Derivational Parallelism and Ellipsis Parallelism

Ash Asudeh and Richard Crouch
Stanford University and Palo Alto Research Center

A central topic in work on ellipsis is the parallelism between missing VPs and their antecedents. There is evidence that parallelism is conditioned both syntactically and semantically. This has resulted in complex hybrid principles that regulate both syntactic structures and their interpretations (e.g., Fox 1995, 2000). We present an alternative, unified account based on parallelism of semantic derivations (i.e., proofs) in Glue Semantics (Dalrymple 1999, 2001). Our proposal captures the insights of hybrid accounts, but is conceptually simpler and empirically superior. It is also theoretically important, as it constitutes an implicit argument for a level of semantic representation, favoring indirect translation (Montague 1973) over direct model-theoretic interpretation (Montague 1970).

The empirical focus of the paper is scope parallelism in ellipsis in general (Sag 1976, Williams 1977), and the problem of "Canadian flag" sentences in particular (Hirschbühler 1982). Sag (1976) noticed that a scopally ambiguous sentence seems to be disambiguated to surface scope when it is the source sentence for VP-ellipsis, as in (1).

(1) A rhino sat in every corner. Babar did, too.

Hirschbühler (1982) subsequently noticed that if the subject of the target ellipsis sentence is also quantificational, as in (2), both inverse and surface scope are possible for the source and target sentences.

(2) A rhino sat in every corner. An elephant did, too.

However, the scope of source and target must be parallel: either both surface or both inverse. These issues have been resurrected in important recent work by Fox (1995, 2000), but his analysis has faced conceptual, theoretical, and especially empirical challenges in recent work (Asher et al. 1997, 2001, Jacobson 1998, Johnson and Lappin 1997, 1999).

1. Fox's Theory

Fox (1995, 2000) proposes the *Ellipsis Scope Generalization* (ESG):

(3) A sentence S will disambiguate its syntactic image S' (in favor of surface scope), whenever S is semantically equivalent under surface and inverse scope (i.e, whenever S is scopally uninformative). (Fox 2000:34)

Notice that the ESG makes predictions that are in line with Sag's and Hirschbühler's observations. On the assumption that proper names are not scopally ambiguous, the ellipsis target in (1) is semantically equivalent under

 WCCFL 21 Proceedings, ed. L. Mikkelsen and C. Potts, pp. 1–1. Somerville, MA: Cascadilla Press.

surface and inverse scope, and the ellipsis source is predicted to have only surface scope. But in (2) the target is not semantically equivalent under surface and inverse scope, and the source is predicted to allow inverse scope.

There are several counterexamples to the ESG in the literature (Lappin 1993, Shieber et al. 1996). In reaction to the counterexamples, Fox (1995) proposes that the stage-/individual-level distinction between predicates (Kratzer 1989, 1995) is crucial, and that the ESG only holds for individual-level predicates. However, Asher et al. (1997, 2001), Jacobson (1998), and Johnson and Lappin (1997, 1999) showed that there are counterexamples even with individual-level predicates.

Furthermore, it is easy to construct pairs of otherwise identical sentences that alternate between a stage- and individual-level predicate, such that both sentences allow inverse scope readings. Consider a context in which the speaker is talking about a prestigious annual dinner; at the latest dinner, each table was served by not just one, but several waiters and in fact the chef made the rounds of the tables, too. In addition, this dinner has been held at the same restaurant several times. In such a context, both (4) and (5) easily allow inverse scope readings:

(4) Several waiters served at every table. Chef Pierre did too.

(5) Several waiters knew every customer. Chef Pierre did too.

Serve is a stage-level predicate and *know* is an individual-level predicate.

2. Need for a Preference Mechanism

The counterexamples to Fox's theory show that it rules out attested readings; in other words, the theory is too narrow in its coverage and too strong in its predictions. But other theories simply allowing both surface and inverse scope are too weak. They have little so say about the observed fact that, while not impossible, it is hard to give an inverse scope reading to an ellipsis source if the parallel element in the target is a name.

What is required is some principled way of calculating preferences for interpretation. The surface scope readings for sentences like (1) are preferred over inverse scope readings, but inverse scope readings are nonetheless generally available.

In the following sections we will account for parallelism possibilities and preferences within the framework of Glue Semantics (Dalrymple 1999, 2001). In this framework, syntactic analysis produces a collection of lexical semantic premises. These premises are consumed in a linear logic derivation to construct the semantic interpretation of the sentence. We will argue that the linear logic derivation provides an appropriate level on which to define parallelism preferences. Our account shares certain insights with that of Asher et al. (1997, 2001), but there are also important differences, which we briefly discuss in section 5.4.

3. Semantic Derivations as Linear Logic Proofs

The framework we assume, Glue Semantics (GLUE), is a modular theory of the syntax-semantics interface. A variety of syntactic frameworks have been coupled with GLUE — including Lexical Functional Grammar (Dalrymple 1999, 2001), Lexicalized Tree Adjoining Grammar (Frank and van Genabith 2001), and HPSG (Asudeh and Crouch 2002) — and the meaning language can be any logic for semantics with lambda terms; for example there are existing versions of Glue Semantics using Montague's IL, DRT, and UDRT (Dalrymple et al. 1999b, Crouch and van Genabith 1999, van Genabith and Crouch 1999).

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 colon is an uninterpreted pairing symbol. The constructors are used as premises in a linear logic proof that consumes the lexical premises to produce a sentential meaning. Note that linear logic (Girard 1987) is unlike traditional logic in being resource sensitive: premises are literally used up in producing conclusions. 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 (Howard 1980) ensures that the meaning term constructed for the sentence is determined by the structure of the linear logic proof (Dalrymple et al. 1999b). It does so by pairing each linear logic proof rule with an operation in the meaning language. The two (Natural Deduction) proof rules we will be concerned with here are Implication Elimination (modus ponens) and Implication Introduction (hypothetical reasoning), which correspond respectively to function application and lambda abstraction in the meaning language (' $\multimap$ ' is linear logic implication):

(6) **Implication Elimination** **Implication Introduction**

$$\frac{P : A \multimap B \qquad Q : A}{P(Q) : B} \multimap_{\mathcal{E}}$$

$$\frac{\begin{array}{c}[Q : A]^i \\ \vdots \\ P : B\end{array}}{\lambda Q.P : A \multimap B} \multimap_{\mathcal{I},i}$$

The following simple proof of $A \multimap B \vdash A \multimap B$ shows the interaction of the proof rules and the meaning language:

(7)
$$\dfrac{\dfrac{[Q : A]^1 \qquad P : A \multimap B}{P(Q) : B} \multimap_{\mathcal{E}} \text{ (function application)}}{\lambda Q.P(Q) : A \multimap B} \multimap_{\mathcal{I},1} \text{ (lambda abstraction)}$$

In the first step, $Q : A$ is assumed (indicated by square brackets) and the assumption is flagged with the superscript *1*. We take this assumption and combine it with our one premise $A \multimap B$ by elimination, which corresponds to function application in the meaning language. The assumption is discharged in the second step, re-introducing the assumed linear logic atom A. On the meaning language side this corresponds to abstracting over the associated variable, Q.

Although the linear logic proofs are our main concern in this paper, it is useful to present a couple of very simple examples to illustrate how GLUE works (see Dalrymple (1999) for a fuller introduction). We will assume a simplified LFG syntax (Kaplan and Bresnan 1982, Bresnan 2001) and a very generic predicate calculus, for the sake of exposition. Consider (8) and its lexical items (9):

(8) Gonzo smokes.

(9)	*Gonzo* N	*smokes* V
	$(\uparrow \textsc{pred}) = \text{`Gonzo'}$	$(\uparrow \textsc{pred}) = \text{`smoke'}$
	$gonzo : \uparrow_{\sigma_e}$	$smoke : (\uparrow \textsc{subj})_{\sigma_e} \multimap \uparrow_{\sigma_t}$

The $\uparrow$ meta-variable in the lexical entries refers (roughly speaking) to the phrase of which the word is the lexical head. 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 (8) both constructs the f(unctional)-structure (10) and instantiates the lexical entries so that the $\uparrow$ meta-variables refer to nodes within (10), instantiating the lexical premises as in (11) (bearing in mind that $(s \ \textsc{subj}) = g$).

$$(10) \quad s\begin{bmatrix} \textsc{pred} & \text{`smoke'} \\ \textsc{subj} & g\begin{bmatrix} \textsc{pred} & \text{`Gonzo'} \end{bmatrix} \end{bmatrix}$$

(11) $gonzo : g_{\sigma_e}$
$smoke : g_{\sigma_e} \multimap s_{\sigma_t}$

The formulas in (11) 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 two premises combine through one instance of introduction elimination, which is function application in the meaning language:

$$(12) \quad \frac{gonzo : g_e \qquad smoke : g_e \multimap s_t}{smoke(gonzo) : s_t} \multimap_{\mathcal{E}}$$

The second simple 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:[1]

(13) Most presidents speak at least one language.

(14)
$$s\left[\begin{array}{ll} \text{PRED} & \text{'speak'} \\ \text{SUBJ} & p\left[\begin{array}{ll}\text{PRED} & \text{'presidents'}\\ \text{SPEC} & \left[\begin{array}{ll}\text{PRED} & \text{'most'}\end{array}\right]\end{array}\right] \\ \text{OBJ} & l\left[\begin{array}{ll}\text{PRED} & \text{'language'}\\ \text{SPEC} & \left[\begin{array}{ll}\text{PRED} & \text{'at least one'}\end{array}\right]\end{array}\right]\end{array}\right]$$

(15)
$$\begin{array}{l} \lambda P.most(president,\ P) : (p_e \multimap X_t) \multimap X_t \\ \lambda u \lambda v.speak(v,u) : l_e \multimap p_e \multimap s_t \\ \lambda Q.\text{ALO}(language,\ Q) : (l_e \multimap Y_t) \multimap Y_t \end{array}$$

The meaning terms $\lambda P.most(president,\ P)$ and $\lambda Q.\text{ALO}(language,\ Q)$ are standard generalized quantifier expressions, which we will henceforth abbreviate as MP and ALOL. 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, denote arbitrary type t atomic resources that the quantifiers could take as their scope.[2] Essentially, the two quantifiers can apply to any type t clause that depends on the meaning of the subject p_e or the object l_e, and discharge this dependency by scoping the quantitifier.

From the three premises in (15), there are two distinct derivations s_t. Both have the same initial derivation (16), producing the semantic resource s_t dependent on both p_e and l_e. The derivations then fork, depending on which of these dependencies are discharged first via scoping a quantifier. ((17) for surface scope and (18) for inverse scope).

(16)
$$\dfrac{[x : p]^2 \qquad \dfrac{[y : l]^1 \qquad \lambda u \lambda v.speak(v,u) : l \multimap p \multimap s}{\lambda v.speak(v,y) : p \multimap s}\multimap_{\mathcal{E}}}{speak(x,y) : s}\multimap_{\mathcal{E}}$$

(17)
$$\dfrac{\text{MP} : (p \multimap X) \multimap X \qquad \dfrac{\dfrac{\text{ALOL} : (l \multimap Y) \multimap Y \qquad \dfrac{speak(x,y) : s}{\lambda y.speak(x,y) : l \multimap s}\multimap_{\mathcal{I},1}}{\text{ALOL}(\lambda y.speak(x,y)) : s}\multimap_{\mathcal{E}\,Y=s}}{\lambda x.\text{ALOL}(\lambda y.speak(x,y)) : p \multimap s}\multimap_{\mathcal{I},2}}{\text{MP}(\lambda x.\text{ALOL}(\lambda y.speak(x,y))) : s}\multimap_{\mathcal{E}\,X=s}$$

1. We have combined the quantifiers with their restrictions for simplicity's sake.
2. In Dalrymple et al. (1999a) these variable resources are explicitly quantified — we omit this detail here for the sake of readability.

(18)
$$\dfrac{\textsc{alol} : (l \multimap Y) \multimap Y \qquad \dfrac{\dfrac{\textsc{mp} : (p \multimap X) \multimap X \qquad \dfrac{\mathit{speak}(x,y) : s}{\lambda x.\mathit{speak}(x,y) : p \multimap s}\multimap_{\mathcal{I},2}}{\textsc{mp}(\lambda x.\mathit{speak}(x,y)) : s}\multimap_{\mathcal{E}\,X=s}}{\lambda y.\textsc{mp}(\lambda x.\mathit{speak}(x,y)) : l \multimap s}\multimap_{\mathcal{I},1}}{\textsc{alol}(\lambda y.\textsc{mp}(\lambda x.\mathit{speak}(x,y))) : s}\multimap_{\mathcal{E}\,Y=s}$$

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, and no special assumptions need to be made about the meaning terms.

4. Identity Criteria for Proofs

There is a strong tradition within proof theory, (e.g. Girard 1987), that views logical proofs as interesting objects in their own right. It is not only interesting to ask whether conclusion c follows from premises P, but also to ask how it follows, and whether it can follow in more than one way. We have just seen this perspective at work in connection with quantifier scope ambiguity. It is not only interesting to prove that a semantic interpretation can be derived from the premises in (15). How it is derived is also important, since different derivations construct different meanings.

For proofs to be regarded as first class objects, they need to come equipped with interesting, non-trivial identity criteria. These identity criteria need to guard against the following problem: although we may be interested in proofs, we do not have direct access to the underlying proof objects. All we have are marks on paper, i.e. derivations, showing how proofs can be represented in a particular formal system such as natural deduction. What is required is a way of showing how syntactically distinct derivations can be mapped onto a canonical representation corresponding to the underlying proof.

For certain logical systems, including the fragment of linear logic used in Glue Semantics, there are two convergent ways of stating the required identity criteria: via the Curry-Howard isomorphism (Howard 1980), and via proof normalization (cut elimination) (Prawitz 1965). To illustrate both of these, consider the question: how many distinct ways are there of deriving B from $A \multimap B$ and A? Here are two possible derivations, both corresponding to the same underlying proof:

(19) (a) $\dfrac{A \qquad A \multimap B}{B}\multimap_{\mathcal{E}}$ (b) $\dfrac{A \qquad \dfrac{\dfrac{[A]^1 \qquad A \multimap B}{B}\multimap_{\mathcal{E}}}{A \multimap B}\multimap_{\mathcal{I},1}}{B}\multimap_{\mathcal{E}}$

The second derivation introduces a "detour": it eliminates the implication $A \multimap B$ only to immediately re-introduce exactly the same implication, and then re-eliminate it. The first derivation avoids this detour. Proof normalization provides a set of rules for expunging certain detours from a derivation. The rule for expunging a β-detour of an introduction immediately followed by an elimination is shown in (20). This rule normalizes (19b) to (19a).

(20)
$$\dfrac{A \qquad \dfrac{\begin{array}{c}[A]^i\\ \vdots\\ B\end{array}}{A \multimap B}\multimap_{\mathcal{I},i}}{B} \quad \stackrel{\beta}{\Longrightarrow} \quad \begin{array}{c}A\\ \vdots\\ B\end{array}$$

We will focus on derivations with expunged β-detours, though η-detours of eliminations immediately followed by introductions will be permitted.

The motivation for the nomenclature β- and η-detour becomes apparent if the formulas in the two derivations above are paired with proof terms via the Curry-Howard isomorphism:

(21) (a)
$$\dfrac{a : A \quad p : A \multimap B}{p(a) : B}\multimap_{\mathcal{E}}$$
(b)
$$\dfrac{a : A \qquad \dfrac{\dfrac{[x : A]^1 \qquad p : A \multimap B}{p(x) : B}\multimap_{\mathcal{E}}}{\lambda x.p(x) : A \multimap B}\multimap_{\mathcal{I},1}}{(\lambda x.p(x))(a) : B}\multimap_{\mathcal{E}}$$

Note that the two resulting λ-terms, $p(a)$ and $(\lambda x.p(x))(a)$, are equivalent via β- and η-reduction. The λ-terms motivate the following terminology. The (a) derivations are in β-normal form, and the (b) derivations are in η-long normal form. In η-long normal form, all implicational formulas are paired with explicit λ-abstractions, making their functional nature overt. Although it appears more verbose, η-long normal form is widely used in automated proof search (Jay and Ghani 1995) and also corresponds to derivations produced via proof nets (Girard 1987), a special purpose proof method for linear logic.

Proof normalization and the Curry-Howard isomorphism converge in that normalizing a proof corresponds to λ-reducing its proof term. It is the existence of these normalization / reduction techniques that lead to interesting identity criteria for proofs. We can view normal form derivations (for a given choice of normal form) as standing for the underlying proofs.

4.1. Identity Criteria for Semantic Compositions

Within Glue Semantics, semantic composition is modeled as linear logic proof. We can therefore take over the identity criteria for proofs to stand as identity criteria for semantic compositions. As a result, we can regard semantic compositions as first class objects in linguistic theory, separate from but linked to syntactic structures on the one hand, and to model theoretic interpretations of sentences on the other hand. The modularity of GLUE with regard both to the syntactic framework (e.g. LFG, HPSG, TAG), and to the meaning language (e.g. IL, DRT) indicates that GLUE proofs are a way of reifying the syntax-semantics interface.

Positing composition as a level of semantic representation goes against the direct model-theoretic interpretation approach of Montague (1970). This says that nothing should sit between syntactic structure and model-theoretic interpretation. Composition as a level of representation also goes against the perspective of much work in categorial and type-logical grammar, which otherwise shares with GLUE the use of the Curry-Howard isomorphism. This claims that there is no real level of syntactic structure (beyond categorial derivations). The GLUE account allows for both syntactic structure and model-theoretic interpretation, with a significant level of representation in between. We believe that this level of representation is invaluable in accounting for parallelism phenomena in ellipsis.

The basic intuition lying behind most accounts of ellipsis is that the source material is reconstructed in some way so as to make the minimal changes necessary to incorporate the explicit parallel material in the target. Intuitions differ, however, as to whether ellipsis interpretation is conditioned by syntax (e.g. Fiengo and May 1994, Fox 1995, 2000) or by semantics (e.g. Dalrymple et al. 1991, Hardt 1993, 1999, Shieber et al. 1996, Asher et al. 1997, 2001). There is evidence that both levels are significant (Lappin 1996, Merchant 2001). Framing ellipsis resolution at the syntax-semantics interface is therefore an attractive option.

Our approach is inspired by the equational approach to ellipsis resolution of Dalrymple et al. (1991) and the earlier GLUE approach of Crouch (1999), which used constraints to describe derivations.[3] We assume the equational approach (Dalrymple et al. 1991), reformulated for GLUE.[4] In the next section we will just focus on how parallelism preferences between ellipsis source and target can be formulated at the level of GLUE derivations.

5. Parallelism and Scope in Ellipsis Resolution

Ellipsis is interpreted by taking the source derivation and replacing source premises by corresponding target premises. This yields a target derivation that differs minimally from the source derivation only insofar as it is necessary to incorporate the new target premises.

In the case of overt quantified NPs in both source and target (e.g. Canadian Flag sentences, section 5.1), this predicts exact scope parallelism. But in cases where a non-quantificational NP in the source (target) is matched by a quantificational NP in the target (source) it turns out that this allows both

3. The constraint / description approach is also shared by the work on ellipsis in the Constraint Language for Lambda Structures (CLLS) framework (Egg et al. to appear, Egg and Erk 2002, Erk and Koller 2001).

4. Space prohibits details about this reformulation, which in any case are yet to be fully worked out. In outline, the original equational approach relied on using higher-order unification to solve equations relating meaning expressions. The reformulation uses higher-order unification to solve equations relating proof terms of GLUE derivations.

surface and inverse scope for the quantificational NP. However, to maximize the degree of parallelism between source and target derivations, there is a preference for the quantified NP to scope wide (section 5.2).

5.1. Quantifier-Quantifier Parallelism

Consider the Canadian Flag example (2), with source premises (22). From these premises we get derivations (23) and (24).

(2) A rhino sat in every corner. An elephant did, too.

(22) $\lambda P.a(rhino,\ P) : (g \multimap X) \multimap X$
$sit\text{-}in : h \multimap g \multimap f$
$\lambda Q.every(corner,\ Q) : (h \multimap Y) \multimap Y$

(23) **Surface Scope Source (∃∀)**

$$\dfrac{(g \multimap X) \multimap X \qquad \dfrac{\dfrac{(h \multimap Y) \multimap Y \qquad \dfrac{\dfrac{[g]^2 \qquad \dfrac{[h]^1 \qquad h \multimap g \multimap f}{g \multimap f}}{f}}{h \multimap f}\,{\multimap_{\mathcal{I},1}}}{f}}{g \multimap f}\,{\multimap_{\mathcal{I},2}}}{f}$$

(24) **Inverse Scope Source (∀∃)**

$$\dfrac{(h \multimap Y) \multimap Y \qquad \dfrac{\dfrac{(g \multimap X) \multimap X \qquad \dfrac{\dfrac{[g]^2 \qquad \dfrac{[h]^1 \qquad h \multimap g \multimap f}{g \multimap f}}{f}}{g \multimap f}\,{\multimap_{\mathcal{I},2}}}{f}}{h \multimap f}\,{\multimap_{\mathcal{I},1}}}{f}$$

The semantic derivation of the target sentence is obtained by replacing the source subject's premise with the target subject's premise:

(25) $\lambda P.a(elephant,\ P) : (g' \multimap X) \multimap X$

All instances of g in the source derivation are replaced by g' in the target derivation. With these minimal changes to the source derivation to obtain a target derivation, we obtain exact scope parallelism between source and target. Whichever of surface or inverse scope is chosen for the source, the same choice will be imposed on the target.

5.2. Quantifier-Name / Name-Quantifier Parallelism

Now consider sentences like (1), which prefer surface scope due to the subject of the ellipsis target being a proper name:

(1) A rhino sat in every corner. Babar did, too.

The surface and inverse scope derivations for the source sentence in (1) were just shown in (23) and (24). We replace the premise for *a rhino* in the source derivation with the premise for *Babar* (the parallel item in the target):

(26) $babar : g'$

Depending on which of the surface or inverse scopings ((23) or (24)) are chosen for the source, this replacement leads to two target derivations

(27) **Target from Surface Scope Source (∃∀)**

$$\dfrac{babar : g' \qquad \dfrac{\dfrac{(h \multimap Y) \multimap Y \qquad \dfrac{\dfrac{[g']^2 \qquad \dfrac{[h]^1 \qquad h \multimap g' \multimap f}{g' \multimap f}}{f}}{h \multimap f}{\scriptstyle \multimap_{\mathcal{I},1}}}{f}}{g' \multimap f}{\scriptstyle \multimap_{\mathcal{I},2}}}{f}$$

(28) **Target from Inverse Scope Source (∀∃)**

$$\dfrac{(h \multimap Y) \multimap Y \qquad \dfrac{\dfrac{babar : g' \qquad \dfrac{\dfrac{[g']^2 \qquad \dfrac{[h]^1 \qquad h \multimap g' \multimap f}{g' \multimap f}}{f}}{g' \multimap f}{\scriptstyle \multimap_{\mathcal{I},2}}}{f}}{h \multimap f}{\scriptstyle \multimap_{\mathcal{I},1}}}{f}$$

Note, however, that neither of these derivations is in normal form. Eliminating β-detours via the rule (20) to reduce the derivations to their canonical form shows that the two derivations correspond to the same underlying proof:

(29) **Normalized target (scopally unambiguous)**

$$\cfrac{(h \multimap Y) \multimap Y \qquad \cfrac{\cfrac{babar : g' \qquad \cfrac{[h]^1 \qquad h \multimap (g' \multimap f)}{g' \multimap f}}{f}}{h \multimap f}\multimap_{\mathcal{I},1}}{f}$$

Comparing the (β-normal) derivation (29) to the surface scope source (β-normal) derivation in (23), we can see that (29) is a subproof of (23). By contrast, when we compare (29) to the inverse scope source (β-normal) derivation in (24), we see that the proofs diverge after the third line. In other words (29) and (23) are parallel to a greater degree than (29) and (24) are, and the surface scope reading for ellipsis source in sentence (1) is the preferred reading for maximizing the degree of parallelism.

Thus, either surface or inverse source scoping is permissible on the GLUE approach, both leading to a scopally unambiguous target derivation. However, in terms of maximizing the degree of parallelism between source and target derivations, there is a preference for the source to have surface scope, where the quantified subject scopes wide.

It is important that the calculation of degree of parallelism compares normal form derivations. It is the underlying proofs that need to be compared, not the way in which they happen to be written down, with or without detours.

Let us now consider what happens when a non-quantified source subject is replaced by a quantified target subject, as in (30):

(30) Babar sat in every corner. A rhino did, too.

Here we predict that either scoping is available for the target, but with a preference for surface scope / wide scope subject.

The (normal form) source derivation is unambiguously:

(31) **Source derivation (scopally unambiguous)**

$$\cfrac{(h \multimap Y) \multimap Y \qquad \cfrac{\cfrac{babar : g \qquad \cfrac{[h]^1 \qquad h \multimap (g \multimap f)}{g \multimap f}}{f}}{h \multimap f}\multimap_{\mathcal{I},1}}{f}$$

At first sight it would appear that it is straightforward to replace the source premise $babar : g$ by $\lambda P.a(rhino,\ P) : (g' \multimap X) \multimap X$ to produce a target derivation. However, this replacement is misleadingly straightforward. It relies on the fact that we just happened to write the verb premise as $\lambda y \lambda x.see(x, y) : h \multimap g \multimap f$ We could just as easily have written an equivalent premise $\lambda x \lambda y.see(x, y) : g \multimap h \multimap f$ (by currying the function), in which case the source derivation would have been:

(32)
$$\dfrac{(h \multimap Y) \multimap Y \qquad \dfrac{\dfrac{[h]^1 \qquad \dfrac{babar : g \qquad g \multimap (h \multimap f)}{h \multimap f}}{f}}{h \multimap f}\multimap_{\mathcal{I},1}}{f}$$

In this essentially equivalent derivation, it is not obvious how to replace g by $(g \multimap X) \multimap X$ to produce a valid derivation. Yet clearly such a replacement should be possible. We therefore need to be clearer about what happens when a source name is replaced by a target quantifier.

There is a difference between (i) replacing a source quantifier by a target name and (ii) replacing a source name by a target quantifier. Case (i) works out easily because a name GLUE constructor such as g always entails the corresponding quantifier constructor $(g \multimap X) \multimap X$, as the following (type raising) derivation shows

(33)
$$\dfrac{\dfrac{[P : g \multimap x]^1 \qquad G : g}{P(G) : x}}{\lambda P.P(G) : (g \multimap x) \multimap x}\multimap_{\mathcal{I},1}$$

This means that whenever we want to replace a quantifier by a name, we always have the option of type-raising the name to a quantifier and replacing it directly. But for case (ii) the reverse entailment does not hold: a quantifier constructor cannot be type-lowered to entail a name constructor.

A general mechanism for replacing a name by a quantifier is instead to (a) replace the name premise by an assumption, and (b) at a suitable point in the derivation discharge this assumption and introduce the quantifier premise. This leads to the following two possible target derivations from (31):[5]

(34) **Wide Scope**

$$\dfrac{(g' \multimap X) \multimap X \qquad \dfrac{\dfrac{(h \multimap Y) \multimap Y \qquad \dfrac{\dfrac{[g']^2 \qquad \dfrac{[h]^1 \qquad h \multimap (g' \multimap f)}{g' \multimap f}}{f}}{h \multimap f}\multimap_{\mathcal{I},1}}{f}}{g' \multimap f}\multimap_{\mathcal{I},2}}{f}$$

(35) **Narrow Scope**

5. Source derivation (32) likewise gives rise to two possible target derivations.

$$\dfrac{(h \multimap Y) \multimap Y \qquad \dfrac{\dfrac{(g' \multimap X) \multimap X \qquad \dfrac{\dfrac{[g']^2 \qquad \dfrac{[h]^1 \qquad h \multimap (g' \multimap f)}{g' \multimap f}}{f}}{g' \multimap f}\multimap_{\mathcal{I},2}}{f}}{h \multimap f}\multimap_{\mathcal{I},1}}{f}$$

Comparing these normal form wide and narrow scope target derivations with the normalized source derivation reveals that the wider scope target displays a greater degree of parallelism to the source.

The preference for a quantifier replacing a name to scope wide is independent of the way in which we happen to write the verb premise down.

5.3. Measuring Parallelism

In the preceding examples we have measured derivational parallelism in terms of the degree of contiguous overlap between normal form source and target derivations. Target derivations that recapitulate larger contiguous chunks of source derivation are preferred. While intuitively natural, we do not at present have any solid formal justification for this measure of parallelism. Empirically, in terms of predicting scope preferences, this measure of parallelism seems justified.

5.4. Conclusion

According to this theory, the amount of overlap between ellipsis source and target derivations determines which scope readings are preferred. The preferred reading is the one that maintains maximum overlap. Thus, we are not ruling out inverse scope on the basis of the availability of surface scope. This contrasts sharply with Fox's economy approach. It also means that the inverse scope readings for sentences (1) and (30) are available, although we would still seek an explanation for why certain sentences resist inverse scope in the source more strongly than others (for one possible explanation, see Asher et al. 1997, 2001).

Asher et al. (1997, 2001) also consider ellipsis, scope, and parallelism and state the need for a preference ordering on scope readings. Although space restrictions prevent us from doing their account justice, it is useful to enumerate some of the similarities and differences between our approach and theirs. The two approaches are similar in their use of a logical form (GLUE derivations vs. Segmented Discourse Representation Structures), provision of a preference ordering, and a symmetric account of parallelism (source can influence target or vice versa). The differences, as we see them, are: (1) the Asher et al. approach provides no normal form for their logical forms, thus crucially relying on an intermediate representation without firm identity criteria; (2) they postulate a specific parallelism between names and wide scope existentials; (3) they require further modifications to their theory to handle general scope parallelism.

References

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

Asher, Nicholas, Daniel Hardt and Joan Busquets, 2001. Discoure parallelism, ellipsis, and ambiguity. *Journal of Semantics*, 18(1).

Asudeh, Ash and Richard Crouch, 2002. Glue semantics for HPSG. In van Eynde et al. (2002).

Bresnan, Joan, 2001. *Lexical-Functional Syntax*. Oxford: Blackwell.

Bunt, H and Reinhard Muskens, editors, 1999. *Computing Meaning*, volume 1. Dordrecht: Kluwer.

Crouch, Richard, 1999. Ellipsis and glue languages. In Shalom Lappin and Elabbas Benmamoun, editors, *Fragments: Studies in Ellipsis and Gapping*.

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, editor, 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, Stuart M. Shieber and Fernando C. N. Pereira, 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14(4):399–452.

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

Egg, Markus and Katrin Erk, 2002. A compositional account of VP ellipsis. In van Eynde et al. (2002).

Egg, Markus, Alexander Koller and Joachim Niehren, to appear. The Constraint Language for Lambda Structures. *Journal of Logic, Language, and Information.*

Erk, Katrin and Alexander Koller, 2001. VP ellipsis by tree surgery. In *Proceedings of the 13th Amsterdam Colloquium*. Amsterdam.

van Eynde, Frank, Lars Hellan and Dorothee Beermann, editors, 2002. *Proceedings of the 8th International HPSG Conference*. Stanford, CA: CSLI Publications.

Fiengo, Robert and Robert May, 1994. *Indices and Identity*. Cambridge, MA: MIT Press.

Fox, Danny, 1995. Economy and scope. *Natural Language Semantics*, 3(3):283–341.

Fox, Danny, 2000. *Economy and Semantic Interpretation*. Cambridge, MA: MIT Press.

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 LFG 2001 Conference*. Stanford, CA: CSLI Publications. URL `http://www-csli.stanford.edu/publications/`

van Genabith, Josef and Richard Crouch, 1999. How to glue a donkey to an f-structure. In Bunt and Muskens (1999).

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

Hardt, Daniel, 1993. *Verb Phrase Ellipsis: Form, Meaning, and Processing*. Ph.D. thesis, University of Pennsylvania.

Hardt, Daniel, 1999. Dynamic interpretation of verb phrase ellipsis. *Linguistics and Philosophy*, 22:185–219.

Hirschbühler, Paul, 1982. VP-deletion and across-the-board quantifier scope. In James Pustejovsky and Peter Sells, editors, *Proceedings of NELS 12*, pp. 132–139. University of Massachusetts, Amherst: GLSA.

Howard, William A., 1980. The formulae-as-types notion of construction. In Jonathan P. Seldin and J. Roger Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pp. 479–490. London: Academic press.

Jacobson, Pauline, 1998. Where (if anywhere) is transderivationality located? In Louise McNally and Peter W. Culicover, editors, *The Limits of Syntax*, pp. 303–366. San Diego, CA: Academic Press.

Jay, Barry and Neil Ghani, 1995. The virtues of eta-expansion. *Journal of Functional Programming*, 5(2):135–154.

Johnson, David and Shalom Lappin, 1997. A critique of the Minimalist Program. *Linguistics and Philosophy*, 20:273–333.

Johnson, David and Shalom Lappin, 1999. *Local Constraints Vs. Economy*. Stanford, CA: CSLI Publications.

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

Kratzer, Angelika, 1989. Stage-level and individual-level predicates. In Emmon Bach, Angelika Kratzer and Barbara Hall Partee, editors, *Papers in Quantification*. Amherst, MA: University of Massachusetts, Amherst.

Kratzer, Angelika, 1995. Stage-level and individual-level predicates. In Gregory N. Carlson and Francis Jeffry Pelletier, editors, *The Generic Book*. Chicago: University of Chicago Press.

Lappin, Shalom, 1993. Ellipsis resolution at S-structure. In A. J. Schafer, editor, *Proceedings of NELS 23*, pp. 255–269. University of Massachusetts, Amherst: GLSA.

Lappin, Shalom, editor, 1996. *The Handbook of Contemporary Semantic Theory*. Oxford: Blackwell.

Merchant, Jason, 2001. *The Syntax of Silence*. Oxford: Oxford University Press.

Montague, Richard, 1970. English as a formal language. In Bruno Visentini et al., editors, *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, editors, *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.

Prawitz, Dag, 1965. *Natural Deduction: A Proof-theoretical Study*. Stockholm: Almquist and Wiksell.

Sag, Ivan A., 1976. *Deletion and Logical Form*. Ph.D. thesis, MIT.

Shieber, Stuart M., Fernando C. N. Pereira and Mary Dalrymple, 1996. Interactions of scope and ellipsis. *Linguistics and Philosophy*, 19(5):527–552.

Williams, Edwin, 1977. Discourse and logical form. *Linguistic Inquiry*, 8(1):101–139.