etc.) would expand out to different instances of the coordination rule.

and the schema we use for coordination in (21) are similar to Emms's (1990) polymorphic generalization of Steedman's (1985) work, which is also adopted for Type-Logical Grammar by Carpenter (1997). We also noted that our syntactic schema is compatible with the previous LFG work on nonconstituent coordination and coordination of unlikes by Maxwell and Manning (1996).

The crucial distinction between our approach and the CCG/TLG approaches is that GLUE assumes a level of syntax that is separate from the level of semantic composition. Another interesting difference between the GLUE approach and categorial approaches is that the latter have tended to assume binary coordination.[7] The independent level of syntax in GLUE allows for constructional premises associated with c-structure rules, permitting a straightforward analysis of n-ary coordinations, such as *Kim slept, dreamt and laughed*. On a binary approach, one is forced to treat coordination syncategorematically or to treat the comma in written language or some phonetic cue in spoken language as an additional lexical conjunction. However, this is empirically unmotivated, since the comma is merely an orthographic deviceand there is no clear phonetic correlate of the "coordinating" comma in normal, connected speech.[8]

3.4 The Paths-as-resources Approach to Coordination

Kehler et al. (1995, 1999) offer a different solution to the problem of resource sharing caused by reentrancy in f-structures, of which coordination is just one example. They propose a modification to GLUE in which *paths* through f-structures contribute resources, rather than f-structure *nodes* contributing resources, as we have been assuming and as is assumed in most work on Glue Semantics (see Dalrymple 1999, 2001 and references therein). For example, even if there is only one occurence of the subject in a VP-coordination f-structure, there are as many paths leading to the shared subject as there are heads subcategorizing for it. Thus, there will in fact be as many subject resources contributed as there are verbs requiring subjects, solving the resource sharing problem. Beside solving this problem, Kehler et al. (1995, 1999) show how their account gets correct results for the interaction of coordination with intensional verbs and with right-node raising. Dalrymple (2001:377-378) additionally notes that the Kehler et al. (1995, 1999) approach correctly forces a shared quantified subject in VP-coordination to have wide scope. For example, in *Someone laughed and cried* it is the same person doing the laughing and crying (Partee 1970).

However, there are empirical and theoretical objections to the Kehler et al. approach. First, as Dalrymple (2001:378) points out, we do not want the resource duplication offered by this approach in other cases with shared arguments at f-structure, such as in raising and possibly control (Asudeh 2000, 2002) and in unbounded dependencies involving sharing of an argument function with TOPIC or FOCUS. Second, their approach makes crucial use of the *of course* or *bang* modality (!) of linear logic. This modality turns off resource accounting for any formula it takes as its argument.[9] The problem with using this modality is that it undermines the potentially powerful explanation of natural language resource sensitivity offered by GLUE and the potential for simplifying or eliminating several principles and generalizations offered in the literature, such as Full Interpretation, the Theta Criterion, Completeness and Coherence, and possibly others (for discussion see Asudeh in progress).

By contrast, we stick to the multiplicative fragment of linear logic without bang, thus preserving a strict notion of resource sensitivity. Furthermore, the coordination-as-modification approach solves the resource sharing problem introduced by structure sharing while maintaining the usual GLUE notion of resources being contributed by f-structure nodes rather than paths. Thus, our approach does not run into resource duplication problems with raising, control, or unbounded dependencies. Lastly, our

---

[7] There are at least two possible exceptions to this. The first is Morrill's (1994) proposal to give coordinators the schematic form $(X+\backslash X/X)$, where X+ expands into one or more categories of type X. The second is the proposal by Steedman (1989:210–212). Steedman (1989) does not explicitly discuss a solution to n-ary coordination, but it is clear from Steedman (1990:fn. 9) that he means the generalization of coordination in Steedman (1990) and the syncategorematic treatment in Steedman (1989) to extend to such cases. However, syncategorematic treatments of lexically-realized elements are generally disfavoured in CG, and this is abandoned in Steedman (2000).

[8] Orthographic devices may be indicative of some linguistically-relevant factor, but this is unreliable. For example, no linguist would agree that a good test for whether something is a German noun is whether it is written capitalized or not.

[9] The bang modality is used when adding the rules for Weakening and Contraction and is therefore useful for showing relations between linear logic and classical logics.

approach also achieves correct results for wide-scope quantified subjects and for coordinations involving intensional verbs.[10]

4 Semantic Parallelism

In the next section we turn to parallelism in coordination, and in particular to examples of non-parallelism discussed by Kehler (2002). In this section we lay some groundwork by describing how semantic parallelism, cast as parallelism in GLUE derivations, can be measured.

The starting point is the observation made in section 2.1 that, for certain logical systems including linear logic, derivations have non-trivial identity criteria. These are sufficient to make proofs first class objects in logical theory, so that it is interesting not only to study *what* is proved, but also *how* it is proved.[11] These identity criteria also allow us to view normal forms derivations as canonical representations of underlying proofs.

Within a linguistic setting, this means that normal form GLUE derivations can be genuine objects in semantic theory, reifying the syntax-semantics interface to show *how* meanings are constructed, abstracting away from details of *what* meanings are constructed. This in turn enables one to compare derivations of different meanings for parallel structures.[12]

4.1 A Level of Semantic Representation

Logical formulas are traditionally not regarded as a genuine level of semantic representation (Montague 1970), as they generally have no non-trivial identity criteria other than through model theoretic semantics. For example, the two apparently distinct scopings of (34)

(34) Every man saw every woman

a. $\forall x.\, man(x) \rightarrow \forall y.\, woman(y) \rightarrow see(x, y)$

b. $\forall y.\, woman(y) \rightarrow \forall x.\, man(x) \rightarrow see(x, y)$

are model theoretically equivalent. Other than by exploiting arbitrary properties of the logical notation, there is no semantic basis for distinguishing these two formulas. Yet the urge to state generalizations over a level of semantic representation is very strong.

In this respect, GLUE contrasts with various closely related frameworks, such as Montague Grammar, GPSG, CCG, and TLG. Montague Semantics, as in Montague's own work (Montague 1973) or as in the variant espoused in GPSG, uses a level of purely syntactic representation that is systematically translated into a semantic formulae. However, there is no true level of semantic representation beyond the model theory. On the other hand we have Categorial Grammar (CCG and Type-Logical Grammar). Categorial derivations do have identity criteria that are distinct from the model theory used to interpret the semantics, but there is no separation of syntax and semantics. There is no level of syntactic representation that is distinct from the syntax of the type theory. Thus, Montague Grammar has no separate level of semantic representation and Categorial Grammar has no separate level of syntactic representation.

Glue Semantics, by contrast, posits both a level of syntactic representation and a separate level of semantic representation. There is flexibility in the choice of both levels, however. The syntactic frameworks GLUE has been defined for include Lexical Functional Grammar (Dalrymple 1999, 2001), Lexicalized Tree Adjoining Grammar (Frank and van Genabith 2001), Head-driven Phrase Structure Grammar (Asudeh and Crouch 2002), Categorial Grammar (Asudeh and Crouch 2001), and Context

---

[10] The Kehler et al. (1995, 1999) account assumes a Montogovian treatment of intensional verbs. While our account does work for such a treatment, one of us has argued in separate work (Condoravdi et al. 2001) that the Montogovian approach is flawed, based on problems with existence predicates embedded under predicates such as *prevent*. Our account also works for the concept-based treatment of intensional verbs offered by Condoravdi et al. (2001).

[11] This strand within proof theory was in fact a key motivation behind the development of linear logic (Girard et al. 1989).

[12] Asher et al. (1997, 2001) offer an alternative theory of semantic parallelism, cast in Segmented Discourse Representation Theory; however, they do not provide identity criteria for their representations (for futher discussion, see Asudeh and Crouch to appear).

Free Grammars (Asudeh and Crouch 2001). The semantic framework can be any logic for semantics that supports the lambda calculus, such as Intensional Logic, Discourse Representation Theory (Dalrymple et al. 1999c), and Underspecified DRT (Crouch and van Genabith 1999, van Genabith and Crouch 1999). GLUE derivations are similar to categorial derivations and likewise possess identity criteria distinct from model theory, but they differ in being solely semantic, since there is a separate level of syntactic representation.

4.2 Defining Semantic Parallelism

In order to show that two normal form GLUE derivations are parallel, we need to establish that there is a homomorphism (a structure preserving map) between them. Given two structured objects G and H, with structral relations R_G and R_H holding between elements of G and H respectively, a homomorphism from G to H is a mapping $\mathcal{F}$ such that

(35) If $R_G(x, y)$ then $R_H(\mathcal{F}(x), \mathcal{F}(y))$

We need to decide on two things to measure semantic parallelism: (1) what kind mapping is f and between what kinds of objects in the derivations, and (2) what kind of structural relation, R, should be preserved.

We will define f in terms of the σ projection from f-structures to semantic structures (Dalrymple 2001). Suppose that we wish to compare the derivations arising from two elements of f-structure, p and q. Then in the first instance, we want $\mathcal{F}(p_\sigma) = q_\sigma$: that is, the semantic resources for these two elements should be made parallel. And then, recursively, we want $\mathcal{F}((p\ \textsc{subj})_\sigma) = (q\ \textsc{subj})_\sigma$, $\mathcal{F}((p\ \textsc{comp})_\sigma) = (q\ \textsc{comp})_\sigma$, $\mathcal{F}((p\ \textsc{comp subj})_\sigma) = (q\ \textsc{comp subj})_\sigma$, etc., for all cases where matching f-structure paths from the roots p and q exist. In other words, the $\mathcal{F}$ mapping pairs (atomic) semantic resources for syntactically matching elements.

In cases where there are mismatched syntactic elements (i.e. when there is a path in one f-structure, but no corresponding path in the other), then the $\mathcal{F}$ mapping is undefined and filters out unmatched elements from an assessment of parallelism. This allows us to compare derivations for sentences like *Every boy saw a girl* and *Every young man saw a woman*, where the extra adjective *young* is filtered out of the comparison.[13]

The relation R needs to be defined over atomic resources / linear logic propositions, since f maps atomic resources to atomic resources. Such a relation was defined in Crouch and van Genabith (1999). In a normal form GLUE derivation, one can identify the last point of occurrence of atomic semantic resources. These indicate the points in the derivation at which the corresponding syntactic elements make their final semantic contributions. An ordering, $\prec$, over these final semantic contributions provides a high level description of the topology of the derivation; as shown in Crouch and van Genabith (1999), this ordering can be used to express scope relations.

As an example of the resource ordering relation $\prec$, re-consider the following two derivations showing the alternative scopings of (12):

(36)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[e]^2 \quad \dfrac{[g]^1 \quad g \multimap e \multimap f}{e \multimap f}}{f}}{\underline{e} \multimap f} \quad (e \multimap X) \multimap X}{f}}{\underline{g} \multimap f} \quad (g \multimap Y) \multimap Y}{\underline{f}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[e]^2 \quad \dfrac{[g]^1 \quad g \multimap e \multimap f}{e \multimap f}}{f}}{\underline{g} \multimap f} \quad (g \multimap Y) \multimap Y}{f}}{\underline{e} \multimap f} \quad (e \multimap X) \multimap X}{\underline{f}}$$

The final occurrences of the subject resource e, object resource g and sentential resource $\mathcal{F}$ are shown underlined. The tree structure of the derivations imposes a partial ordering over these final occurrences: $e \prec g \prec f$ (subject outscopes object) and $g \prec e \prec f$ (object outscopes subject) respectively.

---

[13] Here we are assuming an obvious notion of a syntactic match: subjects match subjects, objects match object, etc. Other ways of matching syntactic elements may be empirically motivated. In some cases we may want looser matches, e.g. objects can match obliques when no matching objects are present. We leave these, and many other questions, open.

In summary, to show that two derivations are semantically parallel, we need to do three things. First establish a pairing, $\mathcal{F}$, between atomic resources on the basis of their grammatical roles. Second, compute the $\prec$ ordering of these resources in the two derivations. Finally, ensure that precedence orderings coincide, so that if $a \prec b$ in derivation 1, then $\mathcal{F}(a) \prec \mathcal{F}(b)$ in derivation 2.

4.3 Scope parallelism in coordination

We now look at scope parallelism in coordination. The following example shows that the preferred reading is that in which the parallel quantifiers have parallel scopes, even if this goes against the general scope preferences of particular quantifiers:

(37) [**Context**: The animals really misbehaved last night.]
Every dog ate a bun and a cat gnawed each table leg.

The semantics of *eat* make the first conjunct plausible only with surface scope, i.e. *every* $\succ$ *a*. The parallelism between the conjuncts makes surface scope preferred in the second conjunct, too, despite the fact that *each* normally prefers wide scope.

(38)

$$\dfrac{c \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathbf{d \multimap b \multimap e} \quad [\mathbf{d}]^1}{\mathbf{\underline{b} \multimap e}} \quad \mathbf{(b \multimap X) \multimap X}}{\mathbf{e}}}{\mathbf{\underline{d} \multimap e}} \quad \mathbf{(d \multimap Y) \multimap Y}}{\mathbf{\underline{e}}} \quad e \multimap (c \multimap f) \multimap (c \multimap f)}{(c \multimap f) \multimap (c \multimap f)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathbf{c \multimap l \multimap g} \quad [\mathbf{c}]^2}{\mathbf{\underline{l} \multimap g}} \quad \mathbf{(l \multimap X) \multimap X}}{\mathbf{g}}}{\mathbf{\underline{c} \multimap g}} \quad \mathbf{(c \multimap Y) \multimap Y}}{\mathbf{\underline{g}}} \quad g \multimap (c \multimap f)}{(c \multimap f)}}{(c \multimap f)}}{f}$$

Computing parallelism proceeds as follows. First, we are comparing the derivation for clause g with that of clause e. We therefore concentrate on the sub-derivations (shown in bold) terminating at g and e. Within these, we have a pairing of subject and object resources, such that $\mathcal{F}(d) = c$ and $\mathcal{F}(b) = l$. For the two sub-derivations we have the resource ordering $b \prec d \prec e$ and $l \prec c \prec g$, i.e. $\mathcal{F}(b) \prec \mathcal{F}(d) \prec \mathcal{F}(e)$. Thus this derivation preserves semantic parallelism between the two conjuncts.

5 Violations to the Element Constraint

We have presented a theory of coordination as modification in Glue Semantics and shown how GLUE derivations can be used as a real level of semantic representation, which can be used in defining semantic parallelism. Next we will bring these two strands together and show how parallelism in GLUE can be interfaced with Kehler's (2002) theory of discourse parallelism to deal with violations of the Element Constraint (Grosu 1972, 1973), a subpart of the Coordinate Structure Constraint (Ross 1967). We conclude this section by comparing our results with the treatment of the Element Constraint in Montague Grammar, CCG, and TLG. We argue that these other theories are either too strict, allowing no exceptions to the Element Constraint, or too permissive, not capturing the Element Constraint at all.

5.1 The Coordinate Structure Constraint

The Coordinate Structure Constraint (CSC), one of Ross's (1967) island constraints, reads as follows:[14]

(39) **The Coordinate Structure Constraint**
In a coordinate structure, no conjunct may be moved, nor may any element contained in a conjunct be moved out of that conjunct.

---

[14] As the initial work on the CSC was done in Transformational Grammar, (39)–(41) make reference to movement, which does not make sense in non-transformational theories such as LFG. The constraints should be read as making appropriate restrictions on unbounded dependencies, no matter how these are dealt with.

Grosu (1972, 1973) subsequently pointed out that there are two parts to this constraint and that there are tests for distinguishing them. The parts are:

(40) **The Conjunct Constraint**: No conjunct of a coordinate structure may be moved.

(41) **The Element Constraint**: No element in a conjunct of a coordinate structure may be moved.

The key distinction between the two parts of the CSC, for our purposes, is that there are exceptions to the Element Constraint, but not to the Conjunct Constraint. Ross himself noticed certain exceptions with asymmetric coordination; Grosu points out that these are exceptions to the Element Constraint, but not to the Conjunct constraint:

- Element Constraint violations

  (42) I went to the store and bought some whiskey.
    a. This is the whiskey which I went to the store and bought.
    b. This is the store which I went to and bought some whiskey.

- No corresponding Conjunct Constraint violations

  (43) John is looking forward to going to the store and buying some whiskey.
    a. *What John is looking forward to and buying some whiskey is going to the store.
    b. *What John is looking forward to going to the store and is buying some whiskey.

The Conjunct Constraint violations should be compared to across-the-board (ATB) extraction. Ross noticed that the CSC is violable if an element is extracted from all conjuncts, or across the board:

(44) What John is looking forward to and excited about is buying some whiskey.

We will return to the Conjunct Constraint in section 5.5.

Kehler (2002) is the latest in a long line of literature that argues that the Element Constraint has principled exceptions (see Kehler (2002) for references). He notes that there are three main classes of exception to the Element Constraint and uses his theory of discourse coherence relations to explain these cases. His key insight is that *if discourse parallelism holds then the Element Constraint holds*. In other words, if the discourse coherence relation governing the conjuncts is *Parallel*, then there is either no movement out of conjuncts or there is across-the-board (i.e., parallel) movement.

Kehler's coherence relations (those relevant here) are defined as follows, with consequences for extraction possibilities as indicated:[15]

(45) **Parallel**: Infer $p(a_1, a_2, \ldots)$ from the assertion of S_1 and $p(b_1, b_2, \ldots)$ from the assertion of S_2, where for some property vector $\vec{q}$, $q_i(a_i)$ and $q_i(b_i)$ for all i.

$\boxed{\text{Any extraction must be across the board}}$

  a. *What book did John buy and read the magazine?
  b. What book did John buy and read?

(46) **Cause-Effect**

1. **Violated Expectation**: Infer P from the assertion of S_1 and Q from the assertion of S_2, where normally $P \rightarrow \neg Q$.
2. **Result**: Infer P from the assertion of S_1 and Q from the assertion of S_2, where normally $P \rightarrow Q$.

---

[15] S_1 and S_2 are respectively the first and second clauses being compared for discourse coherence; p_1 is a relation holding over a set of entities $a_1 \ldots a_n$ from S_1 and p_2 is a relation holding over a corresponding set of entities $b_1 \ldots b_n$ from S_2; q_i is a common or contrasting property for the i^{th} arguments (a_i and b_i), and the set of such properties is $\vec{q}$ (see Kehler 2002:ch. 2).

Extraction possible from initial (primary) clause

a. How much can you drink and still stay sober?

b. That's the stuff that the guys in the Caucasus drink and live to be a hundred.

c. That's the kind of firecracker that I set off and scared the neighbors.

(47) **Contiguity**

1. **Occasion (i)**
   Infer a change of state for a system of entities from S_1, inferring the final state for this system from S_2.
2. **Occasion (ii)**
   Infer a change of state for a system of entities from S_2, inferring the initial state for this system from S_1.

Extraction not necessary from "scene-setting" or "supporting" clauses

a. Here's the whiskey which I went to the store and bought.

Kehler (2002) thus gives a theory of discourse coherence that classifies and explains the exceptions to the Element Constraint systematically. In the next section we relate his discourse theory to Glue Semantics by relating proof parallelism to discourse parallelism.

5.2 Discourse Coherence and Derivational Parallelism

Kehler (2002) provides us with a theory of discourse coherence relations that explains the seeming exceptions to the Element Constraint. His theory essentially states that if discourse parallelism holds, then the Element Constraint holds. We have outlined a proof-theoretic notion of parallelism in Glue Semantics, as well as a GLUE theory of coordination as modification. Now we would like to relate Kehler's discourse theory to GLUE. Our central claim is that *if discourse parallelism holds, then proof parallelism holds*. The implication is read more usefully from right to left, and we name it the *Discourse-Proof Relation*, which sounds fancy but is just meant for us to be able to refer to it more easily:

(48) **Discourse-Proof Relation (DPR)**
If there is no proof parallelism then there is no discourse parallelism.

In other words, if parallelism does not hold in the GLUE proof for a coordination, the discourse relation cannot be parallelism. If the discourse relation is not parallelism, then it must be some other discourse relation; *if another discourse relation is compatible with the coordination, then the coordination is licensed, otherwise it is not licensed*. This allows the following understanding of the element constraint:

(49) **The Element Constraint**: when an element is extracted from some but not all conjuncts in a coordinate structure, there is no proof parallelism, and therefore no discourse parallelism.

With (48) and (49) in hand, we can now look at how our theory handles exceptions to the Element Constraint by appealing to Kehler's (2002) theory where proof parallelism does not hold. We need to consider examples like (45a), where the Element Constraint is violated and ungrammaticality results, the across-the-board case (45b), and the cases in (46) and (47) where the Element Constraint is violated but no ungrammaticality results.

Before turning to the exposition of the relevant examples we need to make clear one important caveat. Throughout this paper we have been assuming an LFG syntax. In the standard LFG treatment of unbounded dependencies and coordination (Kaplan and Maxwell 1988, Dalrymple 2001), a strict version of the Element Constraint is upheld, allowing no unbounded dependencies that terminate in some but not all of the conjuncts of a coordination. This is due to the way that inside-out functional uncertainties (which are used to handle unbounded dependencies) interact with sets. Grammatical functions are standardly defined as distributive features (Dalrymple and Kaplan 2000, Dalrymple 2001) and an inside-out path written in terms of grammatical functions will therefore distribute to all members of a coordination set. One solution is to make grammatical functions nondistributive features instead. We cannot explore the consequences of such a solution here. However, as we stressed above, GLUE can be coupled with a variety of syntactic frameworks and as such this contingent fact about LFG does not impact our GLUE account. The account could be paired with another syntactic theory instead; the only desideratum would be that the syntax have coordination rules like (33), which capture the Conjunct Constraint, as we will see in section 5.5 below.

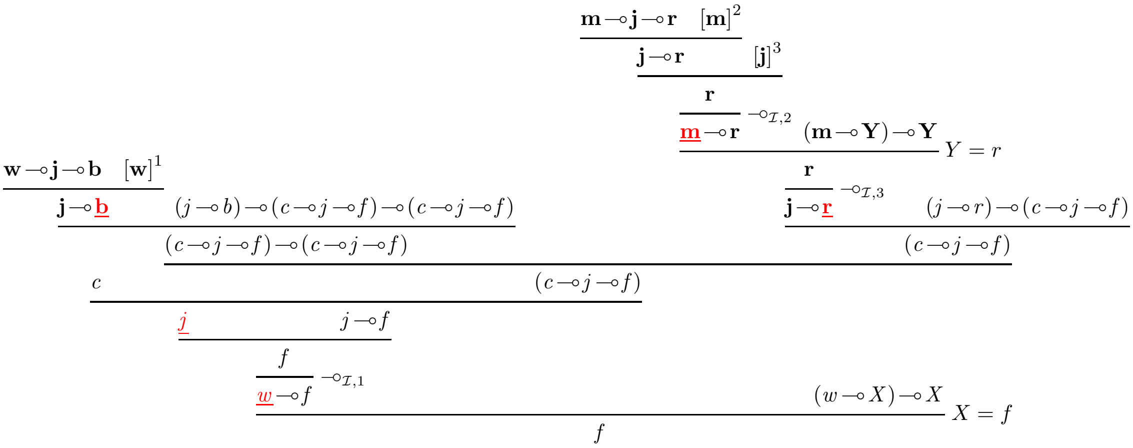
5.3 The Element Constraint and Derivational Parallelism

5.3.1 Non-Parallel Extraction

Consider (45a), repeated as (50) below, which violates the Element Constraint, since the object of *buy* has been extracted, but the object of *read* has not. The (perfectly valid) GLUE derivation for this example is shown in (51), with the resource f corresponding to the coordination and all other resources named mnemonically as usual.

(50) *What book did John buy and read the magazine?

(51)



The sub-derivations for the two conjunct resources, b and r, are shown in bold, and terminate at the conclusions $j \multimap b$ and $j \multimap r$ respectively. The mapping of subjects to subjects, objects to objects, etc., gives the following pairings of resources:

(52) $\mathcal{F}(b) = r,\ \mathcal{F}(j) = j,\ \mathcal{F}(w) = m$

The final occurrences of each of these resources are shown in red and underlined; these induce the non-parallel orderings

(53) $b \prec j \prec w$
$m \prec r \prec j$ (i.e. not $\mathcal{F}(b) \prec \mathcal{F}(j) \prec \mathcal{F}(w)$)

Thus, although the GLUE derivation is logically valid, it does not preserve parallelism between the conjuncts.

According to the DPR, in (48) above, since there is no proof parallelism there is no discourse parallelism. Therefore, the discourse relation cannot be *Parallel*. However, none of the other discourse relations in (46) and (47) are compatible with (50) either. The *Cause-Effect* relations *Violated Expectation* and *Result* cannot relate the conjuncts in (50), since there is no implicational relationship between the assertion of *John bought x* and *John read the magazine*. The two different kinds of *Contiguity* relation, *Occasion (i)* and *Occasion (ii)* also do not hold: *John bought x* does not set the scene for or support *John read the magazine* (more specifically, there is no change of state from S_1 to S_2, or vice versa).

5.3.2 Across-the-Board Extraction

The situation for (50) contrasts with the across-the-board extraction case (45b) and with the cases in (46a–c) and (47a). In the latter cases, there is no proof parallelism either, and therefore no discourse parallelism, but the relevant discourse relation licenses each case. As for across-the-board extraction, this does lead to derivational parallelism, as shown in proof (55) for example (54):

(54) What book did John buy and read?

(55)

$$\dfrac{\dfrac{\dfrac{\underline{j} \qquad \dfrac{[w]^1 \qquad \dfrac{c \qquad \dfrac{\dfrac{\mathbf{w}\multimap\mathbf{j}\multimap\underline{\mathbf{b}} \qquad (w\multimap j\multimap b)\multimap(c\multimap w\multimap j\multimap f)\multimap(c\multimap w\multimap j\multimap f)}{(c\multimap w\multimap j\multimap f)\multimap(c\multimap w\multimap j\multimap f)} \qquad \dfrac{\mathbf{w}\multimap\mathbf{j}\multimap\underline{\mathbf{r}} \qquad (w\multimap j\multimap r)\multimap(c\multimap w\multimap j\multimap f)}{c\multimap w\multimap j\multimap f}}{c\multimap w\multimap j\multimap f}}{w\multimap j\multimap f}}{j\multimap f}}{f}}{\underline{w}\multimap f}\,{\multimap_{\mathcal{I},1}} \qquad (w\multimap X)\multimap X}{f}\; X = f$$

For this derivation we have the resource mappings and the parallel orderings

(56) $\mathcal{F}(b) = r,\ \ \mathcal{F}(j) = j,\ \ \mathcal{F}(w) = w$

(57) $b \prec j \prec w,\quad r \prec j \prec w$

Across the board extraction preserves derivational parallelism.

5.4 Syntactic and Semantic Parallelism

Lack of discourse parallelism as per Kehler's (2002) theory is indeed reflected by lack of GLUE derivational parallelism. However, one could arguably also point to a more obvious lack of syntactic parallelism in these cases. That is, there will be a difference between the syntactic structure corresponding to a conjunct with no element extracted and the structure for a conjunct with an extracted element. In LFG terms, for example, the extracted GF will be shared with a discourse function, TOPIC or FOCUS, while the parallel unextracted GF will not.

We feel that there are still compelling reasons for preferring the GLUE parallelism account of the Element Constraint to an alternative account based on purely syntactic parallelism. First, GLUE is framework-independent and can be combined with a variety of syntactic frameworks, as noted above. This makes the GLUE treatment more general than a syntactic treatment cast in some specific framework. For example, the grammatical functions used in LFG do not have clear correspondences in most other syntactic frameworks. Second, proof parallelism in GLUE generalizes to scope parallelism, as we showed above, and to ellipsis parallelism, as we have shown elsewhere (Asudeh and Crouch to appear). Thus, the GLUE parallelism theory has potential as a general theory of parallelism, unlike a theory of syntactic parallelism devised just for coordination. Third, the close relationship between GLUE, Combinatory Categorial Grammar, and Type-Logical Grammar allows us to compare the GLUE account of the Element Constraint to the standing of this constraint in these other theories. We turn to this comparison now.

5.5 The Goldilocks Effect

We have noted that a desirable situation is one where the Conjunct Constraint holds generally, but where the Element Constraint is allowed certain systematic exceptions governed by discourse parallelism. In our theory the Conjunct Constraint follows from the syntactic rule for forming coordinate structures, shown for VP-coordination in (21) above. The general form of the syntactic portion of the rule, which was shown in (33) above, is repeated here:

(58)
$$\mathrm{X} \rightarrow \quad \underset{\downarrow\ \in\ \uparrow}{\mathrm{X}^{+}} \qquad \underset{\uparrow=\downarrow}{\textsc{conj}} \qquad \underset{\downarrow\ \in\ \uparrow}{\mathrm{X}}$$

As the category-specific rules for coordination will each be coordinating one or more conjuncts on the left with a conjunct on the right, it does not license structures that are missing conjuncts. This is not new; a similar rule is also a feature of Generalized Phrase Structure Grammar (Gazdar et al. 1985) and Combinatory Categorial Grammar (Steedman 1985), as well as the earlier account of Dougherty (1970). The rule is also fairly neutral with respect to syntactic formalism: most syntactic approaches could readily accommodate such a rule.

However, the Element Constraint does not *necessarily* follow from this rule. If the categories that the rule conjoins register extraction (as in GPSG, with slashes, or CCG, with functors), then the strict version of the Element Constraint does follow, because we would be coordinating a slashed category, for example, with an unslashed one. This is undesirable, since it is too strong for the Element Constraint to hold in general; exceptions must be allowed.

By contrast, the Element Constraint does not hold at all in Type-Logical Grammar, without further stipulation. If a conjunct has an extracted element, it is always possible to reduce the categories of all the other conjuncts in the coordination to the category of the conjunct with the missing element. This is done by hypothetical reasoning. The assumptions can be discharged after the coordination has been carried out. In fact, the same thing is true of our GLUE theory of coordination as modification if it is not related to a discourse theory like Kehler's (2002). Given the proof-theoretic similarity between TLG and GLUE, it is quite feasible that our parallelism account could be ported to TLG. However, as it stands TLG does not capture the Element Constraint at all. This is too weak, as not all exceptions to the Element Constraint result in grammatical outputs and there are generalizations about the grammatical exceptions that would be missed by allowing the Element Constraint to fail in general.

Thus, we have what we call the Goldilocks Effect. One brand of Categorial Grammar (CCG), as well as Montague Grammar and GPSG, is too strong: it allows no exceptions to the Element Constraint. The other brand of Categorial Grammar (TLG) is too weak: all exceptions to the Element Constraint are permitted. Perhaps Glue Semantics is just right.

6 Conclusion

We set out do three things in this paper. The first was to develop the GLUE account of coordination as modification, which was accomplished in section 3. The resulting theory not only preserves resource sensitivity, it is also able to deal with n-ary coordination and does not overgenerate in the manner of certain Categorial Grammar treatments, since GLUE preserves a notion of constituency through its pairing with a syntactic theory. The second goal was to present a theory of proof parallelism and to show how this theory extends to coordination; this was presented in section 4. The third goal, which we ended with in section 5, was to interface the theory of proof parallelism with Kehler's (2002) theory of discourse coherence, in order to capture the Element Constraint in general while allowing a systematic class of exceptions.

References

Asher, Nicholas, Daniel Hardt, and Joan Busquets (1997). Discourse parallelism, scope, and ellipsis. In Aaron Lawson (ed.), *Proceedings of SALT VII*.

Asher, Nicholas, Daniel Hardt, and Joan Busquets (2001). Discoure parallelism, ellipsis, and ambiguity. *Journal of Semantics, 18.*

Asudeh, Ash (2000). Functional identity and resource-sensitivity in control. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the LFG00 conference*, Stanford, CA. CSLI Publications.

Asudeh, Ash (2002). A resource-sensitive semantics for equi and raising. In David Beaver, Stefan Kaufmann, Brady Clark, and Luis Casillas (eds.), *The construction of meaning*. Stanford, CA: CSLI Publications.

Asudeh, Ash (in progress). *Resumption as resource management*. Ph.D. thesis, Stanford University, Stanford, CA.

Asudeh, Ash, and Richard Crouch (2001). Glue semantics: A general theory of meaning composition. Talk given at Stanford Semantics Fest 2, March 16, 2001.

Asudeh, Ash, and Richard Crouch (2002). Glue semantics for HPSG. In Frank van Eynde, Lars Hellan, and Dorothee Beermann (eds.), *Proceedings of the 8th international HPSG conference*, Stanford, CA. CSLI Publications.

Asudeh, Ash, and Richard Crouch (to appear). Derivational parallelism and ellipsis parallelism. In Line Mikkelsen and Christopher Potts (eds.), *WCCFL 21 proceedings*, (pp. 1–14). Somerville, MA: Cascadilla Press.

Carpenter, Robert (1997). *Type-logical semantics*. Cambridge, MA: MIT Press.

Condoravdi, Cleo, Richard Crouch, Martin van den Berg, John Everett, Valeria de Paiva, Reinhard Stolle, and Daniel Bobrow (2001). Preventing existence. In *Proceedings of Formal Ontologies in Information Systems (FOIS)*, Maine.

Crouch, Richard (1999). Ellipsis and glue languages. In Shalom Lappin and Elabbas Benmamoun (eds.), *Fragments: Studies in ellipsis and gapping*. Oxford: Oxford University Press.

Crouch, Richard, and Josef van Genabith (1999). Context change, underspecification, and the structure of glue language derivations. In Dalrymple (1999), (pp. 117–189).

Dalrymple, Mary (ed.) (1999). *Semantics and syntax in Lexical Functional Grammar: The resource logic approach*. Cambridge, MA: MIT Press.

Dalrymple, Mary (2001). *Lexical Functional Grammar*. San Diego, CA: Academic Press.

Dalrymple, Mary, Vaneet Gupta, John Lamping, and Vijay Saraswat (1999a). Relating resource-based semantics to categorial semantics. In Dalrymple (1999), (pp. 261–280).

Dalrymple, Mary, and Ron Kaplan (2000). Feature indeterminacy and feature resolution in description-based syntax. *Language*, *74*, 759–798.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell, III, and Annie Zaenen (eds.) (1995). *Formal issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications.

Dalrymple, Mary, John Lamping, Fernando Pereira, and Vijay Saraswat (1999b). Overview and introduction. In Dalrymple (1999), (pp. 1–38).

Dalrymple, Mary, John Lamping, Fernando Pereira, and Vijay Saraswat (1999c). Quantification, anaphora, and intensionality. In Dalrymple (1999), (pp. 39–89).

Dalrymple, Mary, John Lamping, and Vijay Saraswat (1993). LFG semantics via constraints. In *Proceedings of the sixth meeting of the European ACL*, (pp. 97–105), University of Utrecht. European Chapter of the Association for Computational Linguistics.

Dougherty, Ray (1970). A grammar of coordinate conjoined structures. *Language*, *46*, 850–898.

Emms, Martin (1990). Polymorphic quantifiers. In *Studies in Categorial Grammar*, no. 5 in Edinburgh Working Papers in Cognitive Science, (pp. 65–111). Edinburgh: Centre for Cognitive Science.

Frank, Anette, and Josef van Genabith (2001). LL-based semantics construction for LTAG — and what it teaches us about the relation between LFG and LTAG. In *Proceedings of the LFG01 conference*. CSLI Publications.

Gazdar, Gerald (1980). A cross-categorial semantics for coordination. *Linguistics and Philosophy, 3,* 407–410.

Gazdar, Gerald (1981). Unbounded dependencies and coordinate structure. *Linguistic Inquiry, 12,* 155–184.

Gazdar, Gerald (1982). Phrase structure grammar. In Pauline Jacobson and Geoffrey K. Pullum (eds.), *The nature of syntactic representation*, vol. 15 of *Synthese Language Library*, (pp. 131–186). Dordrecht: Reidel.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag (1985). *Generalized phrase structure grammar*. Cambridge, MA: Harvard University Press.

Girard, Jean-Yves (1987). Linear logic. *Theoretical Computer Science, 50,* 1–102.

Girard, Jean-Yves, Paul Taylor, and Yves Lafont (1989). *Proofs and types*. Cambridge: Cambridge University Press.

Grosu, Alexander (1972). *The strategic content of island constraints*. Ph.D. thesis, Ohio State University, Columbus, OH. Reprinted as Working Papers in Linguistics no. 13, OSU.

Grosu, Alexander (1973). On the nonunitary nature of the Coordinate Structure Constraint. *Linguistic Inquiry, 4,* 88–92.

Howard, William A. (1980). The formulae-as-types notion of construction. In Jonathan P. Seldin and J. Roger Hindley (eds.), *To H.B. Curry: Essays on combinatory logic, lambda calculus and formalism,* (pp. 479–490). London: Academic press. Often cited as Howard (1969).

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

Kaplan, Ronald M., and John T. Maxwell, III (1988). Constituent coordination in Lexical-Functional Grammar. In *Proceedings of COLING-88*, vol. 1, Budapest. Reprinted in Dalrymple et al. (1995:199–214).

Keenan, Edward L., and Leonard M. Faltz (1985). *Boolean semantics for natural language*. Dordrecht: Reidel.

Kehler, Andrew (2002). *Coherence, reference, and the theory of grammar*. Stanford, CA: CSLI Publications.

Kehler, Andrew, Mary Dalrymple, John Lamping, and Vijay Saraswat (1995). The semantics of resource-sharing in Lexical-Functional Grammar. In *Proceedings of the 1995 meeting of the European Chapter of the Association for Computational Linguistics*, Dublin, Ireland.

Kehler, Andrew, Mary Dalrymple, John Lamping, and Vijay Saraswat (1999). Resource sharing in glue language semantics. In Dalrymple (1999), (pp. 191–208).

Maxwell, John T., III, and Christopher D. Manning (1996). A theory of non-constituent coordination based on finite-state rules. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the LFG96 conference*, Stanford, CA. CSLI Publications.

Montague, Richard (1970). English as a formal language. In Bruno Visentini et al. (eds.), *Linguaggi nella societá e nella tecnica,* (pp. 189–224). Milan: Edizioni di Communità. Reprinted in Montague (1974:188–221).

Montague, Richard (1973). The proper treatment of quantification in ordinary English. In Jakko Hintikka, Julian Moravcsik, and Patrick Suppes (eds.), *Approaches to language*, (pp. 221–242). Dordrecht: D. Reidel. Reprinted in Montague (1974:247–270).

Montague, Richard (1974). *Formal philosophy: Selected papers of Richard Montague*. New Haven: Yale University Press. Edited and with an introduction by Richmond H. Thomason.

Morrill, Glyn (1994). *Type Logical Grammar*. Dordrecht: Kluwer.

Partee, Barbara (1970). Negation, conjunction, and quantifiers: Syntax vs. semantics. *Foundations of Language*, *6*, 153–165.

Partee, Barbara, and Mats Rooth (1983). Generalized conjunction and type ambiguity. In Rainer Bäuerle, Christoph Schwarze, and Arnim von Stechow (eds.), *Meaning, use, and interpretation of language*. Berlin: Walter De Gruyter.

Prawitz, Dag (1965). *Natural deduction: A proof-theoretical study*. Stockholm: Almquist and Wiksell.

Ross, John R. (1967). *Constraints on variables in syntax*. Ph.D. thesis, Massachusetts Institute of Technology.

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

Steedman, Mark (1989). Constituency and coordination in a combinatory grammar. In Mark Baltin and Anthony Kroch (eds.), *Alternative conceptions of phrase structure*, (pp. 201–231). Chicago: University of Chicago Press.

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

Steedman, Mark (2000). *The syntactic process*. Cambridge, MA: MIT Press.

van Genabith, Josef, and Richard Crouch (1999). How to glue a donkey to an f-structure. In Harry Bunt and Reinhard Muskens (eds.), *Computing meaning*, vol. 1. Dordrecht: Kluwer.