Functional Identity and Resource-Sensitivity in Control

Ash Asudeh

Stanford University and Xerox PARC

Proceedings of the LFG00 Conference

University of California, Berkeley

Miriam Butt and Tracy Holloway King (Editors)

2000

CSLI Publications

http://csli-publications.stanford.edu/

1 Introduction[1]

Glue semantics provides a semantics for Lexical Functional Grammar (LFG) that is expressed using linear logic (Girard, 1987; Dalrymple, 1999) and provides an interpretation for the f(unctional)-structure level of syntactic representation, connecting it to the level of s(emantic)-structure in LFG's parallel projection architecture (Kaplan, 1987, 1995). Due to its use of linear logic for meaning assembly, Glue is resource-sensitive: semantic resources contributed by lexical entries and resulting f-structures must each be used in a successful proof *exactly once*. In this paper, I will examine the tension between a resource-sensitive semantics which interprets f-structures and structure-sharing in f-structures as expressed by functional control resulting from lexical functional identity equations. The empirical phenomenon I concentrate on is equi, also known as obligatory control.[2] Although at first blush it seems that structure-sharing poses a serious problem for Glue semantics, I will show that this is not so. In fact, this tension leads to a very restrictive theory, and the analysis I present here solves several long-standing problems in the semantics of equi, by exploiting LFG's grammatical architecture.

In this paper I will:

1. Give a Glue semantics for equi.
2. Show how the analysis can yield either a propositional or property denotation for the clausal complement of an equi verb. This flexibility arises naturally from the architecture of the theory.
3. Adopt the property theory of equi complements, which has been argued for independently by Chierchia (1984).
4. Counter previous objections to the property theory from anaphoric binding and typological data by exploiting LFG's architecture.
5. Argue that a previous solution to the structure-sharing/resource-sensitivity problem (Kehler et al., 1999) can only yield a propositional denotation. The Kehler et al. proposal is rejected for empirical and theoretical reasons.
6. Show several empirical predictions of the analysis.

2 Glue Semantics

2.1 Motivation

There are two basic conceptual motivations for Glue, beside the empirical motivation provided by successful Glue analyses of semantic phenomena (Dalrymple, 1999). First, the resource-sensitivity of Glue semantics

[1] I owe a great debt to Dick Crouch and Mary Dalrymple at PARC for a lot of feedback, free exchange of ideas, and a great working environment. They should accept much of the credit for this work, but none of the blame. I'd also like to thank the following people for their comments: David Beaver, Joan Bresnan, Daniel Büring, Ron Kaplan, Tracy Holloway King, Hanjung Lee, John Maxwell, Dave McKercher, Line Hove Mikkelsen, Yukiko Morimoto, Ivan Sag, Peter Sells, Ida Toivonen, and audiences at Stanford's Semantics Fest and LFG 2000. I accept full responsibility for any remaining errors. A special thanks to Jim McCloskey for getting me access to facilities at UCSC. This research was supported partly by SSHRC Doctoral Fellowship 752-98-0424.

[2] Throughout this paper I will use the term 'equi', even though the term 'control' is more common in the literature. I do this to avoid confusion between the data being described and the theoretial construct — functional control — that models it.

directly models the resource-sensitivity of natural languages. The words and sentences in a natural language utterance do not make multiple contributions to the utterance's meaning unless they are used multiple times.

Second, the resource-sensitivity of Glue means that completeness and coherence follow from the formalism and do not need to be stipulated as separate principles (Dalrymple et al., 1999a).[3]

- **Completeness**: If any of the Glue resources a predicate needs for its semantics are not present, *there is no successful Glue proof*.
  (Instead of: "every function designated by a PRED [must] be present in the f-structure of that PRED." (Bresnan, 2000: 72))

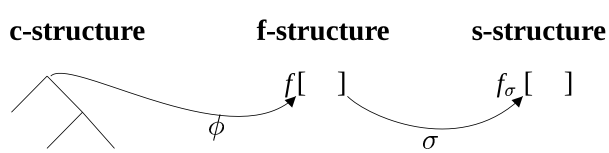
- **Coherence**: If a GF is present and contributing a resource which no predicate's meaning will consume, *there is no successful Glue proof*.
  (Instead of: "every argument function in an f-structure [must] be designated by a PRED." (Bresnan, 2000: 73))

Note that by "successful Glue proof" I mean one that consumes the semantics of the lexical items making up a sentence to provide the sentential meaning. Thus, PRED values in f-structures do not need to be of the form 'pred-name$\langle \ldots \rangle \ldots$' and are instead of the form 'pred-name'. Although completeness and coherence were originally proposed as purely syntactic constraints (Kaplan and Bresnan, 1982) and continue to be formulated as such (Bresnan, 2000), if they can be derived from the formal architecture of the theory, they clearly do not need to be stated as theoretical primitives.

2.2 Glue and the Parallel Projection Architecture of LFG

LFG has a grammatical architecture in which various levels of grammatical representations are simultaneously present, but each level is governed by its own rules and representations. The various levels are then related to each other by mapping functions, which map representations at one level to those in another (Kaplan, 1987, 1995).

Just as the level of c(onstituent)-structure is mapped to f-structure by the ϕ function, f-structure is mapped to s-structure by the σ function. Glue semantics is a theory of the σ-mapping and of s-structure.

(1)


This separation of levels allows one to make simple theoretical statements about the aspet of grammar that the level in question models. Phrase structure, constituency, domination and linear order are represented at c-structure using trees, while grammatical functions, subcategorization, binding, control, and various other aspects of syntax are represented at f-structure using attribute-value matrices. Semantics and the relationship between syntax and meaning are represented at s-structure using Glue: a combination of linear logic and a chosen meaning language.

[3]Expletives do not contribute resources and present a potential complication for this reduction of completeness and coherence.

An important feature of this architecture is that there can be systematic mismatches between grammatical levels. For example, null pronoun subjects in pro-drop languages are not present at c-structure, because they are unmotivated by the syntactic phenomena represented at that level. Rather, null pronouns are present at f-structure, where they can participate in agreement, binding, and other syntactic processes best represented at that level. Similarly, there can be systematic mismatches between f-structure and s-structure, and it is this aspect of the architecture that allows for an adequate semantics of equi that nevertheless does not conflict with structure-sharing in the syntax and in fact uses the syntax to give solutions to certain problems in previous analyses of equi semantics.

2.3 Overview of Glue Semantics

Glue uses two logics: a meaning logic for representing meaning terms, and linear logic (Girard, 1987) for assembling meanings Dalrymple (1999). As already stated, linear logic is resource-sensitive: a linear logic proof is valid only if all premises are used exactly once. This is best exemplified by comparing propositional logic to propositional linear logic and observing the differences in certain entailment patterns.

1. **A premise can only be used once**

(2) Propositional logic implication ($\rightarrow$)

a. $p, p \rightarrow q \vdash q$

b. $p, p \rightarrow q \vdash p \wedge q$ p used to derive q and can be conjoined with q

(3) Propositional linear logic implication ($\multimap$)

a. $p, p \multimap q \vdash q$

b. $p, p \multimap q \nvdash p \otimes q$ p was used up to derive q

2. **Each premise must be used**

(4) Propositional logic conjunction ($\wedge$)

a. $p \wedge q \vdash p$ q ignored

(5) Propositional linear logic conjunction ($\otimes$)

a. $p \otimes q \nvdash p$ must use q

In principle, we can choose any logic for the meaning logic, so long as a systematic relationship can be established between operations in the meaning language and those in the Glue language (linear logic).

2.4 New Glue

Recent work in Glue semantics has used the Curry-Howard (C-H) isomorphism to directly relate the Glue and meaning languages (Dalrymple et al., 1999b). According to the C-H isomorphism, introduction of implication in the Glue language corresponds to lambda abstraction in the meaning language and elimination of implication corresponds to function application. I will use only the implication fragment of linear logic in this analysis, and will present my Glue proofs in the natural deduction (ND) style. The ND proof rules for introduction and elimination of implication are as follows:

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

$$\frac{A \multimap B \qquad A}{B} \multimap_{\mathcal{E}} \qquad\qquad \frac{\begin{matrix}[A]^i \\ \vdots \\ B\end{matrix}}{A \multimap B} \multimap_{\mathcal{I}}, i$$

The elimination rule is just modus ponens. The implication rule involves flagging an assumption in square brackets, and subsequently discharging this assumption if it has been used to prove another premise. In this case [A]^i is used to derive B, and we can discharge the assumption using implication introduction to get A $\multimap$ B.

The following simple example shows the natural deduction rules and Curry-Howard isomorphism working together to prove that $a \multimap b \vdash a \multimap b$. The meaning language appears on the left of the uninterpreted symbol ':' and the linear logic is on the right.

(7)

$$\cfrac{\cfrac{[x : a]^i \qquad P : a \multimap b}{P(x) : b}\, function\ application : \multimap_{\mathcal{E}}}{\lambda x.P(x) : a \multimap b}\, lambda\ abstraction : \multimap_{\mathcal{I},i}$$

In the first step, $x : a$ is assumed (indicated by square brackets) and the assumption is flagged with the superscript i. 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.

Glue with the C-H isomorphism has advantages over previous Glue formalizations:

1. It eliminates the need for higher-order unification.
2. The meaning and Glue languages are kept completely separate, such that a proof cannot fail simply due to failure in the meaning language, only due to failure in the Glue language.
3. By examining operations in the Glue language, we automatically know the corresponding operation in the meaning language.

To stay within the implicational fragment, Glue conjunction ($\otimes$) in the antecedent of an implication will be cashed out as implication, by the following equivalence (which also holds for ordinary (non-linear) propositional logic):

(8) $(a \otimes b) \multimap c \equiv b \multimap (a \multimap c)$

For example, the Glue semantics for a transitive verb can be written using a conjunction, such that the verb consumes the resources of its subject and object to give its meaning, or it can be written such that the verb consumes its object's resource and then its subject's resource to give its meaning:

(9) An example: transitive verbs
$((\uparrow \text{SUBJ})_\sigma \otimes (\uparrow \text{OBJ})_\sigma) \multimap \uparrow_\sigma \equiv (\uparrow \text{OBJ})_\sigma \multimap ((\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma)$

Using the equivalence in (8) with the equivalence in (10), the subject can be consumed first, as in (11).

(10) $(a \otimes b) \equiv (b \otimes a)$

(11)
$$\begin{aligned} & (\uparrow \textsc{obj})_\sigma \multimap ((\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma) \\ \equiv\ & ((\uparrow \textsc{subj})_\sigma \otimes (\uparrow \textsc{obj})_\sigma) \multimap \uparrow_\sigma \\ \equiv\ & ((\uparrow \textsc{obj})_\sigma \otimes (\uparrow \textsc{subj})_\sigma) \multimap \uparrow_\sigma \\ \equiv\ & (\uparrow \textsc{subj})_\sigma \multimap ((\uparrow \textsc{obj})_\sigma \multimap \uparrow_\sigma) \end{aligned}$$

Although this introduction to Glue semantics has been necessarily short, I have presented everything that I use in my analysis (sections 5 and 6). But first let us consider the problem that functional control presents for a resource-sensitive semantics.

3 The Problem

Despite its advantages in modelling natural language meaning, the resource-sensitivity of Glue initially seems to be at odds with functional control in f-structures, which is expressed by functional identity equations[4] in lexical f-descriptions. These equations result in structure-sharing (i.e. token identity) of f-structures. The σ mapping function is must map this one stucture-shared f-structure to one node in s-structure, or else the mapping function is not a function by definition. In other words, the problem is that if an f-structure is structure-shared, then it can only produce one semantic-resource.

Consider the following equi example:

(12) Gonzo tried to go.

I am assuming a functional control analysis of English obligatory control, whereby the equi verb's SUBJ or OBJ functionally controls and is therefore token identical to its XCOMP SUBJ.[5] Simplifying somewhat, we get the following f-structure for (12).

(13)
$$f\begin{bmatrix} \textsc{pred} & \text{'try'} \\ \textsc{subj} & g\begin{bmatrix} \textsc{pred} & \text{'Gonzo'} \end{bmatrix} \\ \textsc{xcomp} & h\begin{bmatrix} \textsc{pred} & \text{'go'} \\ \textsc{subj} & \end{bmatrix} \end{bmatrix}$$

So the attribute paths (f SUBJ) and (f XCOMP SUBJ) share as their value the f-structure g. This f-structure contributes one semantic resource that we must use in the Glue proof of the sentence's meaning. The embedded verb is an ordinary intransitive verb that needs to consume its subject's resource and the matrix equi verb also takes its subject as a semantic argument. Thus it seems that both verbs need to consume the one subject resource. But, if we use this subject resource as a premise in deriving the meaning of the matrix verb, the resource is consumed and is not available for the embedded verb. Likewise, if the embedded verb consumes the subject resource, it is unavailable for the matrix verb.

[4]These are also known as functional control equations.

[5]For motivation of such an analysis, see Bresnan (1982a) and Falk (to appear).

One possible solution to this problem is to treat *paths* in f-structures as contributing resources (Kehler et al., 1999), rather than the standard Glue treatment, whereby *nodes* (i.e. the f-structures values of paths) contribute resources. Consider this proposal with respect to (13). There are two subject paths, (f SUBJ g) and (f XCOMP SUBJ g). Each of these contributes a resource and as a result both the matrix and embedded verbs have a subject resource to consume.

The are two major problems with this proposal. First, it involves a non-trivial extension of Glue theory. It needs an additional axiom and the repercussions of the extension are not clear. Essentially, this modification results in a weakening of the notion of resource-sensitivity, because there can always be multiple paths leading to the same value. Second, by providing both the matrix equi verb and the embedded verb with a subject resource, this solution would force the clausal equi complement to denote a proposition. The embedded verb denotes a property and this would combine withe denotation of the subject, an individual or generalized quantifier, to yield a proposition. There would be no principled way to get a property denotation for the clausal complement instead. However, there is a long-standing literature in theoretical semantics that argues precisely for a property denotation of clausal equi complements (Chierchia, 1984, 1985; Chierchia and Jacobson, 1986). I now turn to a brief summary of these arguments.

4 The Denotation of Clausal Equi Complements

Chierchia and various other semanticists have long argued that the denotation of the clausal complement of an equi verb is a property (14a) and not a proposition (14b) (Chierchia, 1984, 1985; Chierchia and Jacobson, 1986).

(14) a. try(gonzo, λx.go(x))

b. try(gonzo, go(gonzo))

The fundamental motivation for this comes from certain entailments/inference patterns that are very robust.[6]

(15) **Property inference patterns:**

Quantification	**Ellipsis**
Gonzo tried to go.	Gonzo tried to go.
Andrew tried everything that Gonzo tried.	Andrew did too.
Andrew tried to go.	Andrew tried to go.
try(gonzo, λx.go(x))	try(gonzo, λx.go(x))
$\forall$P.[try(gonzo, P) $\rightarrow$ try(andrew, P)]	$\exists$P$\exists$Q.[Q(gonzo, P) $\wedge$ Q(andrew, P)]
try(andrew, λx.go(x))	try(andrew, λx.go(x))

[6]Chierchia (1984) discusses the quantificational inference patterns on the left. The ellipsis patterns were suggested to me by Mary Dalrymple (p.c.). The treatment of ellipsis given here is based very roughly on Dalrymple et al. (1991). None of the logical forms given in this section are intended as real analyses, but are rather presented as suggestive sketches which highlight the problems being discussed.

(16) **Propositional inference patterns:**

Quantification	**Ellipsis**
Gonzo tried to go.	Gonzo tried to go.
Andrew tried everything that Gonzo tried.	Andrew did too.
??	**??**
[* Andrew tried for Gonzo to go.]	[* Andrew tried for Gonzo to go.]
try(gonzo, go(gonzo))	try(gonzo, λx.go(x))
$\forall$P.[try(gonzo, P) $\rightarrow$ try(andrew, P)]	$\exists$P$\exists$Q.[Q(gonzo, P) $\wedge$ Q(andrew, P)]
try(andrew, go(gonzo))	try(andrew, go(gonzo))

The property theory gets the correct entailments, while the propositional theory does not. In fact there are no English sentences corresponding to the conclusions of the propositional inference patterns. For this reason, Chierchia treats the complement of equi as denoting a property. Semantics is ultimately grounded in providing meanings for natural language that capture our intuitions about entailments. Thus, we must reject a semantics that results in denotations with missing or false entailments, as is the case with the propositional denotation of clausal equi complements.

There have been objections to the property theory, which I turn to in the next section, and it may be that the empirical problems with the propositional theory that have been outlined here can be overcome. But in the absence of proposals to this effect, the Kehler et al. (1999) treatment of structure-sharing and its relationship to Glue semantics is unsatisfactory, because it can *only* give a propositional denotation to the equi complements, as discussed above.

4.1 Two Problems for the Property Theory

4.1.1 Locality of Anaphoric Binding

Pollard and Sag (Pollard and Sag, 1994; Sag and Pollard, 1991) have noted that the property theory has trouble with reflexives in clausal complements of equi verbs. Consider:

(17) Gonzo tried to pinch himself/*herself.

This case is problematic for the property theory of equi, because it has been developed in frameworks which are strictly compositional (Montague Grammar, Categorial Grammar). Accordingly, there is no local antecedent for the reflexive in (14). In the syntax, the clausal equi complement is a VP, which has no subject. There *must not* be a syntactic subject, because the theory is strictly compositional and needs a property denotation for the complement in the semantics. If a subject were present in the semantics, it would combine with the property to give a proposition. Thus, there is also no subject in the semantics, where the subject is represented as a variable bound by a lambda.

The result is that in giving a property denotation to the clausal equi complement, there is no local antecedent for the reflexive, and an extremely robust generalization from binding theory is lost. The parallel projection architecture of LFG offers a solution to this apparent dilemma, which I give in section 6.1.

4.1.2 Sentential Complements with Property Denotations

Zec (1987) notes that Serbo-Croatian[7] sentences like the following pose a problem for the property theory of equi semantics (Zec, 1987: 142).

(18) Petar je pokušao da dodje
Petar Aux tried Comp come(Pres)
Peter tried to come.

The complement to the matrix verb *pokušao* is clearly not a subjectless VP for two reasons. First, there is an overt complementizer, indicating that this is a CP, not a VP. Second, Zec argues that there is actually a null pronominal subject in the embedded clause, so there is an f-structure SUBJ. *Pokušao* is nevertheless an equi verb and there is a relationship of obligatory control between its subject and the subject of its complement. Furthermore, Serbo-Croatian control verbs participate in inferences like (15), indicating that the clausal equi complement denotes a property despite initial appearances. This is a problem for Chierchia, because his theory predicts that the sentential complement in (18) should denote a proposition, as it has a subject. Again, the present analysis uses LFG's parallel projection architecture to overcome this problem, as show in section 6.2.

5 A Glue Analysis of Equi

Let us consider the following sentence;

(19) Gonzo tried to go.

I assume the following (partial) lexical entries for this sentence:[8]

(20) Gonzo N $(\uparrow \text{PRED}) = \text{`Gonzo'}$
gonzo : $\uparrow_\sigma$
$= g_\sigma$
tried V $(\uparrow \text{PRED}) = \text{`try'}$
$\lambda x \lambda P.\text{try}(x, P) : \; (\uparrow \text{SUBJ})_\sigma \multimap (((\uparrow \text{XCOMP SUBJ})_\sigma \multimap (\uparrow \text{XCOMP})_\sigma) \multimap \uparrow_\sigma)$
$= g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)$
go V $(\uparrow \text{PRED}) = \text{`go'}$
$\lambda y.\text{go}(y) : \; (\uparrow \text{SUBJ}) \multimap \uparrow_\sigma$
$= g_\sigma \multimap h_\sigma$

These entries are clearly simplified, and many syntactic details have been suppressed. For simplicity's sake, I've also presented a simple extensional predicate calculus as the meaning language, which is to the left of the colons in the Glue formulas.[9]

The Glue formulas have been presented first in their general form, as they are listed in the lexicon, and secondly with the $\uparrow$ metavariables instantiated to nodes in the following f-structure for (19).

[7] This is the term that Zec uses.

[8] I have assumed a co-head analysis of infinitival *to*. If a raising analysis is preferred, then the derivation would essentially be like the one presented for raising embedded under equi in section 6.4.2.

[9] For a presentation of an intensional version of the meaning language see Asudeh (to appear).

(21)
$$f\begin{bmatrix} \textsc{pred} & \text{`try'} \\ \textsc{subj} & g\begin{bmatrix} \textsc{pred} & \text{`Gonzo'} \end{bmatrix} \\ \textsc{xcomp} & h\begin{bmatrix} \textsc{pred} & \text{`go'} \\ \textsc{subj} & \end{bmatrix} \end{bmatrix}$$

Using the instantiated glue premises, we can construct the following proof.

(22) **Glue proof, with meanings**

$$\dfrac{\dfrac{\text{gonzo} : g_\sigma \qquad \lambda x \lambda P.\text{try}(x, P) : g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)}{\lambda P.\text{try}(\text{gonzo}, P) : (g_\sigma \multimap h_\sigma) \multimap f_\sigma}\multimap_{\mathcal{E}} \qquad \lambda y.\text{go}(y) : g_\sigma \multimap h_\sigma}{\text{try}(\text{gonzo}, \lambda y.\text{go}(y)) : f_\sigma}\multimap_{\mathcal{E}}$$

In the first step of the proof, we take the Glue resource provided by the subject, *Gonzo*, and combine it with the Glue resource provided by the equi matrix verb, *tried*, using implication elimination (modus ponens; see (6)). This corresponds to function application in the meaning language, and the first argument of try is the denotation of the subject, $gonzo$. The result of the first step is then combined with the resource provided by the embedded verb, again using implication elimination. This results in a semantics for the f-structure corresponding to the sentence which is the try relation between $gonzo$ and the property of going.

The very same Glue entry for the equi verb can yield the propositional denotation for the clausal complement. The only difference is in the meaning language, where the controllee is given as an argument to the property:

(23) tried V $(\uparrow \textsc{pred}) = \text{`try'}$
λxλP.try(x, P(x)) : $(\uparrow \textsc{subj})_\sigma \multimap (((\uparrow \textsc{xcomp subj})_\sigma \multimap (\uparrow \textsc{xcomp})_\sigma) \multimap \uparrow_\sigma)$
$= g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)$

Combining this with the same meaning for *go*, the last line of the Glue proof would instead be $try(gonzo, \lambda y.go(y)(gonzo)) : f_\sigma$, which reduces to $try(gonzo, go(gonzo)) : f_\sigma$ after function application.

6 Empirical Results

6.1 Anaphoric Binding in Equi Complements

Let us consider again (17), which I noted as a potential problem for the property theory:

(17) Gonzo tried to pinch himself.

At c-structure, the clausal complement (IP) does not have a subject, but at f-structure, there is a subject, which is structure-shared with its controller, the matrix subject. The projection architecture allows this kind of mismatch and the equi verb lexically specifies the functional control equation which identifies its SUBJ and its XCOMP SUBJ.

Anaphoric binding in LFG is defined at the level of f-structure (Dalrymple, 1993; Bresnan, 2000), as indicated by the binding equation in the lexical entry for the reflexive in (24a). For sentence (17), we get the f-structure in (24b).

(24) a. himself N

$$
\begin{array}{l}
(\uparrow \textsc{pers}) = 3 \\
(\uparrow \textsc{num}) = \textsc{sg} \\
(\uparrow \textsc{gend}) = \textsc{masc} \\
\mathbf{(((}\textsc{gf}\ \alpha\mathbf{)} \uparrow \mathbf{)}\ \textsc{gf}'\ \textsc{index}\mathbf{)} = \mathbf{(}\uparrow \textsc{index}\mathbf{)}
\end{array}
$$

b.

$$
f\left[\begin{array}{lll}
\textsc{pred} & & \text{‘try’} \\
\textsc{subj} & g & \left[\begin{array}{ll} \textsc{pred} & \text{‘Gonzo’} \\ \textsc{index} & j \end{array}\right] \\
\textsc{xcomp} & h & \left[\begin{array}{lll}
\textsc{pred} & & \text{‘pinch’} \\
\textsc{subj} & & \\
\textsc{obj} & i & \left[\begin{array}{ll} \textsc{pred} & \text{‘pro’} \\ \textsc{nuclear} & + \\ \textsc{index} & j \end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Instantiating the binding equation in (24a):

(25)

$$
\begin{array}{l}
(((\textsc{gf}\ \alpha) \uparrow)\ \textsc{gf}'\ \textsc{index}) = (\uparrow \textsc{index}) \\
(((\textsc{obj}\ e)\ i)\ \textsc{subj}\ \textsc{index}) = (i\ \textsc{index}) \\
(h\ \textsc{subj}\ \textsc{index}) = j \\
(g\ \textsc{index}) = j \\
j = j
\end{array}
$$

Thus, there is a local antecedent for the reflexive at f-structure, the appropriate level for binding theory.

The relationship between levels of representation in LFG is not isomorphic, but it is systematic, as defined by projection functions. Although the subject is present in f-structure, the *denotation* of the clausal equi complement can be a property, as we've seen. This analysis exploits the parallel projection architecture and the modular nature of information contribution in LFG to capture the correct syntax for constituency and binding (c-structure and f-structure) as well as the right semantics for equi.

6.2 Obligatory Anaphoric Control

Recall the Serbo-Croatian sentence (18):

(18) Petar je pokušao da dodje
Petar Aux tried Comp come(Pres)
Peter tried to come.

Zec (1987) noted this as a problem for the property theory because the clausal equi complement is clearly a CP with a null subject, yet it seems to denote a property. This is only a problem if we assume strict compositionality, and the solution here relies on using normal Glue semantics and exploiting the capacity for systematic mismatches between levels of representation in LFG's grammatical architecture.

I assume the following lexical entries for (18).

(26)
$$
\begin{array}{lll}
\text{Petar} & \text{N} & (\uparrow \textsc{pred}) = \text{`Petar'} \\
 & & \text{petar}: \ \uparrow_\sigma \\
 & & \qquad\quad = g_\sigma \\
\text{je} & \text{I} & (\uparrow \textsc{tense}) = \textsc{past} \\
\text{pokušao} & \text{V} & (\uparrow \textsc{pred}) = \text{`try'} \\
 & & (\uparrow \textsc{comp subj index}) = (\uparrow \textsc{subj index}) \\
 & & \lambda w \lambda P.\text{try}(w, P): \ (\uparrow \textsc{subj})_\sigma \multimap (((\uparrow \textsc{comp subj})_\sigma \multimap (\uparrow \textsc{comp})_\sigma) \multimap \uparrow_\sigma) \\
 & & \qquad\qquad\qquad\quad = g_\sigma \multimap ((i_\sigma \multimap h_\sigma) \multimap f_\sigma) \\
 & & \lambda x \lambda y.x: \ (\uparrow \textsc{subj})_\sigma \multimap ((\uparrow \textsc{comp subj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma) \\
 & & \qquad\qquad = g_\sigma \multimap (i_\sigma \multimap g_\sigma) \\
\text{da} & \text{C} & (\uparrow \textsc{mood}) = \textsc{decl} \\
\text{dodje} & \text{V} & (\uparrow \textsc{pred}) = \text{`come'} \\
 & & (\uparrow \textsc{tense}) = \textsc{pres} \\
 & & \left(\begin{array}{l} (\uparrow \textsc{subj pred}) = \text{`pro'} \\ X: \ (\uparrow \textsc{subj})_\sigma \\ \qquad = i_\sigma \end{array} \right) \\
 & & \lambda z.\text{come}(z): \ (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma \\
 & & \qquad\qquad\qquad = i_\sigma \multimap h_\sigma
\end{array}
$$

The second line in each Glue formula has instantiated node descriptions with the node names from the following f-structure for (18):

(27)
$$
f\left[\begin{array}{ll}
\textsc{pred} & \text{`try'} \\
\textsc{subj} & g\left[\begin{array}{ll} \textsc{pred} & \text{`Petar'} \\ \textsc{index} & j \end{array}\right] \\
\textsc{comp} & h\left[\begin{array}{ll} \textsc{pred} & \text{`come'} \\ \textsc{subj} & i\left[\begin{array}{ll} \textsc{pred} & \text{`pro'} \\ \textsc{index} & j \end{array}\right] \\ \textsc{mood} & \textsc{decl} \end{array}\right] \\
\textsc{tense} & \textsc{past}
\end{array}\right]
$$

I have made the simplifying assumption that the auxiliary verb simply contributes TENSE to the f-structure and that the complementizer contributes MOOD. The entry for the embedded verb *dodje* is general, and not just for its occurrence in complements of equi verbs. This verb optionally specifies its SUBJ PRED as 'pro', since Serbo-Croatian is a pro-drop language. When the null subject is contributed by the verb, the semantics of the null pronominal is also contributed, as indicated by the Glue formula $X : (\uparrow \textsc{subj})_\sigma$.[10]

[10] Clearly this is a not a satisfactory treatment of anaphora, but discussion of the treatment of anaphora in a dynamic Glue, which is being developed at PARC, would take us too far afield.

The entry for the subject equi verb *pokušao* is different from the entry for the English verb *try* in (20) above. Its clausal complement is a COMP, not an XCOMP, because the complement has its own subject rather than the subject being structure-shared with another GF by functional control. Instead, the control relationship is anaphoric as indicated by the equation requiring that *pokušao*'s COMP SUBJ and SUBJ be coindexed. Corresponding to the anaphoric control specified by the coindexation equation, the second Glue formula in the entry consumes the pronoun's meaning. This is motivated because the pronoun can only be bound by the controlling subject. In effect, it is not contributing a normal pronominal meaning, but is rather a device that is employed in certain languages (like Serbo-Croatian) to establish an equi relation. Thus, we can think of this Glue formula as going hand in hand with the anaphoric control equation, just as the embedded verb's contribution of its null subject goes hand in hand with a Glue formula for that subject.

This leaves the final and most important detail of the present analysis of anaphoric control: the first Glue formula, which gives the semantics of the equi verb, is the same as the Glue for the equi verb *try* in English, a language that uses functional control in the syntax. In both cases, the matrix subject is consumed to yield an implication that consumes the *entire* clausal complement to yield the semantics of the outer f-structure and thus the sentence. The following proof demonstrates the parallelism. Notice that once the Glue premises dealing with the anaphoric control relation and embedded subject have been consumed, the remainder of the proof (starting at the third line) is identical to the proof in (22) for the English sentence *Gonzo tried to go*.

(28)

$$\dfrac{\lambda z.\text{come}(z) : i_\sigma \multimap h_\sigma \qquad \dfrac{\dfrac{\dfrac{\text{petar} : g_\sigma \quad \lambda x\lambda y.x : g_\sigma \multimap (i_\sigma \multimap g_\sigma)}{\lambda y.\text{petar} : i_\sigma \multimap g_\sigma} \quad \text{X} : i_\sigma}{\text{petar} : g_\sigma} \quad \lambda w\lambda P.\text{try}(w,P) : g_\sigma \multimap ((i_\sigma \multimap h_\sigma) \multimap f_\sigma)}{\lambda P.\text{try}(\text{petar},P) : (i_\sigma \multimap h_\sigma) \multimap f_\sigma}}{\text{try}(\text{petar},\ \lambda z.\text{come}(z)) : f_\sigma}$$

Again we see that LFG's architecture has been exploited to solve a previous problem in the semantics of equi. Although there is a null pronominal subject in the clausal equi complement, I am essentially proposing here that it is only there to maintain the obligatory syntactic relationship between the controller and the controllee. Thus, there is typological variation in the syntactic mechanisms used to maintain control relationships. English uses functional control, while other languages, such as Serbo-Croation or Icelandic (Andrews, 1982) use anaphoric control. The differences in the syntactic mechanisms are reflected in the Glue semantics, but it is still possible for the controlled clausal complements to have the same denotation, no matter which syntactic mechanism the grammar of a language employs. Therefore, there is a separation of syntax and semantics, but there is a systematic relationship between the two, and there can be typological differences in one, without there *necessarily* being differences in the other.

6.3 Equi Embedded under Equi

At this point the reader may well be wondering if this analysis can handle an equi verb embedded under another equi verb, which will result in three-way structure-sharing. In fact, this is not a problem as the following example and its Glue proof show.

(29) a. Gonzo promised to try to go.

b. $$f\left[\begin{array}{ll} \text{PRED} & \text{‘promise’} \\ \text{SUBJ} & g\left[\begin{array}{ll} \text{PRED} & \text{‘Gonzo’} \end{array}\right] \\ \text{XCOMP} & h\left[\begin{array}{ll} \text{PRED} & \text{‘try’} \\ \text{SUBJ} & \\ \text{XCOMP} & i\left[\begin{array}{ll} \text{PRED} & \text{‘go’} \\ \text{SUBJ} & \end{array}\right] \end{array}\right] \end{array}\right]$$

c.
$$\begin{array}{rcl}
\text{gonzo :} & & \uparrow_\sigma \\
& = & g_\sigma \\
\lambda x \lambda P.\text{promise}(x, P) : & & (\uparrow \text{SUBJ})_\sigma \multimap (((\uparrow \text{XCOMP SUBJ})_\sigma \multimap (\uparrow \text{XCOMP})_\sigma) \multimap \uparrow_\sigma) \\
& = & g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma) \\
\lambda y \lambda Q.\text{try}(y, Q) : & & (\uparrow \text{SUBJ})_\sigma \multimap (((\uparrow \text{XCOMP SUBJ})_\sigma \multimap (\uparrow \text{XCOMP})_\sigma) \multimap \uparrow_\sigma) \\
& = & g_\sigma \multimap ((g_\sigma \multimap i_\sigma) \multimap h_\sigma) \\
\lambda w.\text{go} : & & (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma \\
& = & g_\sigma \multimap i_\sigma
\end{array}$$

d. **Glue proof, without meanings**[11]

$$\dfrac{\dfrac{g_\sigma \quad g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)}{(g_\sigma \multimap h_\sigma) \multimap f_\sigma}\multimap_{\mathcal{E}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{[g_\sigma]^1 \quad g_\sigma \multimap ((g_\sigma \multimap i_\sigma) \multimap h_\sigma)}{(g_\sigma \multimap i_\sigma) \multimap h_\sigma}\multimap_{\mathcal{E}} \quad g_\sigma \multimap i_\sigma}{h_\sigma}\multimap_{\mathcal{E}}}{g_\sigma \multimap h_\sigma}\multimap_{\mathcal{I},1}}{}}{\text{promise}(\text{gonzo}, \lambda z.\text{try}(z,\ \lambda w.\text{go}(w))) : f_\sigma}\multimap_{\mathcal{E}}$$

The lefthand side of the proof shows that the Glue resource for the matrix subject is consumed to prove the consequent of the matrix equi verb, as already demonstrated in section 5. The righthand side of the proof shows the first use of linear implication introduction (see (6) in section 2.4). We assume a Glue resource, g_σ, and flag the assumption. We then use this resource to get the meaning for the outer XCOMP, h. Since we have used the assumed g_σ in proving h_σ, we can discharge the assumption and get the implication $g_\sigma \multimap h_\sigma$. We then combine this resource with the resource from the lefthand side of the proof to get the meaning for the sentence, which is a relation of promising between an individual, $gonzo$, and the property of trying to go. Thus, once again in our semantics we have the property denotation for the clausal equi complement and the Glue proof goes through despite the three-way structure-sharing.

6.4 The Interaction of Equi and Raising

In this section I will demonstrate that the analysis of equi presented thus far interacts nicely with a Glue analysis of raising, which I present first.

[11]Glue proofs with meanings for this example and for examples (31) and (32) are given in the appendix.

6.4.1 Raising

At the syntactic level of f-structure, raising looks very much like equi, because the raising verb's SUBJ (or OBJ) functionally controls its XCOMP SUBJ, as illustrated in (30b) below. The crucial difference between equi and raising is a semantic one. In equi, the controlling GF is simultaneously a semantic argument of the equi verb and of its XCOMP, but in raising the "raised" GF is not a semantic argument of the raising verb, only of its XCOMP. Evidence for this comes from expletives. Raising verbs can take expletive subjects or objects in lieu of a raised one, but equi verbs can never take an expletive instead of the equi controller.

Furthermore, the denotation of a raising verb's clausal complement is a proposition, not a property (Montague, 1974). The verb *seem*, for example, is a one-place predicate, taking a propositional argument, as its subject is not a semantic argument. This is why the Glue entry for the verb *seem* in (30c) is $\lambda\mathcal{P}.seem(\mathcal{P}) : (\uparrow \text{XCOMP})_\sigma \multimap \uparrow_\sigma$. *Seem* only needs its XCOMP's meaning, which is a proposition,[12] to provide its own meaning.

(30) a. Gonzo seemed to vanish.

b.

$$f\left[\begin{array}{ll} \text{PRED} & \text{`seem'} \\ \text{SUBJ} & \\ \text{XCOMP} & h\left[\begin{array}{ll} \text{PRED} & \text{`vanish'} \\ \text{SUBJ} & g\left[\begin{array}{ll}\text{PRED} & \text{`Gonzo'}\end{array}\right] \end{array}\right] \end{array}\right]$$

c.

$$\begin{array}{rcl} \text{gonzo} : & & \uparrow_\sigma \\ & = & g_\sigma \\ \lambda\mathcal{P}.\text{seem}(\mathcal{P}) : & & (\uparrow \text{XCOMP})_\sigma \multimap \uparrow_\sigma \\ & = & h_\sigma \multimap f_\sigma \\ \lambda\text{x.vanish(x)} : & & (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma \\ & = & g_\sigma \multimap h_\sigma \end{array}$$

d. **Glue proof, without meanings**

$$\dfrac{\dfrac{g_\sigma \qquad g_\sigma \multimap h_\sigma}{h_\sigma}\multimap_{\mathcal{E}} \qquad h_\sigma \multimap f_\sigma}{\text{seem}(\text{vanish}(\text{gonzo})) : f_\sigma}\multimap_{\mathcal{E}}$$

e. **Glue proof, with meanings**

$$\dfrac{\dfrac{\text{gonzo} : g_\sigma \qquad \lambda x.\text{vanish}(x) : g_\sigma \multimap h_\sigma}{\text{vanish}(\text{gonzo}) : h_\sigma}\multimap_{\mathcal{E}} \qquad \lambda\mathcal{P}.\text{seem}(\mathcal{P}) : h_\sigma \multimap f_\sigma}{\text{seem}(\text{vanish}(\text{gonzo})) : f_\sigma}\multimap_{\mathcal{E}}$$

We end up with the sentence *Gonzo seemed to vanish* meaning $seem(vanish(gonzo))$. This is a one place predicate with a propositional argument, just as desired.

[12] The variable $\mathcal{P}$ ranges over propositions, unlike the previously encountered P, which ranges over properties.

6.4.2 Raising under Equi

With the analysis of raising in hand, we can derive a meaning for a sentence that has a matrix equi verb and an embedded raising complement. Again, this results in three-way structure-sharing, but the analysis developed here is fully general and the structure-sharing once again poses no problem.

(31) a. Gonzo tried to seem to vanish.

b.
$$f\begin{bmatrix} \textsc{pred} & \text{`try'} \\ \textsc{subj} & g\begin{bmatrix}\textsc{pred} & \text{`Gonzo'}\end{bmatrix} \\ \textsc{xcomp} & h\begin{bmatrix} \textsc{pred} & \text{`seem'} \\ \textsc{subj} & \\ \textsc{xcomp} & i\begin{bmatrix}\textsc{pred} & \text{`vanish'} \\ \textsc{subj} & \end{bmatrix}\end{bmatrix}\end{bmatrix}$$

c.
$$\begin{array}{rcl} \text{gonzo :} & & \uparrow_\sigma \\ & = & g_\sigma \\ \lambda\text{y}\lambda\text{Q.try(y, Q) :} & & (\uparrow \textsc{subj})_\sigma \multimap (((\uparrow \textsc{xcomp subj})_\sigma \multimap (\uparrow \textsc{xcomp})_\sigma) \multimap \uparrow_\sigma) \\ & = & g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma) \\ \lambda\mathcal{P}.\text{seem}(\mathcal{P}) : & & (\uparrow \textsc{xcomp})_\sigma \multimap \uparrow_\sigma \\ & = & i_\sigma \multimap h_\sigma \\ \lambda\text{w.vanish(w) :} & & (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma \\ & = & g_\sigma \multimap i_\sigma \end{array}$$

d. **Glue proof, without meanings**

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[g_\sigma]^1 \quad g_\sigma \multimap i_\sigma}{i_\sigma}\multimap_{\mathcal{E}} \quad i_\sigma \multimap h_\sigma}{h_\sigma}\multimap_{\mathcal{E}}}{g_\sigma \multimap h_\sigma}\multimap_{\mathcal{I},1} \qquad \dfrac{g_\sigma \quad g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)}{(g_\sigma \multimap h_\sigma) \multimap f_\sigma}\multimap_{\mathcal{E}}}{\text{try}(\text{gonzo},\ \lambda z.\text{seem}(\text{vanish}(z))) : f_\sigma}\multimap_{\mathcal{E}}$$

Using simply the equi and raising entries we have already encountered so far, the meaning for the sentence is given as $try(gonzo, \lambda z.seem(vanish(z)))$, which is a relation between the individual $gonzo$ and the property of seeming to vanish. Once again, the denotation of the clausal equi complement is a property, although the complement is headed by a raising verb.

6.4.3 Equi under Raising

The semantics for an equi verb embedded as the complement of a raising verb is equally unproblematic:

(32) a. Gonzo seemed to try to go.

b.

$$
f\begin{bmatrix}
\textsc{pred} & \text{‘seem’} \\
\textsc{subj} & g\begin{bmatrix} \textsc{pred} & \text{‘Gonzo’} \end{bmatrix} \\
\textsc{xcomp} & h\begin{bmatrix}
\textsc{pred} & \text{‘try’} \\
\textsc{subj} & \\
\textsc{xcomp} & i\begin{bmatrix} \textsc{pred} & \text{‘go’} \\ \textsc{subj} & \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

c.

$$
\begin{array}{rcl}
\text{gonzo}: & & \uparrow_\sigma \\
& = & g_\sigma \\
\lambda\mathcal{P}.\text{seem}(\mathcal{P}): & & (\uparrow \textsc{xcomp})_\sigma \multimap \uparrow_\sigma \\
& = & h_\sigma \multimap f_\sigma \\
\lambda y \lambda Q.\text{try}(y, Q): & & (\uparrow \textsc{subj})_\sigma \multimap (((\uparrow \textsc{xcomp subj})_\sigma \multimap (\uparrow \textsc{xcomp})_\sigma) \multimap \uparrow_\sigma) \\
& = & g_\sigma \multimap ((g_\sigma \multimap i_\sigma) \multimap h_\sigma) \\
\lambda w.\text{go}(w): & & (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma \\
& = & g_\sigma \multimap i_\sigma
\end{array}
$$

d. **Glue proof, without meanings**

$$
\dfrac{\dfrac{\dfrac{g_\sigma \quad g_\sigma \multimap ((g_\sigma \multimap i_\sigma) \multimap h_\sigma)}{(g_\sigma \multimap i_\sigma) \multimap h_\sigma}\multimap_{\mathcal{E}} \qquad g_\sigma \multimap i_\sigma}{h_\sigma}\multimap_{\mathcal{E}} \qquad h_\sigma \multimap f_\sigma}{\text{seem}(\text{try}(\text{gonzo},\ \lambda w.\text{go}(w))) : f_\sigma}\multimap_{\mathcal{E}}
$$

The meaning we end up with is a one place predicate, $seem$, which takes a propostional argument, $try(gonzo, \lambda w.go(w))$. And the meaning of the embedded equi verb is the usual relation between an individual and property.

6.5 De Re and De Dicto Scope

Lastly, it can be shown that the semantics for equi and raising developed here naturally yield the de re/de dicto differences between equi and raising verbs (Dowty et al., 1981; Montague, 1974). In particular, a quantified subject of a raising verb can take either wide scope (de re reading) or narrow scope (de dicto reading) with respect to the verb. However, the subject of an equi verb cannot take narrow scope with respect to the equi verb, and the de dicto reading is unavailable.

6.5.1 Raising: Both De Re and De Dicto Readings Available

Consider the example in (33a) below. The de re reading entails the existence of a goblin, whereas the de dicto reading does not. On the de dicto reading, something that *seems* to be a goblin tried to pinch Gonzo, but the thing in question could (for example) be a child in a Halloween costume. Let us assume an existentially quantified denotation of indefinite noun phrases, represented as a generalized quantifier (Barwise and Cooper, 1981). We can then write the Glue for *a goblin* as in (33c), following the Glue treatment of generalized quantifiers in Dalrymple et al. (1999b). There are two ways to instantiate the variable X in the indefinite noun phrase's Glue. To get the de re reading in (33d), we first prove f_σ (i.e. the sentence's semantics) and then instantiate X to f_σ, giving the quantified noun phrase wide scope. For the de dicto reading, we combine the quantifier with the XCOMP's Glue, and then combine the result with the matrix raising verb. This gives the quantifier narrow scope.

(33) a. A goblin seemed to pinch Gonzo.

b.
$$f\begin{bmatrix} \text{PRED} & \text{`seem'} \\ \text{SUBJ} & g\begin{bmatrix}\text{PRED} & \text{`goblin'}\end{bmatrix} \\ \text{XCOMP} & h\begin{bmatrix} \text{PRED} & \text{`pinch'} \\ \text{SUBJ} & \\ \text{OBJ} & i\begin{bmatrix}\text{PRED} & \text{`Gonzo'}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

c.
$$\begin{aligned}
\lambda Q \exists z.[\text{goblin}(z) \wedge Q(z)] : \quad & (\uparrow_\sigma \multimap X) \multimap X \\
= \quad & (g_\sigma \multimap X) \multimap X \\
\lambda \mathcal{P}.\text{seem}(\mathcal{P}) : \quad & (\uparrow \text{XCOMP})_\sigma \multimap \uparrow_\sigma \\
= \quad & h_\sigma \multimap f_\sigma \\
\lambda y \lambda x.\text{pinch}(x, y) : \quad & (\uparrow \text{OBJ})_\sigma \multimap ((\uparrow \text{SUBJ}) \multimap \uparrow_\sigma) \\
= \quad & i \multimap (g \multimap h_\sigma) \\
\text{gonzo} : \quad & \uparrow_\sigma \\
= \quad & i_\sigma
\end{aligned}$$

d. **De re**[13]

$$\dfrac{(g_\sigma \multimap X) \multimap X \qquad \dfrac{\dfrac{\dfrac{\dfrac{i_\sigma \quad i_\sigma \multimap (g_\sigma \multimap h_\sigma)}{g_\sigma \multimap h_\sigma}\multimap_{\mathcal{E}} \qquad [g_\sigma]^1}{h_\sigma}\multimap_{\mathcal{E}} \qquad h_\sigma \multimap f_\sigma}{f_\sigma}\multimap_{\mathcal{E}}}{g_\sigma \multimap f_\sigma}\multimap_{\mathcal{I},1}}{\exists z.[\text{goblin}(z) \wedge \text{seem}(\text{pinch}(z,\ \text{gonzo}))] : f_\sigma}\multimap_{\mathcal{E}}, X = f_\sigma$$

e. **De dicto**

$$\dfrac{h_\sigma \multimap f_\sigma \qquad \dfrac{\dfrac{i_\sigma \quad i_\sigma \multimap (g_\sigma \multimap h_\sigma)}{g_\sigma \multimap h_\sigma}\multimap_{\mathcal{E}} \qquad (g_\sigma \multimap X) \multimap X}{h_\sigma}\multimap_{\mathcal{E}}, X = h_\sigma}{\text{seem}(\exists z.[\text{goblin}(z) \wedge \text{pinch}(z,\ \text{gonzo})]) : f_\sigma}\multimap_{\mathcal{E}}$$

[13]See the appendix for Glue proofs with meanings of (33d), (33e), and (34d).

6.5.2 Equi: Only De Re Reading Available

Now let us consider the same sentence with the equi verb *try* replacing the raising verb *seem*. The resulting sentence only has a de re reading for the quantified subject: the goblin is entailed to exist. It is easy to understand why this should be so. If a goblin tried to pinch Gonzo, then there must be some goblin or other that did this. The de re reading is derived in the same manner as for (33). The sentence's semantics is first derived and then combined with the quantifier, which gets wide scope. However, there is no way to give the quantifier narrow scope. If we attempt to do this, the subject resource (g_σ) is lost and the Glue proof fails.

(34) a. A goblin tried to pinch Gonzo.

b.
$$
f\left[\begin{array}{ll}
\textsc{pred} & \text{'try'} \\
\textsc{subj} & g\left[\begin{array}{ll}\textsc{pred} & \text{'goblin'}\end{array}\right] \\
\textsc{xcomp} & h\left[\begin{array}{ll}
\textsc{pred} & \text{'pinch'} \\
\textsc{subj} & \\
\textsc{obj} & i\left[\begin{array}{ll}\textsc{pred} & \text{'Gonzo'}\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

c.
$$
\begin{array}{rcl}
\lambda Q\exists z.[\text{goblin}(z) \wedge Q(z)] : & & (\uparrow_\sigma \multimap X) \multimap X \\
& = & (g_\sigma \multimap X) \multimap X \\
\lambda w\lambda P.\text{try}(w, P) : & & (\uparrow \textsc{subj})_\sigma \multimap (((\uparrow \textsc{xcomp subj})_\sigma \multimap (\uparrow \textsc{xcomp})_\sigma) \multimap \uparrow_\sigma) \\
& = & g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma) \\
\lambda y\lambda x.\text{pinch}(x, y) : & & (\uparrow \textsc{obj})_\sigma \multimap ((\uparrow \textsc{subj}) \multimap \uparrow_\sigma) \\
& = & i \multimap (g \multimap h_\sigma) \\
\text{gonzo} : & & \uparrow_\sigma \\
& = & i_\sigma
\end{array}
$$

d. **De re**

$$
\frac{(g_\sigma \multimap X) \multimap X \qquad \dfrac{\dfrac{\dfrac{[g_\sigma]^1 \quad g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)}{(g_\sigma \multimap h_\sigma) \multimap f_\sigma}\multimap_{\mathcal{E}} \qquad \dfrac{i_\sigma \qquad i_\sigma \multimap (g_\sigma \multimap h_\sigma)}{g_\sigma \multimap h_\sigma}\multimap_{\mathcal{E}}}{f_\sigma}\multimap_{\mathcal{E}}}{g_\sigma \multimap f_\sigma}\multimap_{\mathcal{I},1}}{\exists z.[\text{goblin}(z) \wedge \text{try}(z,\ \lambda x.\text{pinch}(x,\ \text{gonzo}))] : f_\sigma}\multimap_{\mathcal{E}}, X = f_\sigma
$$

e. **De dicto - no proof**

$$
\frac{\dfrac{g_\sigma \quad g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)}{(g_\sigma \multimap h_\sigma) \multimap f_\sigma}\multimap_{\mathcal{E}} \qquad \dfrac{\dfrac{i_\sigma \multimap (g_\sigma \multimap h_\sigma) \quad i_\sigma}{g_\sigma \multimap h_\sigma}\multimap_{\mathcal{E}} \qquad (g_\sigma \multimap X) \multimap X}{h_\sigma}\multimap_{\mathcal{E}}, X = h_\sigma}{}
$$

7 Conclusion

The analysis of equi developed here has shown that the resource-sensitivity of Glue semantics is not necessarily at odds with functional control and structure-sharing at f-structure. In fact, the tension between these features of the architecture leads to a restrictive semantics for equi which has several advantages. First, the semantics naturally yields a property denotation for the clausal equi complement, which has been argued for by Chierchia (1984) and others. At the same time, a propositional denotation is possible, without changing the linear logic portion of the Glue semantics for equi verbs. The only change is in the meaning language. This analysis therefore offers a better solution than Kehler et al.'s (1999) proposal that paths in f-structures contribute resoures, because it involves no modification of Glue theory and it has the empirical advantage of allowing the property denotation for clausal equi complements, which the Kehler et al. analyis does not allow.

Second, the analysis in this paper exploits LFG's parallel projection architecture to capture the anaphoric binding facts and typological facts which have been a problem for other formulations of the property theory of equi semantics. Binding is handled at f-structure as usual, with the structure-shared controlled subject providing a local antecedent for a reflexive in the embedded clause. But, the projection architecture and Glue language allow for a semantics in which the resource for the controlled subject does not have to be contributed, which also naturally leads to the right analysis in the meaning language. The second problem, that of anaphoric control, can also be handled naturally. Languages that use anaphoric control for equi simply need an additional Glue premise which consumes the controllee pronominal and relates it to the controller.

Third, this semantics is robust and can handle a range of empirical phenomena, including embedded equi verbs, the interaction with the semantics for raising verbs, and de dicto/de re differences between equi and raising verbs. As a final aside, the analysis suggests a general program for dealing with structure-sharing, which has preliminarily been extended to the analysis of VP conjunction and right node raising in our implementation at PARC.

References

Andrews, Avery (1982). The representation of case in Modern Icelandic. In Bresnan (1982b), (pp. 427–503).

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

Barwise, Jon, and Robin Cooper (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy*, *4*, 159–219.

Bresnan, Joan (1982a). Control and complementation. *Linguistic Inquiry*, *13*, 343–434.

Bresnan, Joan (ed.) (1982b). *The mental representation of grammatical relations*. Cambridge, MA: MIT Press.

Bresnan, Joan (2000). *Lexical-Functional Syntax*. Cambridge, MA: Blackwell.

Chierchia, Gennaro (1984). 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 (1985). Formal semantics and the grammar of predication. *Linguistic Inquiry*, *16*, 417–443.

Chierchia, Gennaro, and Pauline Jacobson (1986). Local and long distance control. In *Proceedings of NELS 16*.

Dalrymple, Mary (1993). *The syntax of anaphoric binding*. Stanford, CA: CSLI Publications.

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

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

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

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

Dowty, David R., Robert E. Wall, and Stanley Peters (1981). *Introduction to montague semantics*. Dordrecht: Kluwer Academic Publishers.

Falk, Yehuda (to appear). *Lexical-Functional Grammar: An introduction to parllel constraint-based syntax*. Stanford, CA: CSLI Publications.

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

Kaplan, Ronald M. (1987). Three seductions of computational psycholinguistics. In P. Whitelock, M. M. Wood, H. L. Somers, R. Johnson, and P. Bennett (eds.), *Linguistic theory and computer applications*, (pp. 149–181). London: Academic Press. Reprinted in Dalrymple et al. (1995), 339–367.

Kaplan, Ronald M. (1995). The formal architecture of Lexical-Functional Grammar. In Mary Dalrymple, Ronald M. Kaplan, John T. Maxwell, and Annie Zaenen (eds.), *Formal issues in Lexical-Functional Grammar*, (pp. 7–27). Stanford, CA: CSLI.

Kaplan, Ronald M., and Joan Bresnan (1982). Lexical-Functional Grammar: A formal system for grammatical representation. In Bresnan (1982b), (pp. 173–281).

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

Montague, Richard (1974). The proper treatment of quantification in ordinary English. In *Formal philosophy: Selected papers of Richard Montague*. New Haven: Yale University Press. Edited and with an introduction by Richmond H. Thomason.

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.

Sag, Ivan A., and Carl Pollard (1991). An integrated theory of complement control. *Language*, *67*, 63–113.

Zec, Draga (1987). On obligatory control in clausal complements. In Masayo Iida, Stephen Wechsler, and Draga Zec (eds.), *Workng papers in grammatical theory and discourse structure, vol. I: Interactions of morphology, syntax, and discourse*, (pp. 139–168). CSLI Publications.

Appendix: Glue Proofs with Meanings

(29d′)

$$\dfrac{\dfrac{\text{gonzo} : g_\sigma \quad \lambda x\lambda P.\text{promise}(x,P) : g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)}{\lambda P.\text{promise}(\text{gonzo},P) : (g_\sigma \multimap h_\sigma) \multimap f_\sigma}{\multimap_{\mathcal{E}}} \quad \dfrac{\dfrac{\dfrac{[z : g_\sigma]^1 \quad \lambda y\lambda Q.\text{try}(y,Q) : g_\sigma \multimap ((g_\sigma \multimap i_\sigma) \multimap h_\sigma)}{\lambda Q.\text{try}(z,Q) : (g_\sigma \multimap i_\sigma) \multimap h_\sigma}{\multimap_{\mathcal{E}}} \quad \lambda w.\text{go}(w) : g_\sigma \multimap i_\sigma}{\text{try}(z,\lambda w.\text{go}(w)) : h_\sigma}{\multimap_{\mathcal{E}}}}{\lambda z.\text{try}(z,\lambda w.\text{go}(w)) : g_\sigma \multimap h_\sigma}{\multimap_{\mathcal{I},1}}}{\text{promise}(\text{gonzo},\lambda z.\text{try}(z,\lambda w.\text{go}(w))) : f_\sigma}{\multimap_{\mathcal{E}}}$$

(31d′)

$$\dfrac{\dfrac{\dfrac{\dfrac{[z : g_\sigma]^1 \quad \lambda w.\text{vanish}(w) : g_\sigma \multimap i_\sigma}{\text{vanish}(z) : i_\sigma}{\multimap_{\mathcal{E}}} \quad \lambda \mathcal{P}.\text{seem}(\mathcal{P}) : i_\sigma \multimap h_\sigma}{\text{seem}(\text{vanish}(z)) : h_\sigma}{\multimap_{\mathcal{E}}}}{\lambda z.\text{seem}(\text{vanish}(z)) : g_\sigma \multimap h_\sigma}{\multimap_{\mathcal{I},1}} \quad \dfrac{\text{gonzo} : g_\sigma \quad \lambda y\lambda Q.\text{try}(y,\ Q) : g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)}{\lambda Q.\text{try}(\text{gonzo},\ Q) : (g_\sigma \multimap h_\sigma) \multimap f_\sigma}{\multimap_{\mathcal{E}}}}{\text{try}(\text{gonzo},\ \lambda z.\text{seem}(\text{vanish}(z))) : f_\sigma}{\multimap_{\mathcal{E}}}$$

(32d′)

$$\dfrac{\dfrac{\dfrac{\text{gonzo} : g_\sigma \quad \lambda y\lambda Q.\text{try}(y,\ Q) : g_\sigma \multimap ((g_\sigma \multimap i_\sigma) \multimap h_\sigma)}{\lambda Q.\text{try}(\text{gonzo},\ Q) : (g_\sigma \multimap i_\sigma) \multimap h_\sigma}{\multimap_{\mathcal{E}}} \quad \lambda w.\text{go}(w) : g_\sigma \multimap i_\sigma}{\text{try}(\text{gonzo},\ \lambda w.\text{go}(w)) h_\sigma}{\multimap_{\mathcal{E}}} \quad \lambda \mathcal{P}.\text{seem}(\mathcal{P}) : h_\sigma \multimap f_\sigma}{\text{seem}(\text{try}(\text{gonzo},\ \lambda w.\text{go}(w))) : f_\sigma}{\multimap_{\mathcal{E}}}$$

(33d′)

$$\dfrac{\dfrac{\lambda Q \exists z.[\text{goblin}(z) \wedge Q(z)] : (g_\sigma \multimap X) \multimap X \quad \dfrac{\dfrac{\dfrac{\dfrac{\text{gonzo} : i_\sigma \quad \lambda y\lambda x.\text{pinch}(x,y) : i_\sigma \multimap (g_\sigma \multimap h_\sigma)}{\lambda x.\text{pinch}(x,\text{gonzo}) : g_\sigma \multimap h_\sigma}{\multimap_{\mathcal{E}}} \quad [w : g_\sigma]^1}{\text{pinch}(w,\text{gonzo}) : h_\sigma}{\multimap_{\mathcal{E}}} \quad \lambda \mathcal{P}.\text{seem}(\mathcal{P}) : h_\sigma \multimap f_\sigma}{\text{seem}(\text{pinch}(w,\text{gonzo})) : f_\sigma}{\multimap_{\mathcal{E}}}}{\lambda w.\text{seem}(\text{pinch}(w,\text{gonzo})) : g_\sigma \multimap f_\sigma}{\multimap_{\mathcal{I},1}}}{\exists z.[\text{goblin}(z) \wedge \lambda w.\text{seem}(\text{pinch}(w,\ \text{gonzo}))(z)] : f_\sigma}{\multimap_{\mathcal{E}}, X = f_\sigma}}{\exists z.[\text{goblin}(z) \wedge \text{seem}(\text{pinch}(z,\ \text{gonzo}))] : f_\sigma}$$

(33e′)

$$
\dfrac{\lambda\mathcal{P}.\text{seem}(\mathcal{P}) : h_\sigma \multimap f_\sigma \qquad \dfrac{\dfrac{\text{gonzo} : i_\sigma \qquad \lambda y \lambda x.\text{pinch}(x,y) : i_\sigma \multimap (g_\sigma \multimap h_\sigma)}{\lambda x.\text{pinch}(x, \text{gonzo}) : g_\sigma \multimap h_\sigma}\multimap_{\mathcal{E}} \qquad \lambda Q \exists z.[\text{goblin}(z) \wedge Q(z)] : (g_\sigma \multimap X) \multimap X}{\exists z.[\text{goblin}(z) \wedge \lambda x.\text{pinch}(x, \text{gonzo})(z)] : h_\sigma}\multimap_{\mathcal{E}}, X = h_\sigma}{\text{seem}(\exists z.[\text{goblin}(z) \wedge \text{pinch}(z,\ \text{gonzo})]) : f_\sigma}\multimap_{\mathcal{E}}
$$

(34d′)

$$
\dfrac{\dfrac{\lambda Q \exists z.[\text{goblin}(z) \wedge Q(z)] : (g_\sigma \multimap X) \multimap X \qquad \dfrac{\dfrac{\dfrac{[u : g_\sigma]^1 \qquad \lambda w \lambda P.\text{try}(w,\ P) : g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)}{\lambda P.\text{try}(u,\ P) : (g_\sigma \multimap h_\sigma) \multimap f_\sigma}\multimap_{\mathcal{E}} \qquad \dfrac{\text{gonzo} : i_\sigma \qquad \lambda y \lambda x.\text{pinch}(x,y) : i_\sigma \multimap (g_\sigma \multimap h_\sigma)}{\lambda x.\text{pinch}(x, \text{gonzo}) : g_\sigma \multimap h_\sigma}\multimap_{\mathcal{E}}}{\text{try}(u, \lambda x.\text{pinch}(x, \text{gonzo})) : f_\sigma}\multimap_{\mathcal{E}}}{\lambda u.\text{try}(u, \lambda x.\text{pinch}(x, \text{gonzo})) : g_\sigma \multimap f_\sigma}\multimap_{\mathcal{I},1}}{\exists z.[\text{goblin}(z) \wedge \lambda u.\text{try}(u,\ \lambda x.\text{pinch}(x,\ \text{gonzo}))(z)] : f_\sigma}\multimap_{\mathcal{E}}, X = f_\sigma}{\exists z.[\text{goblin}(z) \wedge \text{try}(z,\ \lambda x.\text{pinch}(x,\ \text{gonzo}))] : f_\sigma}
$$