Glue Semantics for HPSG*

Ash Asudeh[†‡] and Richard Crouch[‡]
Stanford University[†] and Xerox PARC[‡]

1 Introduction

The glue approach to semantic interpretation (Dalrymple, 1999) has been developed principally for Lexical Functional Grammar. Recent work has shown how glue can be used with a variety of syntactic theories (Asudeh and Crouch, 2001; Frank and van Genabith, 2001) and this paper outlines how it can be applied to HPSG. As well as providing an alternative form of semantics for HPSG, we believe that the benefits of HPSG glue include the following: (1) simplification of the Semantics Principle (Pollard and Sag, 1994); (2) a simple and elegant treatment of modifier scope, including empirical phenomena like quantifier scope ambiguity, the interaction of scope with raising, and recursive modification; (3) an analysis of control that handles agreement between controlled subjects and their co-arguments while allowing for a property denotation for the controlled clause (Chierchia, 1984a,b); (4) re-use of highly efficient techniques for semantic derivation already implemented for LFG, and which target problems of ambiguity management also addressed by Minimal Recursion Semantics (Copestake et al., 1995, 1999).

Glue semantics embodies a notion of ‘interpretation as deduction’ closely related to categorial grammar’s ‘parsing as deduction’. Syntactic analysis of a sentence yields a set of glue premises, which essentially state how bits of lexical meaning attach to words and phrases. Deduction in (linear) logic then combines the premises to derive a conclusion that attaches a meaning to the sentence as a whole. The innovation in this paper is to sketch how glue premises can be obtained from HPSG analyses; the subsequent stage of linear logic deduction is the same as when premises are obtained from LFG analyses.

The paper is organized as follows. Section 2 briefly reviews the way in which linear logic / glue deduction assembles sentence meanings given a set of lexical glue premises. Section 3 describes the adjustments to HPSG’s feature geometry necessary for it to construct sets of glue premises. Section 4 show how these adjustments give rise to semantic analyses of a selected number of illustrative phenomena (quantifier scope ambiguity, raising and *de dicto* scope, recursive modification, and control). Section 5 makes some initial comparisons between glue and Minimal Recursion Semantics for HPSG, and concludes.

2 Glue Deductions

This section reviews how linear logic deduction is used to assemble meanings from sets of lexical premises — see (Dalrymple, 1999, 2001) for more details. We postpone for the moment the question of how these premises are obtained, which is the topic of the next section. To start with, consider the sentence

(1) Kim walked.

and the relevant parts of its analysis as summarized in

*For comments and criticisms, we would like to thank Mary Dalrymple, Carl Pollard, Ivan Sag, two anonymous HPSG 2001 reviewers, and the audience at HPSG 2001. Asudeh is supported in part by SSHRC Doctoral Fellowship 752-98-0424.

(2)

$$\begin{bmatrix} \text{CONT} & \begin{bmatrix} \text{HEAD-RESOURCE} & \boxed{1} \end{bmatrix} \\ \text{GLUE} & \left\langle \textit{kim}{:}\boxed{2},\ \textit{walk}{:}\boxed{2} \multimap \boxed{1} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \text{CONT} & \begin{bmatrix} \text{HEAD-RESOURCE} & \boxed{2} \end{bmatrix} \\ \text{GLUE} & \left\langle \textit{kim}{:}\boxed{2} \right\rangle \end{bmatrix} \qquad \begin{bmatrix} \text{CONT} & \begin{bmatrix} \text{HEAD-RESOURCE} & \boxed{1} \end{bmatrix} \\ \text{GLUE} & \left\langle \textit{walk}{:}\boxed{2} \multimap \boxed{1} \right\rangle \end{bmatrix}$$

The GLUE feature of the mother lists two premises. The first, $kim{:}\boxed{2}$, associates the constant kim with the semantic resource $\boxed{2}$, which is the HEAD-RESOURCE of the subject noun phrase. The premise $walk{:}\boxed{2} \multimap \boxed{1}$, associates the one-place predicate $walk$ with the linear implication $\boxed{2} \multimap \boxed{1}$. This implication says that we can consume the semantic resource $\boxed{2}$ to produce the resource $\boxed{1}$, which is the HEAD-RESOURCE of both the verb and the sentence of which the verb is the head.

The principal feature of linear logic, for our purposes, is its resource sensitivity. Premises in linear logic are resources that get used up in inference to produce conclusions. This is unlike traditional logic where premises are not consumed, and may be re-used or even not used at all. These differences between traditional and linear logic can be illustrated by the following inference patterns

(3) Premise re-use

Traditional Logic	Linear Logic
$A,\ A \to B \vdash B$	$A,\ A \multimap B \vdash B$
$A,\ A \to B \vdash B \wedge A$	$A,\ A \multimap B \not\vdash B \otimes A$
Premise A re-used, conjoined with conclusion B	Premise A is consumed to produce conclusion B, no longer available for conjunction with B

(4) Premise non-use

Traditional Logic	Linear Logic
$A, B \vdash A$	$A, B \not\vdash A$
Can ignore premise B	Cannot ignore premise B

Within both categorial grammar and glue semantics, the resource sensitivity of linear logic has been invoked to account for the resource sensitivity of natural language. Each word contributes a premise, and each word/premise must make exactly one contribution to the analysis of the sentence.

Our glue premises pair meaning terms (on the left of colons) with linear logic formulas (on the right). This is standard given the Curry-Howard Isomorphism (Curry and Feys, 1961). Rules of inference determine how the meaning terms of premises are combined to give the meaning term of the conclusion. For example *modus ponens* (implication elimination) leads to the functional application of meaning terms, whereas hypothetical reasoning (implication introduction) leads to λ-abstraction, as shown:

(5)

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

The symbol "$\multimap$" is linear implication. The notation "$\multimap_{\mathcal{E}}$" indicates that the rule of implication elimination has been applied, and "$\multimap_{\mathcal{I},i}$" indicates that the rule of implication introduction has been applied, discharging assumption i.

Let us now return to sentence (1) and the premises in the GLUE of (2). Ignoring for the moment the meaning terms decorating the two glue premises, a one step derivation consumes both premises to produce a single semantic resource for the sentence:

(6)

$$\dfrac{\boxed{2} \multimap \boxed{1} \qquad \boxed{2}}{\boxed{1}} \multimap_{\mathcal{E}}$$

Including the meaning terms, this derivation dictates that there should be a functional application of $walk$ to kim to construct the meaning of $\boxed{1}$, as follows:

(7)

$$\dfrac{walk : \boxed{2} \multimap \boxed{1} \qquad kim : \boxed{2}}{walk(kim) : \boxed{1}} \multimap_{\mathcal{E}}$$

Note that whatever the internal nature of the meaning terms, the linear logic derivation determines that they are combined by functional application. More generally, the form of the semantic composition is completely driven by the structure of the linear logic derivation, and is independent of the details of the meaning language used to provide the meaning terms. This independence from the meaning language allows ready application of the glue approach to a variety of semantic representations, such as Montague's IL, DRT and UDRT (Dalrymple, 1999; van Genabith and Crouch, 1999; Crouch et al., 1999). Moreover, the vast majority of phenomena can be handled using an extremely restricted fragment of essentially propositional, implication-only linear logic. Efficient proof techniques exist for this fragment, especially given the prevalence of modifiers (treated as $\phi \multimap \phi$ glue identities) in linguistically typical input (Gupta and Lamping, 1998).

An important feature of glue semantics is its account of semantic ambiguity: given a single set of premises from a syntactically unambiguous or disambiguated sentence, multiple derivations constructing alternate meanings may be possible. Thus, semantic ambiguity does not need to be pushed down into the syntax. Quantifier scope ambiguity is the best known, but by no means the only, example of such semantic ambiguity. For "everyone saw something", suppose we have the following premises:

(8)

$$\begin{array}{rcl} \lambda y.\ \lambda x.\ see(x,y) & : & \boxed{3} \multimap (\boxed{2} \multimap \boxed{1}) \\ \lambda P.\ \forall x.[person(x) \rightarrow P(x)] & : & (\boxed{2} \multimap S_1) \multimap S_1 \\ \lambda Q.\ \exists y.[thing(y) \wedge Q(y)] & : & (\boxed{3} \multimap S_2) \multimap S_2 \end{array}$$

The variables S_1 and S_2 range over atomic resources, corresponding to the phrase that the quantified NPs take as their scope. Given the resource-sensitive nature of linear logic, the resources to which S_1 and S_2 are instantiated must be ones (i) provided by the lexical premises, and (ii) not otherwise consumed in the derivation. Moreover the semantic resources are sorted (see section 3.2 below), such that $\boxed{1}$ is a *t-resource* and bears a truth-denoting or propositional meaning, while $\boxed{2}$ and $\boxed{3}$ are *e-resources*, and bear entity-denoting meanings. S_1 and S_2 are variables over *t-resources* only. There are two distinct glue derivations consuming the above premises, giving rise to two scopings, though in both cases $S_1 = S_2 = \boxed{1}$. The two derivations, without meaning terms are:

(9) a. $\exists y.[thing(y) \wedge \forall x.[person(x) \rightarrow see(x,y)]]$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\boxed{3} \multimap (\boxed{2} \multimap \boxed{1}) \qquad [\boxed{3}]^1}{\boxed{2} \multimap \boxed{1}} \multimap_{\mathcal{E}} \qquad [\boxed{2}]^2}{\boxed{1}} \multimap_{\mathcal{E}}}{\boxed{2} \multimap \boxed{1}} \multimap_{\mathcal{I},2} \qquad (\boxed{2} \multimap S_1) \multimap S_1}{\boxed{1}} \multimap_{\mathcal{E}}}{\boxed{3} \multimap \boxed{1}} \multimap_{\mathcal{I},1} \qquad (\boxed{3} \multimap S_2) \multimap S_2}{\boxed{1}} \multimap_{\mathcal{E}}$$

b. $\forall x.[person(x) \to \exists y.[thing(y) \wedge see(x, y)]]$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\boxed{3} \multimap (\boxed{2} \multimap \boxed{1}) \qquad [\boxed{3}]^1}{\boxed{2} \multimap \boxed{1}} \multimap_{\mathcal{E}} \qquad [\boxed{2}]^2}{\boxed{1}} \multimap_{\mathcal{E}}}{\boxed{3} \multimap \boxed{1}} \multimap_{\mathcal{I},1} \qquad (\boxed{3} \multimap S_2) \multimap S_2}{\boxed{1}} \multimap_{\mathcal{E}}}{\boxed{2} \multimap \boxed{1}} \multimap_{\mathcal{I},2} \qquad (\boxed{2} \multimap S_1) \multimap S_1}{\boxed{1}} \multimap_{\mathcal{E}}$$

Where $S_1 = S_2 = \boxed{1}$ in both proofs.

Both derivations start in the same manner: applying the verb's premise $\boxed{3} \multimap (\boxed{2} \multimap \boxed{1})$ to assumptions of $\boxed{3}$ and $\boxed{2}$ to obtain an intermediate conclusion $\boxed{1}$. The derivations then diverge in the order in which they discharge these assumptions to derive either $\boxed{2} \multimap \boxed{1}$ or $\boxed{3} \multimap \boxed{1}$, and combine these with the quantified NP premises.

Showing the meaning terms explicitly for the first of these derivations, we have

(10)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lambda y.\lambda x.see(x, y) : \boxed{3} \multimap (\boxed{2} \multimap \boxed{1}) \qquad [y' : \boxed{3}]^1}{\lambda x.see(x, y') : \boxed{2} \multimap \boxed{1}} \multimap_{\mathcal{E}} \qquad [x' : \boxed{2}]^2}{see(x', y') : \boxed{1}} \multimap_{\mathcal{E}}}{\lambda x'.see(x', y') : \boxed{2} \multimap \boxed{1}} \multimap_{\mathcal{I},2} \qquad \lambda P.\forall x.[person(x) \to P(x)] : (\boxed{2} \multimap S_1) \multimap S_1}{\forall x.[person(x) \to see(x, y')] : \boxed{1}} \multimap_{\mathcal{E}}}{\lambda y'.\forall x.[person(x) \to see(x, y')] : \boxed{3} \multimap \boxed{1}} \multimap_{\mathcal{I},1} \qquad \lambda Q.\exists y.[thing(y) \wedge Q(y)] : (\boxed{3} \multimap S_2) \multimap S_2}{\exists y.[thing(y) \wedge \forall x.[person(x) \to see(x, y)]] : \boxed{1}} \multimap_{\mathcal{E}}$$

The combination of meaning terms for the second derivation is analagous.

An important feature of this analysis of scope ambiguity is that it arises solely from interactions of the standard rules of inference for implication. No additional machinery, like quantifier storage, is required. Indeed, one could view this deductive account of scope ambiguity as providing a logical reconstruction of Cooper storage, where quantifier storage is modeled by hypothesis introduction, and quantifier retrieval by hypothesis discharge (Pereira, 1990). As such, this reconstruction simply falls out from mechanisms already required elsewhere.

3 Glue Premises for HPSG

The glue approach necessitates certain adjustments to HPSG theory, particulary to its feature geometry. It also allows certain simplifications. We present these changes in this section.

3.1 Simplifying the Semantics Principle

The Semantics Principle from Pollard and Sag (1994:323, (16)) is considerably simplified to:

(11) **SEMANTICS PRINCIPLE (glue version)**
1. The GLUE of a *phrase* is the multiset union of the GLUE values of its daughters.
2. The CONTENT of a *phrase* is the CONTENT of its head daughter.

We can state this as a constraint on *phrases*:[1]

[1] We represent multiset union as list append, but order is not relevant.

(12) **SEMANTICS PRINCIPLE (glue/constraint version)**

$$\textit{phrase} \mapsto \begin{bmatrix} \text{GLUE} & \boxed{A} \oplus \ldots \oplus \boxed{N} \\ \text{CONTENT} & \boxed{1} \\ \text{HEAD-DTR} & \boxed{0}\begin{bmatrix}\text{CONTENT} & \boxed{1}\end{bmatrix} \\ \text{DAUGHTERS} & \left\langle \boxed{0}\begin{bmatrix}\text{GLUE} & \boxed{A}\end{bmatrix}, \ldots, \begin{bmatrix}\text{GLUE} & \boxed{N}\end{bmatrix}\right\rangle \end{bmatrix}$$

Unlike the Semantics Principle of Pollard and Sag (1994), the glue Semantics Principle does not need to: 1) distinguish between the semantic contributions of quantifiers and other signs; 2) distinguish between adjunct and non-adjunct daughters; 3) handle quantifier scoping.

3.2 The Internal Structure of CONTENT

The price for simplifying the Semantics Principle is the introduction of resource features in CONTENT. These correspond to the atomic propositions in the linear logic proofs. Readers familiar with Minimal Recursion Semantics (Copestake et al., 1995, 1999) may notice certain similarities between that approach and this one. There are several important points of convergence, but also some interesting differences. The two approaches will be discussed further in section 5.1.

Let us introduce some lexical entries, to make the following discussion a little more concrete. The lexical entries for *every*, *student*, and *solves* are:

(13) a.

$$\begin{bmatrix} \textbf{“EVERY”} & \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & {}_{det}\begin{bmatrix} \text{SPEC} & \text{N}': \begin{bmatrix} \text{HEAD-RES} & \boxed{3}\,{}_{ref}\begin{bmatrix} \text{PERSON} & \textit{3rd} \\ \text{NUMBER} & \textit{sing} \end{bmatrix} \\ \text{VAR-RES} & \boxed{1} \\ \text{RESTR-RES} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{GLUE} & \langle \lambda P.\lambda Q.\forall\, x.[P(x) \rightarrow Q(x)]:(\boxed{1} \multimap \boxed{2}) \multimap ((\boxed{3} \multimap G) \multimap G) \rangle \end{bmatrix}$$

b.

$$\begin{bmatrix} \textbf{“STUDENT”} & \\ \text{CAT} \mid \text{HEAD} & \textit{noun} \\ \text{CONT} & {}_{common}\begin{bmatrix} \text{HEAD-RES} & {}_{ref}\begin{bmatrix} \text{PERSON} & \textit{3rd} \\ \text{NUMBER} & \textit{sing} \end{bmatrix} \\ \text{VAR-RES} & \boxed{1} \\ \text{RESTR-RES} & \boxed{2} \end{bmatrix} \\ \text{GLUE} & \langle \lambda x.\text{student}(x):\boxed{1} \multimap \boxed{2} \rangle \end{bmatrix}$$

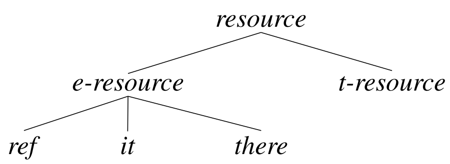
c. $$\left[\begin{array}{ll}
\textbf{``SOLVES''} & \\
\text{CAT} & \left[\begin{array}{ll}
\text{HEAD} & \left[\text{VFORM} \quad \textit{fin}\right] \\
\text{VALENCE} & \left[\begin{array}{ll}
\text{SUBJ} & \langle \text{NP:}\left[\text{HEAD-RES} \quad \boxed{2}\,{}_{\textit{ref}}\!\left[\begin{array}{ll}\text{PER} & \textit{3rd} \\ \text{NUM} & \textit{sing}\end{array}\right]\right]\rangle \\
\text{COMPS} & \langle \text{NP:}[\text{HEAD-RES}\ \boxed{3}]\rangle
\end{array}\right]
\end{array}\right] \\
\text{CONT} & {}_{\textit{solve}}\!\left[\begin{array}{ll}
\text{HEAD-RES} & \boxed{1} \\
\text{SOLVER} & \boxed{2} \\
\text{SOLVED} & \boxed{3}
\end{array}\right] \\
\text{GLUE} & \langle \lambda y.\lambda x.\text{solve}(x,y){:}\boxed{3} \multimap (\boxed{2} \multimap \boxed{1})\rangle
\end{array}\right]$$

The type *content* introduces the feature HEAD-RESOURCE (HEAD-RES), with value *resource*. HEAD-RES provides a unique identifier for the maximal projection of each head.

(14) *content:* $\left[\text{HEAD-RESOURCE} \quad \textit{resource}\right]$

The type hierarchy for *resource* is:

(15)



The type *e-resource* does double duty. Not only is it used in resource sorting for the linear logic (see the discussion preceding (9) above), it also takes over the function of *index*, which we do not have in our system. Thus, it introduces the agreement features:

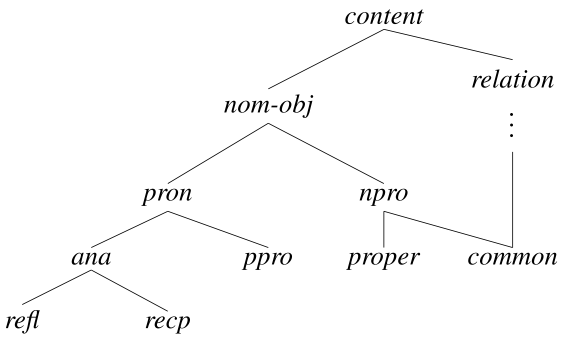
(16) $$\textit{e-resource:}\left[\begin{array}{ll}
\text{PERSON} & \textit{person} \\
\text{NUMBER} & \textit{number} \\
\text{GENDER} & \textit{gender}
\end{array}\right]$$

For example, notice that the HEAD-RESOURCE value for "student" in (13b) is of type *ref*, a subtype of *e-resource*, with [*3rd*, *sing*] agreement features. Notice also that the lexical entry for the verb "solves" requires that its subject's HEAD-RES is *ref* with [*3rd*, *sing*] agreement features. This enforces the agreement between the subject and the verb through unification (Pollard and Sag, 1994), such that "student" could be the subject of "solves", but not of "solve", which would have different agreement features.

The type *content* has the usual subtypes (Pollard and Sag, 1994), with three exceptions. First, there is no longer any need for the type *quantifier*. This type introduced the features SEMDET, which registered the quantificational determiner (e.g., *forall*, *exists*, etc.), and RESTIND, which took the index of the quantifier's restriction as its value. The quantificational determiner is now registered on the meaning language side of the glue constructor in the lexical entry for the quantifier (see (13a) above). The function of RESTIND is fulfilled on the linear logic side of the quantifier's glue: the quantifier consumes the implicational resource introduced by its restriction. Second, we have removed the type *psoa* and its subtypes (*qfpsoa*, *control-qfpsoa*, etc.), which are relevant if situation semantics (Barwise and Perry, 1983) is assumed for interpretation. Glue is in principle compatible with situation semantics, but in this system that would mean having situation semantics as the meaning language on the left hand side of glue meaning constructors.

We replace *psoa* with *relation*, which is effectively just the type *qfpsoa* from Pollard and Sag (1994). Third, the type *npro* has been split into two types: *proper*, for e-type nominals like proper names, and *common* for ⟨e, t⟩-type nominals like common nouns. The *content* type hierarchy is:

(17)



We assume, following Pollard and Sag (1994:338), appropriate subtypes for *relation* which state their own feature declarations for their semantic roles.

The type *nom-obj* and its subtypes no longer introduce a feature RESTRICTION, as this information would be redundant with the meaning language in the glue meaning constructor (see the lexical entry for "student" above). However, to account for nominals with semantic roles to assign (e.g., nominalizations, nouns with complements, etc.), we allow the type *common* to inherit from *relation*. In addition, lexical items with *nom-obj* type CONTENTS restrict the typing of the value of HEAD-RESOURCE, which is of type *e-resource*, to encode appropriate agreement information.

A further feature declaration is stated on the type *common*, as shown in (18). The features VARIABLE-RESOURCE and RESTRICTION-RESOURCE introduce semantic resources corresponding to an N′. A quantificational determiner consumes these to make a quantified NP.

(18)
$$\textit{common:}\begin{bmatrix}\text{VARIABLE-RESOURCE} & \textit{e-resource}\\ \text{RESTRICTION-RESOURCE} & \textit{t-resource}\end{bmatrix}$$

We will henceforth abbreviate these features as VAR-RES and RESTR-RES. There is a distinction between HEAD-RES, VAR-RES and RESTR-RES. The values of the latter two features place sortal restrictions on common nouns in combination with determiners. In other words, a generalized quantifier (type ⟨⟨e,t⟩, ⟨⟨e,t⟩, t⟩⟩), will consume the resources contributed by its noun's VAR-RES and RESTR-RES, but a verb will consume the resource corresponding to the maximal NP projection, which will be the HEAD-RES value.

The value of HEAD-RES is not only the identifier for the maximal projection, it also serves the same function as INDEX in our system. As already discussed, nominals have a HEAD-RES value of type *e-resource*, with agreement features and a verb whose meaning consumes the nominal's HEAD-RES will also be lexically required to agree with it, by placing restrictions on the NP's agreement features. Binding theory will also make reference to the HEAD-RES value to guarantee agreement between a pronominal and its antecedent. Lastly, the HEAD-RES value will be structure-shared with the values of semantic roles in *relations*.

4 Analyses of Selected Phonemena

4.1 Quantifier Scoping

The fundamentals of the glue account of scope ambiguity were reviewed in section 2. This section illustrates how the mechanism applies to noun phrases more complex than the quantified pronouns "everyone" and "something".

4.1.1 Simple Scope

Consider the familiar scope ambiguity for the following sentence:

(19) Every student solves a problem.

The lexical entries for *every*, *student*, and *solves* were given in (13) above. The features VAR-RES and RESTR-RES are introduced by the *nominal-object* subtype of *content*. They introduce semantic resources corresponding to N′, which determiners consume to make quantified NPs. The lexical entries for *a* and *problem* would be similar to that of *every* and *student* respectively, but with appropriate changes to the meaning terms. In combination, these lexical entries result in the following partial specification for *Every student solved a problem*:

(20)
$$\begin{bmatrix} \text{CONT} \mid \text{HEAD-RES } \boxed{1} & \\ & \\ \text{GLUE} & \left\langle \begin{array}{l} \lambda\text{P}.\lambda\text{Q}.\forall\text{x}.[\text{P(x)} \rightarrow \text{Q(x)}]\text{:}(\boxed{4} \multimap \boxed{5}) \multimap ((\boxed{2} \multimap \text{G}) \multimap \text{G}), \\ \lambda\text{y.student(y)}\text{:}\boxed{4} \multimap \boxed{5}, \\ \lambda\text{y}.\lambda\text{x.solve(x,y)}\text{:}\boxed{3} \multimap (\boxed{2} \multimap \boxed{1}), \\ \lambda\text{P}.\lambda\text{Q}.\exists\text{ x}.[\text{P(x)} \wedge \text{Q(x)}]\text{:}(\boxed{6} \multimap \boxed{7}) \multimap ((\boxed{3} \multimap \text{H}) \multimap \text{H}), \\ \lambda\text{z.problem(z)}\text{:}\boxed{6} \multimap \boxed{7} \end{array} \right\rangle \end{bmatrix}$$

Let us focus on the construction of the meaning of the quantified NP *every student*. This proceeds according to the subproof in (21).

(21)
$$\frac{\lambda y.student(y) : \boxed{4} \multimap \boxed{5} \qquad \lambda P.\lambda Q.\forall x.[P(x) \rightarrow Q(x)] : (\boxed{4} \multimap \boxed{5}) \multimap ((\boxed{2} \multimap G) \multimap G)}{\lambda Q.\forall x.[student(x) \rightarrow Q(x)] : (\boxed{2} \multimap G) \multimap G} \multimap_{\mathcal{E}}$$

The common noun *student* has a one place predicate as its meaning term. This is reflected by the linear logic glue formlua $\boxed{4} \multimap \boxed{5}$. This is an implication from an *e-resource* ($\boxed{4}$) corresponding to *student's* VAR-RES to a *t-resource* ($\boxed{5}$) corresponding to *student's* RESTR-RES. Taking the linear logic formula to determine the type declaration for the meaning term, this entails that $\lambda y.student(y)$ is an ⟨e, t⟩ expression.

The generalized determiner *every* has a glue formula $(\boxed{4} \multimap \boxed{5}) \multimap ((\boxed{2} \multimap G) \multimap G)$, where G is a variable over atomic *t-resources*. The formula corresponds to the standard Montagovian type for generalized determiners, ⟨⟨e, t⟩, ⟨⟨e, t⟩, t⟩⟩, and indicates that the determiner consumes the N′ semantics to produce a generalized quantifier semantics, as shown in (21).

The result of subproof (21) is an expression parallel to the *everyone*/*something* quantified pronoun premises in (9). Given a similar subproof for *a problem*, we can derive two scopings for *every student solves a problem* in exactly the same ways as (9).

(22)
$$\dfrac{\lambda Q.\exists x.[problem(x) \wedge Q(x)] : (\boxed{3} \multimap H) \multimap H \qquad \dfrac{\dfrac{\lambda Q.\forall y.[student(y) \rightarrow Q(y)] : (\boxed{2} \multimap G) \multimap G \qquad \dfrac{[z : \boxed{3}]^1 \quad \lambda y.\lambda x.solve(x, y) : \boxed{3} \multimap (\boxed{2} \multimap \boxed{1})}{\lambda x.solve(x, z) : \boxed{2} \multimap \boxed{1}} \multimap_{\mathcal{E}}}{\forall y.[student(y) \rightarrow solve(y, z)] : \boxed{1}} \multimap_{\mathcal{E}}, G = \boxed{1}}{\lambda z.\forall y.[student(y) \rightarrow solve(y, z)] : \boxed{3} \multimap \boxed{1}} \multimap_{\mathcal{I},1}}{\exists x.[problem(x) \wedge \forall y.[student(y) \rightarrow solve(y, x)]] : \boxed{1}} \multimap_{\mathcal{E}}, H = \boxed{1}$$

(23)
$$\dfrac{\lambda Q.\forall y.[student(y) \rightarrow Q(y)] : (\boxed{2} \multimap G) \multimap G \qquad \dfrac{\dfrac{\lambda Q.\exists x.[problem(x) \wedge Q(x)] : (\boxed{3} \multimap H) \multimap H \qquad \dfrac{[z : \boxed{2}]^1 \quad \lambda x.\lambda y.solve(x, y) : \boxed{2} \multimap (\boxed{3} \multimap \boxed{1})}{\lambda y.solve(z, y) : \boxed{3} \multimap \boxed{1}} \multimap_{\mathcal{E}}}{\exists x.[problem(x) \wedge solve(z, x)] : \boxed{1}} \multimap_{\mathcal{E}}, H = \boxed{1}}{\lambda z.\exists x.[problem(x) \wedge solve(z, x)] : \boxed{2} \multimap \boxed{1}} \multimap_{\mathcal{I},1}}{\forall y.[student(y) \rightarrow \exists x.[problem(x) \wedge solve(y, x)]] : \boxed{1}} \multimap_{\mathcal{E}}, G = \boxed{1}$$

4.1.2 Modified NPs

Despite having three quantifiers, it is well known that

(24) Every representative of a company saw a sample.

permits only five, and not 3!=6 scopings. Some permutations of quantifiers lead to unbound variables, e.g. as shown underlined in the semantic representation:

$$\forall x.(\text{rep}(x) \wedge \text{of}(x, \underline{y})) \rightarrow \exists y.\text{company}(y) \wedge \exists z.\text{sample}(z) \wedge \text{see}(x, z)$$

The deductive glue account of scope predicts this, in that the problematic permutation does not correspond to a logically valid glue derivation.

Readers who are not interested in exactly why the glue analysis leads to five and not six scopings may wish to skip the remainder of this subsection. For readers who are interested, assume the following lexical entry for the preposition *of*.

(25)

$$\begin{bmatrix} \textbf{“OF”} & \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & {}_{\textit{prep}}\begin{bmatrix} \text{MOD} & \text{N}'\!:\begin{bmatrix} \text{HEAD-RES} & \boxed{1} \\ \text{VAR-RES} & \boxed{2} \\ \text{RESTR-RES} & \boxed{3} \end{bmatrix} \\ \text{VALENCE} & \begin{bmatrix} \text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \langle \text{NP:}[\text{HEAD-RES}\ \boxed{4}] \rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{HEAD-RES} & \boxed{1} \end{bmatrix} \\ \text{GLUE} & \left\langle \begin{array}{l} \text{of:}\boxed{1}, \\ \lambda\text{P}.\lambda\text{x}.\lambda\text{Q}.\lambda\text{y}.[\text{Q(y)} \wedge \text{P(x,y)}]\text{: } \boxed{1} \multimap \boxed{4} \multimap (\boxed{2} \multimap \boxed{3}) \multimap (\boxed{2} \multimap \boxed{3}) \end{array} \right\rangle \end{bmatrix} \end{bmatrix}$$

The first glue premise provides a two place predicate of as the meaning of $\boxed{1}$. The second premise consumes this predicate, the meaning of the complement NP, and the meaning of the N′ to which the PP attaches, to produce a modifier meaning of the N′. Analysis of the sentence yields the glue premises:

(26)

$$\begin{array}{rl} \lambda z.\lambda x.\ \text{see}(x, z): & \boxed{4} \multimap \boxed{2} \multimap \boxed{1} \\ \lambda P.\lambda Q.\ \exists z.[P(z) \wedge Q(z)]: & (\boxed{v4} \multimap \boxed{r4}) \multimap ((\boxed{4} \multimap G) \multimap G) \\ \lambda z.\ \text{sample}(z): & \boxed{v4} \multimap \boxed{r4} \\ \lambda P.\lambda Q.\ \exists y.[P(y) \wedge Q(y)]: & (\boxed{v3} \multimap \boxed{r3}) \multimap ((\boxed{3} \multimap H) \multimap H) \\ \lambda y.\ \text{company}(y): & \boxed{v3} \multimap \boxed{r3} \\ \lambda P.\lambda Q.\ \forall x.[P(x) \rightarrow Q(x)]: & (\boxed{v2} \multimap \boxed{r2}) \multimap ((\boxed{2} \multimap J) \multimap J) \\ \lambda x.\ \text{rep}(x): & \boxed{v2} \multimap \boxed{r2} \\ \text{of}: & \boxed{2} \\ \lambda P.\lambda y.\lambda Q.\lambda x.[Q(x) \wedge P(x, y)]: & \boxed{2} \multimap \boxed{3} \multimap ((\boxed{v2} \multimap \boxed{r2}) \multimap (\boxed{v2} \multimap \boxed{r2})) \end{array}$$

Subproofs combine some of these premises to yield intermediate conclusions:

(27)

$$\begin{array}{rl} \lambda z.\lambda x.\ \text{see}(x, z): & \boxed{4} \multimap \boxed{2} \multimap \boxed{1} \\ \lambda Q.\ \exists z.[\text{sample}(z) \wedge Q(z)]: & (\boxed{4} \multimap G) \multimap G \\ \lambda Q.\ \exists y.[\text{company}(y) \wedge Q(y)]: & (\boxed{3} \multimap H) \multimap H \\ \lambda P.\lambda Q.\ \forall x.[P(x) \rightarrow Q(x)]: & (\boxed{v2} \multimap \boxed{r2}) \multimap ((\boxed{2} \multimap J) \multimap J) \\ \lambda y.\lambda x.[\text{rep}(x) \wedge \text{of}(x, y)]: & \boxed{3} \multimap (\boxed{v2} \multimap \boxed{r2}) \end{array}$$

Given that $\boxed{r2}$ is a *t-resource*, one way of scoping *a company* is to give it scope over the restriction of the NP it modifies, as follows

(28)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lambda z.\lambda x.[\mathrm{rep}(x) \wedge \mathrm{of}(x,z)] : \boxed{3} \multimap (\boxed{v2} \multimap \boxed{r2}) \quad [y : \boxed{3}]^1}{\lambda x.[\mathrm{rep}(x) \wedge \mathrm{of}(x,y)] : \boxed{v2} \multimap \boxed{r2}} \quad [x : \boxed{v2}]^2}{[\mathrm{rep}(x) \wedge \mathrm{of}(x,y)] : \boxed{r2}}}{\lambda y.[\mathrm{rep}(x) \wedge \mathrm{of}(x,y)] : \boxed{3} \multimap \boxed{r2}}{\scriptstyle \multimap_{\mathcal{I},1}} \quad \lambda Q.\exists y.[\mathrm{company}(y) \wedge Q(y)] : (\boxed{3} \multimap H) \multimap H}{\exists y.[\mathrm{company}(y) \wedge [\mathrm{rep}(x) \wedge \mathrm{of}(x,y)]] : \boxed{r2}}}{\lambda x.\exists y.[\mathrm{company}(y) \wedge [\mathrm{rep}(x) \wedge \mathrm{of}(x,y)]] : \boxed{v2} \multimap \boxed{r2}}{\scriptstyle \multimap_{\mathcal{I},2}} \quad \lambda P\lambda Q.\forall x.[P(x) \to Q(x)] : (\boxed{v2} \multimap \boxed{r2}) \multimap ((\boxed{2} \multimap J) \multimap J)}{\lambda Q.\forall x.[\exists y.[\mathrm{company}(y) \wedge [\mathrm{rep}(x) \wedge \mathrm{of}(x,y)]] \to Q(x)] : (\boxed{2} \multimap J) \multimap J}$$

This quantifier, corresponding to *every representative of a company*, can then take scope over the sentence, $\boxed{1}$, permuting in either order with the quantifier *a sample*.

Further scopings arise when *a company* takes scope over the entire sentence, and not just the subject NP restriction. However, in all these cases, the *a company* quantifier is forced to take scope over the *every representative of* quantifier. This can be seen from the following partial derivation (shown without meaning terms):

(29)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\boxed{3} \multimap (\boxed{v2} \multimap \boxed{r2}) \quad [\boxed{3}]}{\boxed{v2} \multimap \boxed{r2}} \quad (\boxed{v2} \multimap \boxed{r2}) \multimap ((\boxed{2} \multimap J) \multimap J)}{(\boxed{2} \multimap J) \multimap J} \quad \dfrac{\dfrac{\dfrac{\boxed{4} \multimap \boxed{2} \multimap \boxed{1} \quad [\boxed{4}]}{\boxed{2} \multimap \boxed{1}} \quad [\boxed{2}]}{\boxed{1} *}}{\boxed{2} \multimap \boxed{1}}}{\boxed{1} *}}{\boxed{3} \multimap \boxed{1}} \quad (\boxed{3} \multimap H) \multimap H}{\boxed{1} *}$$

The *a sample* quantifier, $(\boxed{4} \multimap G) \multimap G$, can be scoped at any of the points marked with an asterisk by discharging the $\boxed{4}$ assumption. Note, however, that the *a company* quantifier, $(\boxed{3} \multimap H) \multimap H$, has to take wide scope over the *every representative* quantifier $(\boxed{2} \multimap J) \multimap J$. This is because the restriction of *every representative* introduces the dependence on the *a company* head resource $\boxed{3}$. The *every representative* quantifier therefore has to be scoped in before one can form a $\boxed{3} \multimap \boxed{1}$ implication, and scope the *a company* quantifier.

4.2 Recursive Modification

Kasper (1997) has observed that sentences like the following are problematic for the semantics of modifiers resulting from the Semantics Principle in Pollard and Sag (1994):

(30) An apparently smart student failed.

The analysis in Pollard and Sag (1994) results in a CONTENT for *apparently* that includes both the RESTRICTION of *smart* and that of *student*. In other words, Pollard and Sag's (1994) analysis incorrectly predicts that the student is not only smart, but also just apparent.

In addition to the problem of incorrect modification, Kasper notes that the Pollard and Sag (1994) Semantics Principle does not give a uniform analysis to attributive and predicative adjectives. Although this lack of uniformity may be syntactically and semantically motivated for the adjectives themselves, the more serious problem is that the lexical ambiguity is also multiplied out for the modifiers of adjectives (e.g., adverbs), and this is clearly unmotivated.

Kasper (1997) provides a solution to these problems which involves three new attributes for the feature MOD: ARG (the old value of MOD, i.e., the modifiee's *synsem*), ECONT (the combinatorial CONTENT of the phrase which contains the adjunct and its modifiee), and ICONT (the CONTENT of the adjunct's maximal projection). In addition, Kasper retains the feature CONT as the content of the lexical item itself. In this manner, for example, predicative and attributive adjectives share the same CONTENT, although the attributive adjective will have a MOD value containing ARG, ECONT, and ICONT, which the predicative adjective does not share.

The last elements of the analysis are the syntactic and semantic combinatorics. The head-adjunct schema is changed such that the MOD|ARG (rather than MOD) value of the the adjunct daughter is token identical to the SYNSEM value of the head daughter. The Semantics Principle is also changed, but it is still disjunctive (Kasper, 1997). Setting quantifiers aside, for a head-adjunct phrase, the CONTENT is token identical to the MOD|ECONT of the adjunct daughter and the MOD|ICONT value of the adjunct daughter is token-identical with the adjunct daughter's CONTENT; for any other headed phrase, the CONTENT of the phrase is token-identical to the CONTENT of the head daughter.

Kasper's (1997) system does correct both problems noted above: recursive modification is handled properly, and lexical ambiguity is not multiplied out to the modifiers of adjectives. However, it comes at the cost of a tailor-made modification of HPSG theory and the corresponding feature geometry. The glue approach we present below requires no further modifications of HPSG, beyond those necessary to provide a general glue semantics for HPSG.

The recursive modification and multiplication of lexical ambiguity problems do not arise in the glue approach, which readily allows for nested modification. Let us assume the partial lexical entries in (31) for *apparently*, attributive *smart*, predicative *smart*, and the entry for *student* as in (13) above.[2]

(31) a.

$$\begin{bmatrix} \textbf{“APPARENTLY”} & \\ \text{CAT} & \begin{bmatrix} \text{HEAD} \; \begin{bmatrix} \text{MOD} \;\; \text{A:}\begin{bmatrix} \text{HEAD-RES} \;\; \boxed{1} \end{bmatrix} \end{bmatrix}_{\textit{adverb}} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{HEAD-RES} \;\; \boxed{2} \end{bmatrix} \\ \text{GLUE} & \left\langle \begin{array}{l} \text{apparent:}\boxed{2}, \\ \lambda P.\lambda Q.P(Q):\boxed{2} \multimap (\boxed{1} \multimap \boxed{1}) \end{array} \right\rangle \end{bmatrix}$$

b.

$$\begin{bmatrix} \textbf{“SMART”} & \\ \text{CAT} & \begin{bmatrix} \text{HEAD} \; \begin{bmatrix} \text{MOD} \;\; \text{N}'\begin{bmatrix} \text{CONT} \;\; \begin{bmatrix} \text{VAR-RES} & \boxed{2} \\ \text{RESTR-RES} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}_{\textit{adj}} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{HEAD-RES} \;\; \boxed{1} \end{bmatrix} \\ \text{GLUE} & \left\langle \begin{array}{l} \text{smart:}\boxed{1} \; \textit{e-resource}, \\ \lambda P.\lambda Q.\lambda x.[P(x) \wedge Q(x)]:\boxed{1} \multimap ((\boxed{2} \multimap \boxed{3}) \multimap (\boxed{2} \multimap \boxed{3})) \end{array} \right\rangle \end{bmatrix}$$

c.

$$\begin{bmatrix} \textbf{“SMART”} & \\ \text{CAT} & \begin{bmatrix} \text{HEAD} \; \begin{bmatrix} \text{PRD} \;\; + \end{bmatrix}_{\textit{adj}} \end{bmatrix} \\ \text{VALENCE} \mid \text{SUBJ} & \langle \text{NP:}\begin{bmatrix} \text{HEAD-RES} \;\; \boxed{2} \end{bmatrix} \rangle \\ \text{CONT} & \begin{bmatrix} \text{HEAD-RES} \;\; \boxed{1} \end{bmatrix} \\ \text{GLUE} & \left\langle \begin{array}{l} \text{smart:}\boxed{3} \; \textit{e-resource}, \\ \lambda P.\lambda x.P(x):\boxed{3} \multimap (\boxed{2} \multimap \boxed{1}) \end{array} \right\rangle \end{bmatrix}$$

[2] The *e-resource* with the meaning *smart* should strictly speaking be replaced by an implication from an *e-resource* to a *t-resource*, since *smart* is a constant of type $\langle e, t \rangle$. For ease of exposition we simplify matters by treating it as a constant of type e.

Via two applications of the head-adjunct schema, the lexically specified RESOURCEs are instantiated, and the GLUE value of the N′ *apparently smart student* is a list with these elements:[3]

(32) apparent:$\boxed{4}$, λP.λQ.P(Q):$\boxed{4} \multimap (\boxed{5} \multimap \boxed{5})$,
smart:$\boxed{5}$, λP.λQ.λx.[P(x) $\wedge$ Q(x)]:$\boxed{5} \multimap ((\boxed{6} \multimap \boxed{7}) \multimap (\boxed{6} \multimap \boxed{7}))$,
student:$\boxed{6} \multimap \boxed{7}$

This derives a meaning for *apparently smart student* as follows:

(33)

$$\dfrac{\dfrac{\dfrac{\dfrac{apparent : \boxed{4} \qquad \lambda P.\lambda Q.P(Q) : \boxed{4} \multimap (\boxed{5} \multimap \boxed{5})}{\lambda Q.apparent(Q) : \boxed{5} \multimap \boxed{5}} \qquad smart : \boxed{5}}{apparent(smart) : \boxed{5}} \qquad \lambda P.\lambda Q.\lambda x.[P(x) \wedge Q(x)] : \boxed{5} \multimap ((\boxed{6} \multimap \boxed{7}) \multimap (\boxed{6} \multimap \boxed{7}))}{\lambda Q.\lambda x.[(apparent(smart))(x) \wedge Q(x)] : (\boxed{6} \multimap \boxed{7}) \multimap (\boxed{6} \multimap \boxed{7})} \qquad student : \boxed{6} \multimap \boxed{7}}{\lambda x.[apparent(smart)(x) \wedge student(x)] : \boxed{6} \multimap \boxed{7}}$$

As required, *apparent* modifies *smart*, and then *apparently smart* modifies *student*.

Note that this analysis of recursive modification applies equally to adjectives in predicative position, such as *Kim is apparently smart*. Predicative adjectives differ from atrributive ones in replacing the attributive premise, which applies the adjective meaning to the N′ meaning, by a predicative premise that applies the adjective meaning to that of the subject's HEAD-RESOURCE. This gives rise to the premises

(34) apparent:$\boxed{4}$, λP.λQ.P(Q):$\boxed{4} \multimap (\boxed{5} \multimap \boxed{5})$,
smart:$\boxed{5}$, λP.λx.P(x):$\boxed{5} \multimap (\boxed{6} \multimap \boxed{7})$,
kim:$\boxed{6}$

from which there is a derivation

(35)

$$\dfrac{\dfrac{\dfrac{\dfrac{apparent : \boxed{4} \qquad \lambda P.\lambda Q.P(Q) : \boxed{4} \multimap (\boxed{5} \multimap \boxed{5})}{\lambda Q.apparent(Q) : \boxed{5} \multimap \boxed{5}} \qquad smart : \boxed{5}}{apparent(smart) : \boxed{5}} \qquad \lambda P.\lambda x.P(x) : \boxed{5} \multimap (\boxed{6} \multimap \boxed{7})}{\lambda x.(apparent(smart))(x) : (\boxed{6} \multimap \boxed{7})} \qquad kim : \boxed{6}}{apparent(smart)(kim) : \boxed{7}}$$

4.3 Raising and De Dicto Scope

Consider the following raising sentence:

(36) A unicorn seemed to approach.

Pollard and Sag (1994:328, fn. 3) observe that their Semantics Principle does not allow for a de dicto interpretation of this sentence in which the raising verb takes scope over the quantifier.

Asudeh (2000, to appear) provides a glue analysis of raising verbs, which we adopt here. The relevant part of the lexical entry for the verb *seem* is as follows:

[3] To avoid confusion we have given the resources new numbers.

(37)
$$\begin{bmatrix} \textbf{“SEEM”} & \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VALENCE} & \begin{bmatrix} \text{SUBJ} & \langle \boxed{3} \rangle \\ \text{COMPS} & \langle \text{VP}[\textit{inf}, \text{SUBJ } \langle \boxed{3}\text{NP}\rangle]{:}\boxed{0} \rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & {}_{\textit{seem}}\begin{bmatrix} \text{HEAD-RES} & \boxed{1} \\ \text{SOA-ARG} & \boxed{0}\begin{bmatrix} \text{HEAD-RES} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{GLUE} & \langle \text{seem}{:}\boxed{2} \multimap \boxed{1} \rangle \end{bmatrix}$$

This entry treats *seem* as denoting a one-place predicate with a propositional argument. The raising verb gives its own meaning by consuming its infinitival complement's meaning. The quantified noun phrase *a unicorn* and the verb *approach* contribute their usual glue premises. These are all passed up to the root node along with the glue for *seem*.

The glue premises for sentence (36) are shown here:

(38)
$$\begin{bmatrix} \text{CONT} \mid \text{HEAD-RES } \boxed{1} & \\ \text{GLUE} & \left\langle \begin{array}{l} \lambda Q.\forall x[\text{unicorn}(x) \rightarrow Q(x)]{:}(\boxed{4} \multimap G) \multimap G, \\ \text{seem}{:}\boxed{2} \multimap \boxed{1}, \\ \text{approach}{:}\boxed{4} \multimap \boxed{2} \end{array} \right\rangle \end{bmatrix}$$

There are two possible derivations, depending on whether the quantified NP's scope resource G is instantiated to $\boxed{2}$ (the de dicto reading) or $\boxed{1}$ (the de re reading).

(39) a. **De Dicto**

$$\dfrac{\dfrac{(\boxed{4} \multimap G) \multimap G \qquad \boxed{4} \multimap \boxed{2}}{\boxed{2}}\multimap_{\mathcal{E}}, \text{G} = \boxed{2} \qquad \boxed{2} \multimap \boxed{1}}{seem(\exists x.[unicorn(x) \wedge approach(x)]) : \boxed{1}}\multimap_{\mathcal{E}}$$

b. **De Re**

$$\dfrac{(\boxed{4} \multimap G) \multimap G \qquad \dfrac{\dfrac{\dfrac{\boxed{4} \multimap \boxed{2} \qquad [\boxed{4}]^1}{\boxed{2}}\multimap_{\mathcal{E}} \qquad \boxed{2} \multimap \boxed{1}}{\boxed{1}}\multimap_{\mathcal{E}}}{\boxed{4} \multimap \boxed{1}}\multimap_{\mathcal{I},1}}{\exists x.[unicorn(x) \wedge seem(approach(x))] : \boxed{1}}\multimap_{\mathcal{E}}, \text{G} = \boxed{1}$$

Pollard and Yoo (1998) present an alternative solution to the de dicto scope problem mentioned here. Their solution rests on a modification to the HPSG feature geometry of Pollard and Sag (1994), such that QSTORE is a *local* feature, rather than a feature of *sign*. They retain RETRIEVED as a *sign* feature, and add a feature POOL, which is the union of QSTORE and RETRIEVED.[4] In particular, a word's POOL contains the QSTORE values of its dependents, according to the following assumption (Pollard and Yoo, 1998:421):

(40) The POOL is the union of the QSTOREs of all *selected arguments*, defined as either

a. thematic elements selected via the SUBJ or COMPS feature,

b. elements selected via the SPR feature, or

c. elements selected via the MOD feature.

[4] The value of RETRIEVED is actually a list, since order matters, but Pollard and Yoo (1998:420) provide a function set-of-elements that returns the contents of a list as a set.

Pollard and Yoo (1998:421) note that, “This assumption, together with the new assumption that QSTORE is an attribute of the LOCAL value, has as a consequence that a quantifier in the QSTORE of a raising controller will appear in the POOL of the head of the controlled complement.” With these modifications, they are able to scope the quantifier under the raising verb, by storing it in the QSTORE of the controlled complement (e.g., *approach* in (36) above).

We believe there are two theoretical reasons to prefer our approach to the Pollard and Yoo (1998) approach. First, their approach retains much of the disjunctive Semantics Principle from Pollard and Sag (1994), which our approach simplifies substantially, as shown above. Furthermore, they need the addition of (40), which is another disjunctive definition. Their approach needs these complications because they are attempting to calculate semantic composition and semantic interpretation using the same representations. Our approach simplifies things by using a well-understood logic of composition that is related to, but separate from, the interpretation language. In other words, we believe our approach to be essentially simpler than Pollard and Yoo’s (1998): both require adjustments to the feature geometry, but the latter approach also requires several disjunctive definitions and additional stipulations. Once the glue system is in place and the correspondence between the underlying HPSG grammar and the glue language is defined, the relevant scope ambiguities follow purely from linear logic deduction.

4.4 Control Theory

We presuppose a syntax for control that is very similar to that of Pollard and Sag (1994). In particular, a control verb selects for an unsaturated infinitival VP complement. We assume a modified version of Pollard and Sag’s (1994) control theory:

(41) **Control Theory**
If the CONTENT of an unsaturated phrase is the SOA-ARG in a control relation, then the subject of that phrase is coindexed with the INFLUENCED, COMMITOR, or EXPERIENCER value in the control relation, according as the control relation is of type *influence*, *commitment*, or *orientation*, respectively.

We do not assume that the subject of the controlled complement must be a reflexive, as the locality of control is guaranteed by the locality of semantic role selection in (41).

We assume the following basic lexical entry for “tried”:

(42)
$$
\left[\begin{array}{ll}
\textbf{“TRIED”} & \\
\text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \left[\text{VFORM} \;\; \textit{fin}\right] \\ \text{VALENCE} & \left[\begin{array}{ll} \text{SUBJ} & \langle \text{NP}[\text{HEAD-RES } \boxed{1}\,] \rangle \\ \text{COMPS} & \langle \text{VP}[\textit{inf}, \text{SUBJ } \langle \text{NP}[\text{HEAD-RES } \boxed{2}\,]\rangle]{:}\boxed{3} \rangle \end{array}\right] \end{array}\right] \\
\text{CONT} & {}_{\textit{try}}\left[\begin{array}{ll} \text{HEAD-RES} & \boxed{0} \\ \text{COMMITTOR} & \boxed{1} \\ \text{SOA-ARG} & \boxed{3}\left[\text{HEAD-RES} \;\; \boxed{4}\right] \end{array}\right] \\
\text{GLUE} & \langle \text{try}{:}(\boxed{2} \multimap \boxed{4}) \multimap (\boxed{1} \multimap \boxed{0}) \rangle
\end{array}\right]
$$

By the control theory in (41), the resource $\boxed{1}$ that is the value of *try’s* COMMITTOR is coindexed with the subject of the control verb’s unsaturated VP complement. As HEAD-RESOURCE has taken over the function of INDEX in our system, this means that the HEAD-RES values of the committor and the controlled subject are structure-shared. In other words, $\boxed{1}$ and $\boxed{2}$ are identified ($\boxed{1} = \boxed{2}$), and we get the following full lexical entry for “tried”:

(43)
$$
\left[\begin{array}{ll}
\textbf{“TRIED”} & \\
\text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \left[\text{VFORM}\ \ \textit{fin}\right] \\ \text{VALENCE} & \left[\begin{array}{ll}\text{SUBJ} & \langle \text{NP[HEAD-RES } \boxed{1}\,]\rangle \\ \text{COMPS} & \langle \text{VP}[\textit{inf}, \text{SUBJ } \langle \text{NP[HEAD-RES } \boxed{1}\,]\rangle]{:}\boxed{3}\,\rangle \end{array}\right]\end{array}\right] \\
\text{CONT} & {}_{\textit{try}}\left[\begin{array}{ll} \text{HEAD-RES} & \boxed{0} \\ \text{COMMITTOR} & \boxed{1} \\ \text{SOA-ARG} & \boxed{3}\left[\text{HEAD-RES}\ \ \boxed{4}\right]\end{array}\right] \\
\text{GLUE} & \langle \text{try}{:}(\boxed{1} \multimap \boxed{4}) \multimap (\boxed{1} \multimap \boxed{0})\rangle
\end{array}\right]
$$

The glue side of the meaning constructor for the control verb is an implication. The antecedent of this implication is itself an implication corresponding to the controlled VP complement, i.e., the normal single implication between a subject and VP meaning for an intransitive verb. The consequent is also an implication, taking the resource of the control verb's subject as its antecedent and the control verb's resource as its consequent. (Asudeh, 2000, to appear).

The account of control, agreement and binding in (Pollard and Sag, 1994) gave an elegant explanation for the necessity of agreement between a reflexive object in a controlled complement and the controller:

(44) She tried to criticize herself/*himself/*myself/*themselves.

Control theory requires coindexation between the controller and a (reflexive) subject of the unsaturated VP control complement. Agreement is also represented by coindexation. Thus, the coindexation due to control theory entails agreement of PERSON/NUMBER/GENDER (PNG) features between the controller and the controllee. Binding is also represented using coindexation, and the binding theory requires that the object reflexive be bound by the dominating controlled subject. Due to the separately motivated theories of control, binding, and agreement, and the common mechanism of coindexation of PNG-articulated indices, there is agreement between the controller, the controllee, and the reflexive object.[5]

The account given here derives the same result, essentially in the same manner, since *e-resources* do double duty as indices. However, Pollard and Sag (1994) take the agreement facts as evidence that the denotation of the controlled clause is a proposition. A key difference in our approach is that it does not fix the denotation of the control verb complement as a proposition. Chierchia (1984a,b) has argued convincingly that the denotation of the complement is a property. Pollard and Sag (1994:283) point out that

> One crucial aspect of the property-based theory of control is that there is no direct link between the semantic argument position of the controller and that of the unexpressed subject of the controlled complement; the only relation between the two argument positions is inferential.

Based on this, they conclude that the property-based theory of control complement denotation requires a purely semantic approach to agreement (Dowty and Jacobson, 1989), which they argue against (Pollard and Sag, 1994:71–73).

On the glue approach sketched here, this criticism does not hold. As we have argued, this analysis, like that of Pollard and Sag (1994) guarantees the agreement facts in (44). However, with the same glue language in the meaning constructor, we can have either a property or propositional denotation for the control verb's complement, as it is the meaning language side of the meaning constructor that gives the denotation. This can be a property, as in (45), or a proposition, as in (46).

(45) $\lambda P \lambda x.\text{try}(\text{x, P}){:}(\boxed{1} \multimap \boxed{4}) \multimap (\boxed{1} \multimap \boxed{0})$

[5] Pollard and Sag (1994) also derive Visser's generalization, that subject control verbs cannot be passivized, from this interplay; but see Asudeh (1998) for a critical discussion.

(46) $\lambda P \lambda x$.try(x, P(x)):($\boxed{1} \multimap \boxed{4}$) $\multimap$ ($\boxed{1} \multimap \boxed{0}$)

We take it that this is an advantage of our analysis, because this allows the decision regarding the proper denotation of controlled complements to be based on empirical data, such as the inference patterns that Chierchia (1984a,b) presents, while allowing facts about the syntax-semantics interface (e.g., agreement) to be stated simply.

5 Discussion

This paper has outlined how to go about providing a glue semantics for HPSG. The main effort is to change HPSG's feature geometry in order to construct collections of glue premises. The changes we have so far considered all result in greatly simplified feature structures.

It is important to note that the glue premises derived from HPSG analyses are essentially the same as those from LFG analyses, around which glue semantics was originally constructed. Although we have gone to some lengths in this paper to show how glue derivations account for various phenomena — like scope ambiguity, recursive modification and control — none of this material is in fact original: it is all taken from previous work on glue semantics for LFG. We see this lack of originality as a strength. It illustrates how glue analyses can be ported from one grammatical framework to another. In a computational setting, this means that efficient glue proof search algorithms implemented for use with LFG can be directly re-used with HPSG.

Of course, showing that you *can* give a glue semantics for HPSG is not the same as showing that you *should* give one. There are, after all, other ways of doing semantics for HPSG which also provide analyses for the kinds of phenomena discussed in this paper. Two important questions, which we can only begin to discuss in this section, are therefore: (1) in what ways does glue semantics genuinely differ from other proposals for HPSG semantics, particularly Minimal Recursion Semantics (MRS), and (2) to the extent that glue really does differ, are there any reasons for preferring or dispreferring it?

5.1 A Comparison to Minimal Recursion Semantics

Minimal Recursion Semantics (Copestake et al., 1999) builds up a flat description of a logical form. The description comprises a bag of labelled elementary predications. Arguments to predicates may either be logical variables, or handles which either label other elementary predications, or can be related to such labels by scope constraints.

One way of thinking about the resource-denoting atomic propositions in glue semantics is to view them as handles (i) labelling elementary predications in the case of *t-resources*, or (ii) labelling individual constants or variables in the case of *e-resources*. Moreover in glue, there is a partial order over atomic resources reflecting the order in which these resources are consumed in a given glue derivation, which determines scope ordering (Crouch and van Genabith, 1999). The use of partial orders over glue resources to constrain scope relations further heightens the similarity between glue and MRS, where a corresponding scope order over labels is assumed. Overall, the bag of glue premises and scope constraints produced by glue semantics bears some similarity to the bag of elementary predications and scope constraints of MRS.

However, there are differences between glue and MRS. Perhaps the most significant is that MRS is explicitly set up to describe logical forms. There is a long tradition in semantics, dating back to Montague's eliminable level of intensional logic, that regards this as a suspect activity. The basis of this tradition is that logical forms have no relevant, non-trivial identity criteria other than those given by their model theoretic semantics. The logical forms themselves are just arbitrary bits of syntactic notation. It is hard to tell when constraints on logical forms capture constraints on the underlying semantic objects they represent, and when they merely constrain arbitrary features of the chosen logical representation.

Glue semantics does not run into this problem. To the extent that glue premises and scope constraints describe anything, it is the *composition* of a semantic object, and not the form of syntax that happens to be used to represent it. The semantic composition is represented as a linear logic derivation. There is a strict separation between linear logic formulas and the meaning terms attached to them, and this guarantees that the logical derivation / semantic composition is independent of syntactic idiosyncracies of the meaning representation language. Moreover, results in proof theory (e.g., proof normalization, Curry-Howard Isomorphism) establish that glue derivations have precisely the kind of non-trivial identity criteria required of genuine objects ("no entity without identity").

Another difference between glue and MRS is commitment to the λ-calculus. The principal operations of (glue) semantic composition are function application and λ-abstraction, as represented by the rules of implication elimination and introduction. MRS is at some pains to avoid use of the λ-calculus. However, the criticisms of Montague's use of the λ-calculus made by Copestake et al. (2001) do not apply to glue. First, there is no requirement that arguments to predicates be provided in a fixed order, since the rules implication elimination and introduction allow arguments to be re-ordered. Type raising is not an additional complexity, but emerges directly from the interplay of these two inference rules. Finally, the resource sensitivity of linear logic enforces a desirable monotonicity of interpretation: the semantics of a mother cannot just ignore the semantics of one of its daughters.

5.2 Conclusion

The positive aspects of glue for HPSG include:

1. Simplification of HPSG feature geometry and semantic principles
2. Accounts of a wide range of semantic phenomena following solely from the interplay of the standard, and simple, rules of inference for implication elimination and implication introduction. These accounts do not require additional formal machinery, further constraints on the possible form of MRS constraints, etc.
3. Semantic compositions / glue derivations as first class objects of semantic theory
4. Re-use of efficient linear logic proof techniques

The last point may need some explanation. Even though the glue logic we have used is just the implicational fragment of propositional linear logic, the general complexity properties of this logic are not ideal. However, Gupta and Lamping (1998) observed that linguistically natural input results in glue derivations containing many sub-derivations of modifiers or identities of the general form $\phi \multimap \phi$. Special purpose proof search techniques have been devised to separate out these modifier subderivation in an initial phase, and then interpolate them back into the main skeleton derivation. Packing techniques allow one to efficiently generate and represent all ways of permuting these modifiers, leading to a compact representation of all possible semantic ambiguities.

There are (at least) two reasons why one might be wary of glue for HPSG. The first is that to date there has been no practical exploration of reversibility and generation with glue semantics. Theoretical work discussing machine translation (van Genabith et al., 1998) indicates that in theory reversibility is readily obtainable: the monotonicity of interpretation induced by resource sensitivity helps to ensure this. But this theoretical observation has not yet been subjected to algorithmic scrutiny.

The second reason is that glue semantics carries some of the flavour of LFG, in using different sub-logics to construct different types of linguistic representation in a parallel, but mutually constraining fashion. The sign-based approach of HPSG might instead favour encoding the glue linear logic as part of a general logic for talking about AVMs. There are thus several avenues for future research.

References

Asudeh, Ash (1998). *Anaphora and argument structure: Topics in the syntax and semantics of reflexives and reciprocals*. M.Phil. thesis, University of Edinburgh.

Asudeh, Ash (2000). Functional identity and resource-sensitivity in control. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the LFG 2000 conference*, Stanford, CA. CSLI Publications. http://csli-publications.stanford.edu/LFG/.

Asudeh, Ash (to appear). A resource-sensitive semantics for equi and raising. In David Beaver, Stefan Kaufmann, Brady Zack Clark, and Luis Casillas (eds.), *Proceedings of Semantics Fest 2000*. Stanford, CA: CSLI Publications.

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

Barwise, Jon, and John Perry (1983). *Situations and attitudes*. Cambridge, MA: MIT Press.

Chierchia, Gennaro (1984a). Anaphoric properties of infinitives and gerunds. In Mark Cobler, Susannah MacKaye, and Michael Wescoat (eds.), *Proceedings of the third West Coast Conference on Formal Linguistics*, (pp. 28–39), Stanford, CA. Stanford Linguistics Association.

Chierchia, Gennaro (1984b). *Topics in the syntax and semantics of infinitives and gerunds*. Ph.D. thesis, University of Massachusetts, Amherst.

Copestake, Ann, Dan Flickinger, Robert Malouf, Susanne Riehemann, and Ivan A. Sag (1995). Translation using Minimal Recursion Semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*, Leuven, Belgium.

Copestake, Ann, Dan Flickinger, Ivan A. Sag, and Carl Pollard (1999). Minimal Recursion Semantics: An introduction. Ms., Stanford University.

Copestake, Ann, Alex Lascarides, and Dan Flickinger (2001). An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th annual meeting of the association for computational linguistics (acl/eacl 2001)*, (pp. 132–139), Toulouse, France.

Crouch, Richard, Anette Frank, and Josef van Genabith (1999). Glue, underspecification and translation. In H Bunt and Reinhard Muskens (eds.), *Computing meaning*, vol. 2. Dordrecht: Kluwer.

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

Curry, H B, and R Feys (1961). *Combinatory logic*. Amsterdam: North-Holland.

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.

Dowty, David, and Pauline Jacobson (1989). Agreement as a semantic phenomenon. In Joyce Powers and Kenneth de Jong (eds.), *Proceedings of the fifth eastern states conference on linguistics*, Columbus, OH: Ohio State University.

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 Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the LFG 2001 conference*, Stanford, CA. CSLI Publications. http://csli-publications.stanford.edu/LFG/.

Gupta, Vaneet, and John Lamping (1998). Efficient linear logic meaning assembly. In *Coling-ACL '98*.

Kasper, Robert T. (1997). The semantics of recursive modification. Ms., Ohio State University.

Pereira, Fernando (1990). Categorial semantics and scoping. *Computational Linguistics*, *16*, 1–10.

Pollard, Carl, and Ivan A. Sag (1994). *Head-Driven Phrase Structure Grammar*. Chicago, IL and Stanford, CA: The University of Chicago Press and CSLI Publications.

Pollard, Carl, and Eun Jung Yoo (1998). A unified theory of scope for quantifiers and wh-phrases. *Journal of Linguistics*, *34*, 415–442.

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

van Genabith, Josef, Anette Frank, and Michael Dorna (1998). Transfer constructors. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of lfg '98*, (pp. 190–205), Stanford, CA. CSLI. http://www-csli.stanford.edu/publications/.