WHY GLUE A DONKEY TO AN F-STRUCTURE WHEN YOU CAN CONSTRAIN AND BIND IT INSTEAD

Miltiadis Kokkonidis
Computer Laboratory, University of Cambridge, UK
and
Meta Research, Athens, Greece

Proceedings of the LFG05 Conference
University of Bergen
Miriam Butt and Tracy Holloway King (Editors)

2005

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

Abstract

The semantic treatment of anaphora using λ-DRT with Glue which I present combines the strengths of both by assigning to each the task where it arguably fares best: Glue composes meanings and DRT deals with anaphoric resolution. Key to this approach is a simple first-order system for λ-DRT that allows LFG syntactic constraints to be transferred into the dynamic representation language. This parallels the transfer of such constraints into Glue types. Whereas approaches treating anaphora using Glue context management take advantage of this transfer, an earlier approach also leaving the treatment of anaphora to a compositional variant of DRT failed to account for syntactically-motivated anaphoric resolution constraints. On the other hand, the existing Glue context management approaches come not only at the cost of coupling context management with meaning composition, but also at the additional cost of the various remedies to the problems this uneasy cohabitation results in. The best of these approaches and the one presented here are currently very similar with respect to the range of phenomena they can correctly account for, but there are reasons to believe the latter is more scalable.

1 Introduction

DRT (Kamp, 1981; Kamp and Reyle, 1993) and its various compositional variants such as λ-DRT(Bos et al., 1994; Kohlhase et al., 1995) treat anaphors as underspecified terms to be resolved to bound variables. On the other hand, it is common practice in Glue literature (Dalrymple et al., 1999; Crouch and van Genabith, 1999; Dalrymple, 2001) to treat anaphora resolution within some system of anaphoric context management operating alongside meaning composition.

Not only is the common Glue approach more complex computationally and intuitively, but it has also proven hard to get right. The most complete treatment of this genre, that of Dalrymple (2001), resorts to imitating DRT, adapting its scoping rules and maintaining a DRT-style context of discourse referents.

If a DRT-style treatment is the aim, then in the spirit of a modular approach, especially given the close historic links of the Glue community with the LFG community, a design using a compositional variant of DRT as the meaning representation Glue has to work with makes much more sense. As a major part of LFG's success is that it allows talking separately about functional and constituent structure, recognising the importance of both and showing how they relate to each other, one would expect the same approach to be taken in semantics. Combining Glue with a system such as λ-DRT leads to a simple, modular design. Glue can be used to combine the λ-DRT expressions corresponding to the meaning contributions of the parts into an anaphorically underspecified meaning of the whole, which can then lead to different fully-specified meanings by means of DRT anaphoric resolution. The separation of meaning composition from anaphoric resolution is not a novel idea by any means. It underlies the design of compositional variants of DRT that predate the Glue attempts of dealing with both issues together.

Given the popularity of DRT within the LFG community one could reasonably expect that Glue would be used to replace other ways of combining meaning expressions given in λ-DRT (or some other DRT variant) and that anaphoric resolution would be left to DRT. That would not only be reasonable; it would also have been remarkably straightforward. The only technical challenge would be making DRT respect syntactic anaphoric constraints as expressed in LFG.

However, that was not what happened. A fascination with linear logic and the success of Glue in composing meanings lead to research trying to use it for a variety of other somewhat related tasks including anaphora resolution. Even though the work of Dalrymple et al. (1999) on using Glue for anaphora was already showing some signs that doing something like that could be problematic, the limited success that early approach had was taken as an indication that with more work more complete treatments could emerge. The modular solution based on a dynamic meaning representation language first appeared in the work of van Genabith and Crouch (1997), but Glue research remained focused on the context management approach.

Due to recent work advancing the context management approach (Dalrymple, 2001) on one hand and the present work on the other, the two approaches to the semantic treatment of anaphora have now reached the same (not very high) standards of coverage of the phenomenon. This is a welcome development as the previous *status quo* was quite unsatisfactory. Earlier all-Glue attempts had insufficiently developed context management, but could take into account syntactic anaphoric constraints. The proposal of van Genabith and Crouch (1997) got advanced anaphoric management for free by combining Glue with CDRT but failed to address the issue of enforcing LFG syntactic constraints during anaphoric resolution.

The technical contribution of the present work lies not in the straightforward combination of Glue with λ-DRT, but in showing how syntactic information can be imported into the meaning representation language, thus enabling the enforcement of syntactic constraints during anaphoric resolution. This technique is fairly generic and is tied to neither LFG, Glue, nor λ-DRT, but here it will be used to link λ-DRT discourse referents to their corresponding f-structures[1] in order to enforce LFG syntactic anaphoric binding constraints (Dalrymple, 1993) within DRT.

Section 2 shows how LFG, Glue and λ-DRT are combined and how they deal with a simple example involving compositional ambiguity. It will be interesting to note how minimally intrusive the proposed technique is: nothing changes as far as Glue is concerned and all that is added in DRSs is a simple type for each discourse referent. Section 3 discusses how syntactic anaphoric constraints are imported into λ-DRT representations. The one change to the well-formedness rules of DRT is that they now require that the two variables appearing on the left and on the right of an equals sign have the same type. A function mapping f-structures to anaphoric indices is used to encode syntactic anaphoric constraints in a way that can be conveniently combined with the simple DRT system. Section 4 discusses earlier approaches to treatments of anaphora in the Glue literature. Finally, Section 5 argues for the approach presented here, claiming it is simpler and more scalable than its Glue context management counterpart.

[1] Not only is the technique not tied to Glue in any way, it bypasses it completely.

2 Setting the scene: LFG - Glue - λ-DRT

Below we have a sentence and the semantic representations corresponding to its two readings as given by our approach. Comparing these with their plain λ-DRT counterparts, the only addition is the type assignments for the discourse referents. Given that in plain λ-DRT they would all be treated as having the same entity type, the only real difference is that in the approach presented here the types of discourse referents are differentiated according to the anaphoric index associated with their corresponding f-structure. It really is that simple. However, as we will see, in this case at least, with simplicity comes power. Having given away the ending, let us see how we get there. We will start by seeing how LFG, Glue and λ-DRT combine.

$$\text{Every boy loves a girl.} \tag{1}$$

Reading 1

$$\begin{array}{|c|}\hline \\ \hline \begin{array}{|c|}\hline b : e_{\hat{s}} \\ \hline boy(b) \\ \hline \end{array} \Rightarrow \begin{array}{|c|}\hline g : e_{\hat{o}} \\ \hline girl(g) \\ love(b,g) \\ \hline \end{array} \\ \hline \end{array}$$

Reading 2

$$\begin{array}{|c|}\hline g : e_{\hat{o}} \\ \hline girl(g) \\ \begin{array}{|c|}\hline b : e_{\hat{s}} \\ \hline boy(b) \\ \hline \end{array} \Rightarrow \begin{array}{|c|}\hline \\ \hline love(b,g) \\ \hline \end{array} \\ \hline \end{array}$$

Our fundamental assumption will be that the meaning of a sequence of words is composed of the meanings of the words plus the meanings of certain syntactic constructs found in it (e.g. relative clauses) which we recognise as making a semantic contribution when the contributions of the words alone cannot account for its composite meaning. The question then is how we get from sequences of words to semantic representations for these sequences in a precise, systematic fashion.

Our first step will be to assign a meaning to each word of the given sentence. Using plain λ-DRT as our semantic notation, the meaning assignment for (1) is:

every $\quad \lambda N.\lambda P.\begin{array}{|c|}\hline \\ \hline (\begin{array}{|c|}\hline b \\ \hline \\ \hline \end{array} \sqcup N(b)) \Rightarrow P(b) \\ \hline \end{array}$

boy $\quad \lambda b.\begin{array}{|c|}\hline \\ \hline boy(b) \\ \hline \end{array}$

loves $\quad \lambda b.\lambda g.\begin{array}{|c|}\hline \\ \hline love(b,g) \\ \hline \end{array}$

a $\quad \lambda N'.\lambda P'.\begin{array}{|c|}\hline g \\ \hline \\ \hline \end{array} \sqcup N'(g) \sqcup P'(g)$

girl $\quad \lambda g.\begin{array}{|c|}\hline \\ \hline girl(g) \\ \hline \end{array}$

Using the words as shorthands for their meanings we could write the two readings of (1) as

$$\textit{every boy } \lambda x.\ \textit{a girl (loves } x\textit{)}$$[2]

and

$$\textit{a girl } \lambda y.\ \textit{every boy } \lambda x.\ \textit{loves } x\ y$$

respectively.[3] The question now is how we get these two readings while excluding other combinations that are correct DRSs but do not correspond to a possible interpretation of (1). This is where Glue enters the picture. Glue is a type system that is used for discovering the well-formed combinations of meaning contributions. Its resource sensitivity immediately rules out combinations where a meaning contribution is arbitrarily duplicated or omitted and more importantly its label sensitivity rules out combinations where meaning contributions are combined in a manner not justified by the syntactic structure of the given word sequence. Kokkonidis (2006) gives details and examples.

This brings us nicely to our next step in getting from (1) to its meaning. While there is also a long tradition of studying syntax independently of semantics, the study of semantics usually presupposes syntactic structures have been assigned to word sequences that are to subsequently be analysed semantically. We have approached the problem from the point of view of semantics. Now we will approach it from a syntactic viewpoint.

$$
\begin{array}{llrclrcl}
\mathrm{S} & \longrightarrow & & & \mathrm{NP} & & & \mathrm{VP} \\
 & & (\uparrow\ \textsc{subj}) & = & \downarrow & \uparrow & = & \downarrow \\
 & & & & & & & \\
\mathrm{VP} & \longrightarrow & & & \mathrm{V} & & & \mathrm{NP} \\
 & & \uparrow & = & \downarrow & (\uparrow\ \textsc{obj}) & = & \downarrow \\
 & & & & & & & \\
\mathrm{NP} & \longrightarrow & & & \mathrm{Det} & & & \mathrm{N} \\
 & & (\uparrow\ \textsc{spec}) & = & \downarrow & (\uparrow\ \textsc{pred}) & = & \downarrow
\end{array}
$$

LFG (Kaplan and Bresnan, 1982; Dalrymple, 2001) views syntax not only in terms of c(onstituent) structure, but also in terms of f(unctional) structure. LFG researchers have often argued that there are generalisations that can be expressed in terms of f-structure that are not easily expressed in c-structure terms. What is particularly interesting for our purposes is how c-structure and f-structure relate to each other and how Glue and our enhanced λ-DRT draw sufficient information from them to avoid erroneous readings.

LFG constituent structure rules are expressed in a notation not very different from that used in other formalisms. Below each LFG constituent structure rule are the constraints for forming the corresponding functional structure. Up and down arrows are metavariables standing for f-structures. Their use is best understood if we consider these constraints as being evaluated on the arcs of the syntax tree resulting from the application of the constituent structure rules above them: the up arrows stand for the f-structure of the category above (the one on the left-hand side of the constituent structure rule) and the down arrows stand for the f-structure of the category below (the right-hand side category below which the constraint is written). The f-structure for a sentence is the minimal f-structure that satisfies all f-structure constraints.

[2] The η-equivalent expression *every boy* $\lambda x.$ *a girl* $\lambda y.$ *loves* $x\ y$ is perhaps more familiar to some readers.

[3] Strictly speaking these λ-DRT expressions are not identical, but β-equivalent to the DRT readings originally given. We get the original expressions by β-reduction.

The resulting f-structure f for (1) is shown below. The f-structure for the VP is the same as that for the sentence i.e. f, the f-structure for the subject NP is $s = f$ SUBJ. Since the f-structure for the VP is f, the f-structure for the object NP is $o = f$ OBJ.

$$f : \left[\begin{array}{lll} \textit{PRED} & \text{'LOVE'} & \\ \textit{SUBJ} & s : \left[\begin{array}{ll} \textit{SPEC} & \text{'EVERY'} \\ \textit{PRED} & \text{'BOY'} \end{array}\right] \\ \textit{OBJ} & o : \left[\begin{array}{ll} \textit{SPEC} & \text{'A'} \\ \textit{PRED} & \text{'GIRL'} \end{array}\right] \end{array}\right]$$

The SPEC and PRED attributes of the NPs and their values, as well as the VP (and sentence) PRED attribute and its value all come from the lexical entries. The first line of a lexical entry gives the syntactic category of the word and the f-structure constraints it comes with.

every Det

$$(\uparrow \text{ SPEC}) = \text{'EVERY'}$$
$$\textit{every} : (e_{\uparrow_{\text{label}}} \multimap t_{\uparrow_{\text{label}}}) \multimap ((e_{\uparrow_{\text{label}}} \multimap \alpha) \multimap \alpha)$$
$$\lambda N.\lambda P. \begin{array}{|c|}\hline \\ \hline (\begin{array}{|c|}\hline b : e_{\hat{\uparrow}} \\ \hline \\ \hline \end{array} \sqcup N(b)) \Rightarrow P(b) \\ \hline \end{array}$$

boy N

$$(\uparrow \text{ PRED}) = \text{'BOY'}$$
$$\textit{boy} : e_{\uparrow_{\text{label}}} \multimap t_{\uparrow_{\text{label}}}$$
$$\lambda b. \begin{array}{|c|}\hline \\ \hline \textit{boy}(b) \\ \hline \end{array}$$

loves V

$$(\uparrow \text{ PRED}) = \text{'LOVE'}$$
$$\textit{loves} : e_{(\uparrow \text{ SUBJ})_{\text{label}}} \multimap (e_{(\uparrow \text{ OBJ})_{\text{label}}} \multimap t_{\uparrow_{\text{label}}})$$
$$\lambda b.\lambda g. \begin{array}{|c|}\hline \\ \hline \textit{love}(b, g) \\ \hline \end{array}$$

a Det

$$(\uparrow \text{ SPEC}) = \text{'A'}$$
$$a : (e_{\uparrow_{\text{label}}} \multimap t_{\uparrow_{\text{label}}}) \multimap ((e_{\uparrow_{\text{label}}} \multimap \beta) \multimap \beta)$$
$$\lambda N.\lambda P. \begin{array}{|c|}\hline g : e_{\hat{\uparrow}} \\ \hline \\ \hline \end{array} \sqcup N(g) \sqcup P(g)$$

girl N

$$(\uparrow \text{ PRED}) = \text{'GIRL'}$$
$$\textit{girl} : e_{\uparrow_{\text{label}}} \multimap t_{\uparrow_{\text{label}}}$$
$$\lambda g. \begin{array}{|c|}\hline \\ \hline \textit{girl}(g) \\ \hline \end{array}$$

While grammar rules may have semantic content, most do not. On the other hand most words do. The third line of each of the lexical entries contains the meaning of the word expressed in λ-DRT, this time complete with simple types for discourse referents. The second line contains a meaning placeholder (the word itself) and its compositional (Glue) type. This is the interface between syntax and semantics with respect to semantic composition. The simple types added to λ-DRT discourse referents constitute the interface between syntax and the dynamic semantics representation with respect to anaphoric resolution.

The Glue typing context Γ for a sentence is formed of the meaning placeholders and their types.[4] What a Glue implementation does is derive all $\beta\eta$-irreducible terms T of type t_f (where f is the f-structure for the sentence) such that $\Gamma \vdash T : t_f$. Replacing the meaning placeholders with the corresponding meaning at any time gives the composite meaning Glue has formed, although usually β-reduction needs to be applied also to produce more comprehensible, but otherwise equivalent, semantic expressions.

To recap, for each word we have its syntactic specification (line 1), its compositional specification (line 2) and its semantic specification (line 3).[5] This presentation deviates from the standard modern presentations of Glue (Dalrymple, 2001), but only slightly. Meaning placeholders have been introduced for a number of reasons. One is to make it clear that the Glue types do not apply to the meaning expressions but only determine how the latter can combine. Another is to emphasise that Glue does not rely on knowing the details of meaning expressions it works with. A third reason is to make the structure of the Glue derivations more evident. There are also some formal reasons of minor importance. However, one can always follow tradition and use the meaning expressions with their corresponding glue types in the derivation and even perform β-reductions at intermediate steps. The end result will be the same.

The function 'label', written as a subscript to its argument, maps an f-structure to its label. The labels acting as arguments to the base type constructors anchor the base types to f-structures. We also have variables (lowercase Greek letters) as arguments (subscripts) to the base type constructors. They are implicitly universally qualified in prenex normal form, i.e. the actual type is obtained by adding universal quantifiers for the variables on the left hand side (say in order of appearance); so the type of *every* is really $\forall\alpha.(e_{\uparrow_{\text{label}}} \multimap t_{\uparrow_{\text{label}}}) \multimap ((e_{\uparrow_{\text{label}}} \multimap \alpha) \multimap \alpha)$.

The key to the present solution to anaphora using LFG, Glue and λ-DRT is that it uses the same kind of linking between f-structures and types that Glue uses. This linkage is essential for Glue not to compose meaning in an erroneous way; it is also essential for the treatment of anaphora being proposed to avoid erroneous anaphoric binding. With reference to the title of the paper, we should note that instead of using a lot of Glue to treat anaphora, we can thus allow our version of λ-DRT with simple discourse referent types to take care of it through its simple and elegant resolution mechanism based on variable binding while also enforcing the relevant syntactic constraints.

[4] If the same meaning placeholder name as it appears in the lexical rules appears more than once in the typing context, we can number its occurrences using subscripts to make them unique within the context. In our examples this is not necessary.

[5] For certain words and syntactic constructs it could be more practical or even necessary to break a meaning contribution into smaller and simpler ones, in which case there will be two or more compositional-semantic specification pairs. For those that do not make a semantic contribution there will not be any.

Returning to our example sentence (1), given the f-structure for it we obtain the following Glue typing context Γ:

$$\begin{array}{l} \textit{every} : (e_s \multimap t_s) \multimap ((e_s \multimap \alpha) \multimap \alpha), \\ \textit{boy} : e_s \multimap t_s, \\ \textit{loves} : e_o \multimap (e_s \multimap t_f), \\ \textit{a} : (e_o \multimap t_o) \multimap ((e_o \multimap \beta) \multimap \beta), \\ \textit{girl} : e_o \multimap t_o. \end{array}$$

According to the Glue type-inference rules below

$$\Gamma \vdash \textit{every boy } \lambda x.\ \textit{a girl (loves } x)$$

and

$$\Gamma \vdash \textit{a girl } \lambda y.\ \textit{every boy } \lambda x.\ \textit{loves } x\ y.$$

These are the only semantically distinct readings available for the sentence. Replacing the meaning placeholders with their corresponding meaning in the derived terms and β-reducing we get the two readings in DRT as we had originally set out to do.

$$\frac{N : T,\ \Gamma,\ N' : T',\ \Gamma' \vdash E : T''}{N' : T',\ \Gamma,\ N : T,\ \Gamma' \vdash E : T''}\ (Exchange)$$

$$N : T \vdash N : T \quad (Axiom)$$

$$\frac{\Gamma, X : T \vdash E : T'}{\Gamma \vdash \lambda X.E : T \multimap T'}\ (\multimap Intro.)$$

$$\frac{\Gamma \vdash E : T' \multimap T \quad \Gamma' \vdash E' : T'}{\Gamma, \Gamma' \vdash E\ E' : T}\ (\multimap Elim.)$$

$$\frac{\Gamma \vdash E : \forall V.T}{\Gamma \vdash E : T[V := L]}\ (\forall Elim.)$$

Figure 1: First-Order Glue Inference Rules

Notes:

1. The Exchange rule is unecessary if we regard the context as being a multiset.
2. The $\forall Intro$ rule is not needed and has been excluded.

We have at our disposal a tripartite framework that handles syntax (LFG), meaning composition (Glue), and semantics (λ-DRT) and we have seen it at work with a simple example. Our choice of meaning expressions and Glue types (modulo f-structure labels) guarantees the well-formedness of the resulting meaning expression.[6] The f-structure labels used as parameters in our Glue types ensure that meaning composition does not result in arbitrary semantic expressions given a multiset of meaning contributions, but all and only those readings that correspond to the given sentence or discourse. Glue pulls its weight remarkably well and has a clear, simple, yet powerful interface to LFG (and other grammar formalisms). In the following section it will be λ-DRT's turn to demonstrate the same qualities when dealing with its assigned task, anaphoric resolution.

[6] This guarantee does not cover anaphoric resolution which is an matter entirely internal to DRT in the presented approach.

3 Importing Syntactic Anaphoric Constraints

The classic DRT anaphoric resolution mechanism (Kamp, 1981; Kamp and Reyle, 1993) was based on the visibility of discourse referents. It was remarkably simple and elegant. However, in its basic form this mechanism completely disregards basic syntactic requirements such as number and gender agreement, thus potentially generating erroneous readings. There are also constraints on anaphoric resolution that are on the level of discourse. One of the strengths of DRT is in dealing with such constraints. That strength is taken full advantage of here, and so is the strength of the syntactic analysis that informs the anaphoric resolution mechanism of the constraints emanating from the syntactic form of the sentence. The latter is achieved thanks to a simple technique for importing syntactic constraints into the chosen dynamic representation language. So in neither of the two examples below will the interpretation implied by the coreference indicators be allowed, but for the first this will be thanks to DRT semantic form constraints, whereas for the second it will be thanks to the imported LFG syntactic constraints on anaphora.

$$\text{*No student}_1 \text{ arrived. He}_1 \text{ yawned.} \tag{2}$$

$$\text{*Every man}_1 \text{ likes him}_1. \tag{3}$$

The most prominent feature of the classic DRT analysis of pronouns is the introduction of a new discourse referent that comes with a condition that equates it to a question mark. Informally, the question mark may be seen as a promissory note for an accessible discourse referent. Formally, we can treat it as a metavariable ranging over discourse referents. Then according to the scoping rules of DRT, ? can only be an accessible discourse referent. We return to this shortly.

$$\begin{array}{lll} \text{himself} & \text{NP} & (\uparrow\ \textsc{pred}) = \text{`PRO'} \\ & & \textit{himself} : (e_{\uparrow_{\text{label}}} \multimap \alpha) \multimap \alpha \\ & & \lambda P. \boxed{\begin{array}{c} p : e_{\hat{\uparrow}} \\ \hline p = ? \\ male(p) \end{array}} \sqcup P(p) \end{array}$$

Positive and negative constraints for each pronoun are expressed in its lexical entry in terms of expressions involving inside-out functional uncertainty (Dalrymple, 1993). Such expressions determine which parts of the f-structure are the candidates allowed to act as antecedents (positive constraint) and which are disallowed (negative constraint). The antecedent of the reflexive pronoun 'himself' obeys the Minimal Complete Nucleus positive constraint; therefore its f-structure will have to satisfy the expression

$$((\underset{\neg(\rightarrow \text{SUBJ})}{\text{GF}^*} \quad \text{GFpro} \quad \uparrow)\ \text{GF})$$

where '↑' stands for the f-structure of the pronoun. To capture the above positive constraint for 'himself' in our typed λ-DRT, we add the following to its lexical entry:

$$\hat{\uparrow} \in \{\ \hat{L}\ \mid\ L = \ ((\underset{\neg(\rightarrow \text{SUBJ})}{\text{GF}^*} \quad \text{GFpro} \quad \uparrow)\ \ \text{GF})\ \}$$

Central to our discussion is a function $^\wedge$ from f-structures to anaphoric indices, satisfying positive and negative constraints, but otherwise assigning different indices to different f-structures. For stylistic reasons, we write $^\wedge(X)$ as $\hat{X}$. Coreference will be modelled as anaphoric index equality. As a pronoun can corefer, it is possible that $^\wedge$ will map two or more f-structures to the same index. As pronouns do not necessarily have to corefer (exophora), this will not necessarily be the case for all pronouns. The DRT condition $x = y$ is well formed if and only if $x : e_{\hat{X}}$ and $y : e_{\hat{Y}}$ are accessible discourse referents at the point the condition $x = y$ appears and $\hat{X} = \hat{Y}$.

The way syntactic anaphoric constraints are expressed in LFG is powerful, but cryptic. Dalrymple (2001) explains inside-out functional uncertainty, gives a brief overview of the LFG research on such constraints and links that discussion to Glue. However, the following examples should be easy to follow without a deep understanding of LFG and its way of dealing with syntactic constraints on anaphora.

For an example illustrating positive constraints we can take a sentence with a reflexive pronoun such as

$$\text{John hit himself.} \tag{4}$$

$$f : \begin{bmatrix} PRED & \text{‘HIT’} \\ SUBJ & s : \begin{bmatrix} PRED & \text{‘JOHN’} \end{bmatrix} \\ OBJ & o : \begin{bmatrix} PRED & \text{‘PRO’} \end{bmatrix} \end{bmatrix}$$

$$\text{constraints:} \quad \hat{o} \in \{\hat{s}\}$$

The positive constraint for $^\wedge$ simply means that $\hat{s} = \hat{o}$. So with $?$ standing for the subject discourse referent we can only have the following correct reading:

$$\boxed{\begin{array}{c} j : e_{\hat{s}},\ h : e_{\hat{o}} \\ \hline j = John \\ hit(j, h) \\ h = j \end{array}} \quad \begin{bmatrix} \hat{o} = \hat{s} \end{bmatrix}.$$

For an example involving negative constraints we can take a mini-discourse such as the following:

$$\text{An elephant saw a mouse. She frightened her.} \tag{5}$$

This is a rather interesting example as two readings should be available. The f-structure for the first sentence is:

$$f_1 : \begin{bmatrix} PRED & \text{‘SEE’} \\ SUBJ & s_1 : \begin{bmatrix} SPEC & \text{‘A’} \\ PRED & \text{‘ELEPHANT’} \end{bmatrix} \\ OBJ & o_1 : \begin{bmatrix} SPEC & \text{‘A’} \\ PRED & \text{‘MOUSE’} \end{bmatrix} \end{bmatrix}.$$

The f-structure for the second sentence is:

$$f_2 : \begin{bmatrix} PRED & \text{'FRIGHTEN'} \\ SUBJ & s_2 : \begin{bmatrix} PRED & \text{'PRO'} \end{bmatrix} \\ OBJ & o_2 : \begin{bmatrix} PRED & \text{'PRO'} \end{bmatrix} \end{bmatrix}$$

$$\text{constraints:} \quad \hat{o_2} \notin \{\hat{s_2}\}$$

Before anaphoric resolution, we have two distinct question mark metavariables in our DRS.

$$\begin{array}{|c|} \hline e : e_{\hat{s_1}},\ m : e_{\hat{o_1}},\ s : e_{\hat{s_2}},\ h : e_{\hat{o_2}} \\ \hline elephant(e) \\ see(e, m) \\ mouse(m) \\ s = ? \\ frighten(s, h) \\ h = ?' \\ \hline \end{array} \quad \left[\ \hat{o_2} \notin \{\hat{s_2}\}\ \right].$$

If we have $\hat{s_2} = \hat{s_1}$ and resolve ? to e, then the negative constraint on the non-reflexive pronoun 'her' in object position in the second sentence means that $\hat{o_2} \neq \hat{s_1}$, leaving $\hat{o_2} = \hat{o_1}$ and $?' = m$ as the only option. This gets us the first reading:

$$\begin{array}{|c|} \hline e : e_{s_1},\ m : e_{o_1},\ s : e_{s_2},\ h : e_{o_2} \\ \hline elephant(e) \\ see(e, m) \\ mouse(m) \\ s = e \\ frighten(s, h) \\ h = m \\ \hline \end{array} \quad \left[\ \hat{s}_2 = \hat{s}_1,\ \hat{o}_2 = \hat{o}_1, \hat{o}_2 \notin \{\hat{s}_2\}\ \right].$$

If we have $\hat{s_2} = \hat{o_1}$ and resolve ? to m, then the negative constraint on the non-reflexive pronoun 'her' in object position in the second sentence means that $\hat{o_2} \neq \hat{o_1}$, leaving $\hat{o_2} = \hat{s_1}$ and $?' = e$ as the only option. This gets us the second reading:

$$\begin{array}{|c|} \hline e : e_{s_1},\ m : e_{o_1},\ s : e_{s_2},\ h : e_{o_2} \\ \hline elephant(e) \\ see(e, m) \\ mouse(m) \\ s = m \\ frighten(s, h) \\ h = e \\ \hline \end{array} \quad \left[\ \hat{s}_2 = \hat{o}_1,\ \hat{o}_2 = \hat{s}_1, \hat{o}_2 \notin \{\hat{s}_2\}\ \right].$$

4 Earlier work

Dalrymple et al. (1999) present the original Glue-based context management approach to anaphoric resolution. The basic idea behind that is that a pronoun makes an additional copy of the meaning of its antecedent; it does so by consuming that meaning x and producing a pair (x, x). So the semantics of a pronoun is given by the expression $\lambda x.(x, x)$. The glue type for a pronoun found at the part of the sentence f-structure labelled Y that has an antecedent at X is $e_X \multimap e_X \otimes e_Y$. This is the resource duplication Glue treatment of anaphora.[7]

As the point of using linear, rather than, say, intuitionistic, logic in Glue was that it provides resource sensitivity,[8] while as far as anaphora resolution is concerned a discourse referent that is in the current context can be referenced any number of times, one can immediately see a problem with trying to treat anaphora within Glue. The explicit resource duplication Glue treatment of anaphora cleverly addresses this problem, but this problem alone. Dalrymple et al. (1999) find that this approach does not work if sentence-by-sentence processing is assumed. An alternative approach, using the ! ('of course') linear logic modality, addresses the problem resource duplication has when sentence-by-sentence processing is assumed, but only that and at the cost of complicating Glue. Dalrymple et al. (1999) find problems with that approach too. Furthermore, neither of the two approaches takes into account that there is a difference between the anaphoric context available within a sentence and how it affects the context for other sentences. These proof-of-concept approaches address only the problems resource sensitivity causes for the treatment of anaphora within Glue.

Taking the next step in the evolution of the context management approaches, Crouch and van Genabith (1999) make a bold attempt to address intersentential anaphora issues using a para-Glue e-type anaphora context management approach. They add assignments from NP labels to e-type descriptions to the standard Glue system of the time. They also change what the final result of a derivation is in order to allow these assignments to appear alongside the meaning of a sentence at the end of the derivation. The basic idea is fairly simple. For the sentence 'A man walks' the result this approach gives is a pair. The first element is the meaning of the sentence $\exists x.man(x) \wedge walk(x)$ and the second is an assignment of the sentence's subject label to the description $\lambda x.man(x) \wedge walk(x)$. A pronoun consumes a description assignment such as the above, uses it in its meaning and also produces a new one. So the result of subsequently analysing the sentence 'He whistles' also produces a pair. The sentence meaning element is $\exists x.man(x) \wedge walk(x) \wedge whistle(x)$ while the description assignment part assigns to the subject of this sentence the description $\lambda x.man(x) \wedge walk(x) \wedge whistle(x)$.

This is indeed as simple as it should be. Unfortunately, some of the details were omitted. Interactions between quantifier scope and context assignments complicate matters. Reinforcing the arguments against entangling meaning composition with anaphoric context management, Crouch and van Genabith (1999) also identify a problem arising "from the need to build up a collection of context assignments in addition to a single meaning assignment for the sentence". They resort to a higher order solution to solve this. What started with a simple idea ended up being very complicated at the

[7]There is also an alternative version of this treatment that does not require $\otimes$ to be a part of the Glue. In that version, the meaning expression for the pronoun is $\lambda x.\lambda P.P\ x\ x$ and the corresponding glue type is $e_X \multimap (e_X \multimap e_Y \multimap t_\alpha) \multimap t_\alpha$.

[8]Kokkonidis (2006) argues that although resource sensitivity is probably a desirable feature of Glue, it is not as essential as it is believed to be.

end, while only covering simple cases. Also absent from their treatment is an account of the difference between, say ‘A man walks’ and ‘No man walks’ with respect to the anaphoric context that a subsequent sentence will have available.

The same authors address this issue elsewhere (van Genabith and Crouch, 1997) by simply using CDRT and allowing it to deal with anaphora. This is the approach that is closest to the one presented here; indeed they anticipate similar work by noting that dynamic representations other than CDRT can be used as the meaning representation language in such an approach. However, they do not address the issue of imposing syntactic constraints within the dynamic representation language whereas the Glue and para-Glue approaches did.

The approach presented by Dalrymple (2001)[9] also addresses the issue of the management of different contexts successfully, albeit at the cost of additions to standard Glue. In many ways this is a continuation of the research of Crouch and van Genabith (1999). However, a DRT-style approach is taken. In effect what this approach does is take the discourse referents universe of a DRS and stick it next to meanings derived using Glue. This may be seen as having the advantage of offering some of the benefits of DRT when other meaning representations are used. However, it does not make much sense if DRT itself is to be used as the meaning representation language. Furthermore, if one wanted to combine the characteristics of DRT with those of another representation language, an obvious solution would have been doing exactly that and using the result as the meaning representation language. Much of this is a matter of opinion and personal taste. The fact is that, historically, that was the first Glue context management approach that could control context equally well as the DRT-based approaches such that of van Genabith and Crouch (1997) and the present one. Furthermore, it was the first approach that did that and at the same time respected syntactic constraints.[10] Having said that though, it does come with notational clutter and like its predecessor cannot avoid the complications caused by combining meaning composition with context management. This is evident in the proposed lexical entries for quantifiers such as ‘nobody’ and ‘somebody’ (Dalrymple, 2001). Even additional inference rules are added to help deal with the complexities this juggling with too many balls at the same time brings. However, additional rules add complexity in their own right.

5 Conclusions

The two initial approaches described in Dalrymple et al. (1999) did not require anything more than what linear logic had to offer: one required ‘$\otimes$’ but there was also a version that did not need any extension beyond the implicative fragment, and the other required ‘!’. While the fragment of Glue needed for meaning composition is first-order (Kokkonidis, 2006) the approach of Crouch and van Genabith (1999) needs genuine higher order universal quantification to deal with the context, and interestingly enough it needed that for doing something as simple as adding something to the existing context in order to construct a new one. The approach of Dalrymple (2001) is far more drastic: it not only introduces new concepts such as the context and the meaning-context combination, as well as their counterparts on the type-system side, but also new rules

[9] The approach of Dalrymple (2001) is based on joint unpublished work with Martin van den Berg, Dick Crouch, and John Lamping.

[10] There is a problem with enforcing negative constraints in all three Glue context management approaches discussed as using an ANT attribute does not capture the transitivity of coreference. However, this problem can be solved e.g. by using a formal device similar to ‘$^\wedge$’.

for splitting meaning and context and for merging context and meaning. The evolution path seems to have taken us from essentially first-order Glue treatments that can only deal correctly with very simple cases (Dalrymple et al., 1999), to Glue plus genuine higher-order quantification (Crouch and van Genabith, 1999), to a Glue-DRT hybrid with many formal innovations (Dalrymple, 2001).

So what is the return on investment? The hybrid Glue-DRT approach of Dalrymple (2001) only covers the case of singular pronouns. How much more complication will have to be introduced to cover plural anaphora? If DRT is so good at dealing with anaphora, why not adopt it as the semantic representation and get all of it rather than trying to copy its behaviour in a hybrid Glue-DRT system?

The work of van Genabith and Crouch (1997) and the present work effortlessly tackle many problems Glue context-management approaches have found challenging by leaving them to dynamic semantic representations designed to deal with them. Combining CDRT or λ-DRT with Glue required no ingenuity whatsoever. CDRT and λ-DRT can easily work with various systems for composing meanings and Glue can work with various semantic representation languages. DRT does certain things well and Glue does other things well. They complement each other nicely. One problem not addressed by van Genabith and Crouch (1997) was that of syntactic constraints. This is addressed here.

The next step would be to provide analyses for a wider range of anaphoric phenomena. It seems that all that needs to be done if the modular approach is taken is to ensure the existing DRT analyses for this wider range of phenomena fit in well with the type system proposed. On the other hand, it seems that if one takes the approach of Dalrymple (2001) much more of DRT would have to be incorporated into the Glue-DRT hybrid system, most likely at the cost of even more complexity in order to achieve comparable results.

Complexity not only makes the system more difficult to explain, but it also hinders further development. Modular design is usually a good way of managing complexity, and in treating semantic composition and anaphoric resolution separately it certainly seems to have helped keep it to manageable levels. First-order Glue suffices for the composition of meanings. Only very simple types and a new well-formedness requirement on the equals sign were added to λ-DRT. There are no strange interactions and conflicting requirements to be dealt with. This means that one can concentrate on dealing with the phenomena, rather than problems with the formalism.

Acknowledgements

I am grateful to Mary Dalrymple for the very idea behind this project (combining λ-DRT with Glue while somehow making sure syntactic constraints on anaphoric binding are respected), as well as a number of very useful discussions throughout its duration. Most of all, I am grateful to her for her endless enthusiasm and support. Thanks also go to Tracy Holloway King and Miriam Butt for their meticulous proofreading of an earlier version and their good advice. Needless to say, I take full responsibility for any new mistakes that have, no doubt, tried to replace all those they caught. Last but not least, I thank the audience at LFG05 for their comments and questions and the local organising committee for their hospitality.

References

Bos, J., E. Mastenbroek, S. McGlashan, S. Millies, and M. Pinkal: 1994, 'A compositional DRS-based formalism for NLP applications'. In: *Proceedings of the International Workshop on Computational Semantics*. Tilburg.

Crouch, R. and J. van Genabith: 1999, *Context Change, Underspecification, and the Structure of Glue Language Derivations'*. In Dalrymple (1999).

Dalrymple, M.: 1993, *The Syntax of Anaphoric Binding*, No. 36 in CSLI Lecture Notes. CA: Stanford.

Dalrymple, M. (ed.): 1999, *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. MIT Press.

Dalrymple, M.: 2001, *Lexical Functional Grammar*, No. 42 in Syntax and Semantics Series. Academic Press.

Dalrymple, M., J. Lamping, F. C. Pereira, and V. Saraswat: 1999, *Quantification, Anaphora, and Intensionality*. In Dalrymple (1999).

Kamp, H.: 1981, 'A theory of truth and representation'. *Formal Methods in the Study of Language*.

Kamp, H. and U. Reyle: 1993, *From Discourse to Logic*. Dordrecht: Kluwer.

Kaplan, R. M. and J. Bresnan: 1982, 'Lexical Functional Grammar: A formal system for grammatical representation'. In: J. Bresnan (ed.): *The Mental Representation of Grammar Relations*. MIT Press, pp. 173–281.

Kohlhase, M., S. Kuschert, and M. Pinkal: 1995, 'A type-theoretic semantics for λ-DRT.'. In: *Proceedings of the Tenth Amsterdam Colloquium*.

Kokkonidis, M.: 2006, 'First-Order Glue'. *Journal of Logic, Language and Information*. To appear.

van Genabith, J. and R. Crouch: 1997, 'How to glue a donkey to an f-structure or porting a dynamic meaning representation language into LFG's linear logic based glue language semantics'. In: *Proceedings of the International Workshop for Computational Semantics*. Tilburg, pp. 52–65.