THE “LOST” READING OF CONTROL SENTENCES AND PLURAL SEMANTICS IN GLUE

Genady Beryozkin and Nissim Francez

Department of Computer Science
Technion, Israel Institute of Technology

Proceedings of the LFG04 Conference
University of Canterbury
Miriam Butt and Tracy Holloway King (Editors)
2004

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

Abstract

This paper presents a formal semantic analysis of the plural reading of split control sentences in the context of the propositional theory of control in the framework of LFG and Glue. For sentences that involve a collective controlled verb, the proposed analysis provides, for the first time, an adequate semantic derivation that does not require anaphoric resolution of the understood subject.

Two extensions of the existing frameworks are proposed in order to derive the meaning of such sentences. The concept of *multi-functional control* proposes to represent the contribution of multiple grammatical functions to the understood subject of the controlled sentence as a set of f-structures. Such a set can be assigned non-distributive features that describe the understood subject, such as the semantic number. The meaning of multi-functional control sets is derived using *meta meaning constructors* that extend the semantic derivation process by providing the means to derive the meaning of f-structure sets when the number of set elements is a-priori unknown. We also show how resource sensitivity issues can be resolved by employing the multiplicative fragment of the underlying linear logic.

As an additional result, meta-meaning constructors can be used to solve the problem of coordination of more than two conjuncts.

1 Introduction

Split control sentences such as (1) and (2) are sentences in which both the grammatical subject and object jointly contribute to the meaning of the understood subject. Split control sentences may have a single (plural) reading, as in (1) or both singular and plural readings as in (2) (John may or may not accompany Mary upon leaving the restaurant).

(1) *John persuaded Mary to meet in the restaurant*

(2) *John persuaded Mary to leave the restaurant*

Meaning derivation of split control sentences in LFG's Glue interface is challenging in several respects. The first question to consider is anaphoric vs. functional control. While anaphoric control is commonly used in LFG analyses of equi verbs, some phenomena are better described using functional control. For example, the intransitive verb *meet* requires a semantically plural understood subject, as shown in (3), and this restriction is best described using functional control, as shown in section 3.

(3) (a) ** John tried to meet in the afternoon*
(b) *The committee tried to meet in the afternoon*

Asudeh (to appear) also proposes a functional control analysis for some English equi verbs, but points out that a functional control analysis cannot be used to describe split antecedents, since the understood subject can be shared with only one grammatical function of the matrix sentence. In order to overcome this inherent restriction of functional control, we propose to extend the idea of functional control analysis in a way that would account for split control sentences. In this paper we introduce a *multi-functional control* analysis, that uses a set to represent the contributions of the different grammatical functions to the understood subject.

The next step in the meaning derivation is performed at the semantic level. While the meaning derivation in the property approach (Chierchia, 1984; Dowty, 1985) is straightforward, the propositional approach (Pollard and Sag, 1994; Dalrymple, 2001) poses two kinds of challenges. First, the meaning of the understood subject must be computed, while accounting for the possibility of an unlimited number of set elements. Handling sets of unlimited size is problematic in the current Glue analysis and relies on the use of the '!' (of course) operator. We address this challenge by presenting *meta-meaning constructors* that can also be used to derive the meaning of noun phrase coordination. The second challenge is handling resource sensitivity issues, which is addressed by duplicating the required resources in the multiplicative fragment of the linear logic.

The objective of this work is to provide the formal framework that allows the derivation of the plural reading of sentences such as (1). When such a reading constitutes the only possible reading our framework correctly derives its semantics. In other cases[1] both readings are allowed — the plural reading is derived using our framework and other readings can be derived as alternatives.

The rest of the paper is organized as follows: Section 2 presents the basic background and surveys some relevant work. Section 3 discusses the semantic number of noun phrases and its effect on the grammaticality of control sentences. Section 4 briefly presents the c-structure rules that are used in derivation of control sentences. Section 5 presents the multi-functional control approach. Section 6 presents our approach to the meaning derivation of sets. Section 7 provides the meaning derivation of multi-functional control sentences using meta-meaning constructor that were presented in section 6. Section 8 concludes.

[1] In some cases the plural reading is the preferred, but not the only possible reading.

2 Preliminaries and previous work

2.1 Glue

The standard LFG analysis (Kaplan and Bresnan, 1982; Bresnan, 2001; Dalrymple, 2001; Falk, 2001) represents the sentence as a combination of a constituent (c-)structure and a functional (f-)structure. Since LFG is a syntactic theory, neither structure is intended to represent the meaning of the sentence. The Glue language (Dalrymple, 1999, 2001) provides the bridge between the f-structure and the sentence meaning by providing the means for deriving the sentence meaning from the meanings of its constituents, based on their grammatical function in the f-structure.[2]

Before deriving the meaning, the meaning representation language must be chosen. In this work we use model theoretic semantics (Montague, 1970, 1973) in conjunction with the simply typed λ-calculus (Hindley, 1997) as the meaning representation language. The type system is built from two basic types — the t type represents a truth value and the e type represents an entity in the model's domain. Higher types are build upon these two basic types. Other possibilities for meaning representation language exist as well (e.g., intentional logic (Montague, 1973) and DRT (Kamp, 1981)).

Glue makes use of linear logic (Girard, 1987) in the meaning derivation process. Linear logic is a resource-sensitive logic, which makes Glue meaning derivations resource sensitive too. The essence of a linear resource logic is that each premise must be used exactly once in the derivation. That is, premises cannot be freely discarded or duplicated. For example, in propositional logic, $\{a, a \rightarrow b, a \rightarrow c\} \vdash b$. However the linear logic counterpart doesn't have any valid derivation:

$\{a, a \multimap b, a \multimap c\} \nvdash b$	$a \multimap c$ cannot be discarded
$\{a, a \multimap b, b \multimap c\} \vdash c$	all premises are used exactly once
$\{a, a \multimap b, a \multimap b \multimap c\} \nvdash c$	a cannot be used twice

Glue extends the LFG projections infrastructure by presenting the semantic (s-)structure and a mapping function (σ) that maps elements of the f-structure to elements of the s-structure. The σ-projection of an f-structure is denoted by a σ index (e.g., the σ-projection of some f-structure j is j_σ). Apart from representing the semantic data, s-structures also serve as the premises of the Glue linear logic.[3] Each such premise is then paired with a meaning expression in the meaning representation language through the Curry-Howard isomorphism (Curry and Feys, 1958; Howard, 1980) and such a pair is called a *meaning constructor*. Each derivation step in the linear logic is therefore accompanied by the appropriate function application (or abstraction) on the meaning side.[4]

This work uses the multiplicative fragment of linear logic, which includes the linear implication operator '$\multimap$' and the linear conjunction operator '$\otimes$'. The elimination rules of both operators are used in this work, but only the linear implication introduction rule is needed. Another required operator is the universal quantifier '$\forall$' that operates on the linear logic side of the meaning constructor, while leaving the meaning side intact. The '$\forall$' operator allows the substitution of *any* premise for the quantification variable (an example appears in figure 4; see Asudeh (2004) for a discussion of the proof term invariance for this elimination rule). Just as any other premise, once the universal quantification premise is used, it cannot be used again with a different premise substitute. Appendix A lists all linear logic rules that are used in this work.

Linear logic premises (s-structures) in meaning constructors are semantically typed and these types correspond to the types of the λ-expressions on the meaning side according to the rules of the Curry-Howard isomorphism. Therefore, an additional subscript is attached to the σ projection denotation. For example, the f_{σ_t} notation stands for the σ-projection of the f-structure f, that is associated with the semantic type t. In many cases the type index is omitted in favor of brevity.

Consider for example sentence (4) and the corresponding f-structure that appears in figure 1.

(4) *John loves Mary*

$$f\begin{bmatrix}\text{PRED} & \text{`LOVE<SUBJ,OBJ>'}\\ \text{SUBJ} & j\begin{bmatrix}\text{PRED} & \text{`JOHN'}\end{bmatrix}\\ \text{OBJ} & m\begin{bmatrix}\text{PRED} & \text{`MARY'}\end{bmatrix}\end{bmatrix}$$

Figure 1: The f-structure of *John loves Mary*

(5) $\mathsf{love}(\mathsf{john}, \mathsf{mary})$

[2] Glue is not unique to LFG. Asudeh and Crouch (2002b) describe how Glue can be used to provide semantics for HPSG.

[3] Andrews (2003) proposes to eliminate the semantic projection level, and to use the f-structures as linear logic resources.

[4] In that sense the Glue approach is similar to the approach of categorial grammar (Moortgat, 1997).

A possible meaning of (4) is (5). The meaning constructors that appear in (6) (contributed by the lexical entries) are the inputs to the meaning derivation process. The labels on the left are not part of the Glue formalism, they are used to reference the meaning constructors only.

(6) **[John]** $\mathsf{john} : j_{\sigma_e}$
[Mary] $\mathsf{mary} : m_{\sigma_e}$
[loves] $\lambda x.\lambda y.\mathsf{love}(y,x) : m_{\sigma_e} \multimap (j_{\sigma_e} \multimap f_{\sigma_t})$

Figure 2 shows the derivation proof tree of the meaning of (4), that involves two eliminations of the '$\multimap$' operator and two functional applications on the meaning side. The result is a meaning constructor, that describes the meaning of the sentence *John loves Mary*.

$$\dfrac{\mathsf{john} : j_{\sigma_e} \qquad \dfrac{\lambda x.\lambda y.\mathsf{love}(y,x) : m_{\sigma_e} \multimap (j_{\sigma_e} \multimap f_{\sigma_t}) \quad \mathsf{mary} : m_{\sigma_e}}{\lambda y.\mathsf{love}(y,\mathsf{mary}) : j_{\sigma_e} \multimap f_{\sigma_t}} \multimap_{\varepsilon}}{\mathsf{love}(\mathsf{john},\mathsf{mary}) : f_{\sigma_t}} \multimap_{\varepsilon}$$

Figure 2: The meaning derivation of *John loves Mary*

The meaning constructors appear in an uninstantiated form in the lexicon.[5] The lexical entries of *John*, *Mary* and *loves* look like:

(7) *John* NP $(\uparrow \textsc{pred}) = \text{'JOHN'}$
$\mathsf{john} : \uparrow_\sigma$

Mary NP $(\uparrow \textsc{pred}) = \text{'MARY'}$
$\mathsf{mary} : \uparrow_\sigma$

loves V $(\uparrow \textsc{pred}) = \text{'LOVE<SUBJ,OBJ>'}$
$\lambda x.\lambda y.\mathsf{love}(y,x) : (\uparrow \textsc{obj})_\sigma \multimap ((\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma)$

After the f-structure of the sentence is computed, the meaning constructors are instantiated, in a way similar to the instantiation of the functional equations.

Example (8) demonstrates the use of universal quantification in Glue that is provided by the meaning constructor of *everyone* in (9). The f-structure is presented in figure 3 and the meaning derivation is presented in figure 4. It should be noted that the universal quantifier application is usually implicit. However, in figure 4 the substitution of H by f_σ is shown in detail.

(8) *Everyone walked.*

(9) *everyone* N $(\uparrow \textsc{pred}) = \text{'EVERYONE'}$
$\lambda P.\mathsf{every}(x,\mathsf{person}(x),P(x)) : \forall H.[\uparrow_\sigma \multimap H] \multimap H$

$$f\begin{bmatrix}\textsc{pred} & \text{'WALK<SUBJ>'} \\ \textsc{subj} & g\begin{bmatrix}\textsc{pred} & \text{'EVERYONE'}\end{bmatrix}\end{bmatrix}$$

Figure 3: The f-structure of *Everyone walked*

$$\dfrac{\lambda x.\mathsf{walk}(x) : g_{\sigma_e} \multimap f_{\sigma_t} \qquad \dfrac{\lambda P.\mathsf{every}(x,\mathsf{person}(x),P(x)) : \forall H.[g_{\sigma_e} \multimap H] \multimap H}{\lambda P.\mathsf{every}(x,\mathsf{person}(x),P(x)) : [g_{\sigma_e} \multimap f_{\sigma_t}] \multimap f_{\sigma_t}} (H \Rightarrow f_\sigma)}{\mathsf{every}(x,\mathsf{person}(x),\mathsf{walk}(x)) : f_{\sigma_t}}$$

Figure 4: The meaning derivation of *Everyone walked*

[5] Lexical entries are the primary source of meaning constructors, but c-structure rules can contribute meaning constructors as well (Asudeh and Crouch, 2002a).

2.2 Control

Control sentences can be analyzed as having a subject-less sentence as a complement of the main verb. Such a complement is called the *controlled* sentence, and although it does not have an overt syntactical subject, it has an understood subject (the *controllee*), which contributes to the meaning of the controlled sentence just as a regular subject does. Control sentences can be classified according to the grammatical function that controls the understood subject (the *controller*). *Subject control* (10a) and *object control* (10b) describe sentences in which the controller is a subject or an object of the matrix sentence respectively. *Split antecedent control* (10c) describes sentences in which both grammatical functions control the understood subject.

(10) (a) *John promised Mary to become a writer*
(b) *John persuaded Mary to become a writer*
(c) *John persuaded Mary to meet at the local writers' convention*

Another distinction is made between *obligatory* (*unique*) control (all examples in (10)) and *non-obligatory* (*free*) control (11). The difference between the two lies in the fact that in obligatory control the controller is uniquely determined as a grammatical function of the matrix sentence, which is not the case in non-obligatory control.

(11) *Diane begged Daniel to leave early* [6]

The intransitive verb *meet* requires a semantically plural subject, which also restricts the possible understood subject in control sentences. Therefore (12a) is grammatical (the plural reading is possible), while (12b) is not (plural reading is not possible):

(12) (a) *John persuaded Mary to meet in the restaurant.*
(b) ** John tried to meet in the restaurant*
(c) *Mary agreed to meet in the restaurant*
(d) *? Bad weather persuaded Mary to meet in the restaurant*

It appears that some control verbs, such as *agree*, *prefer* and others, allow underspecification of the understood subject. Landau (2000) calls such a form of control a *partial control* (as opposed to *exhaustive control* in (12a)). Partial control is the reason why (12c) is grammatical, although there is no apparent person that Mary will meet with. If selectional restrictions are used, sentence (12d) can be also considered as exhibiting the properties of partial control, although the control verb is not one of the verbs described by Landau. Partial control poses additional challenges to the meaning derivation of control sentences, but providing the appropriate analysis for partial control is outside the scope of this work.

LFG commonly classifies control verbs as *equi* (the controller has a thematic role in the matrix sentence) and *raising* (the controller does not have a thematic role). This distinction and the question of functional vs. anaphoric control analysis are extensively discussed in the LFG literature (Kaplan and Bresnan, 1982; Bresnan, 1982, 2001; Falk, 2001) and recent work (Asudeh, to appear).

The c-structure of control sentences can be rather complex. Our work only focuses on the level of the functional and semantic structures of control sentences and therefore we assume the simplest treatment of control at the c-structure level. In particular, following Falk (2001), this work uses the informal notation of $\overline{\mathrm{VP}}$ to describe the category of the infinitival *to* constructions.

A comprehensive survey of different control types can be found in Jackendoff and Culicover (2003) and Engh and Kristoffersen (1996) list many other resources. Control has received significant attention in other linguistic theories as well (see Chomsky (1981) for treatment in Government and Binding and Sag and Pollard (1991), later revised in Pollard and Sag (1994, Ch.7) for treatment in HPSG).

2.3 The semantics of control

2.3.1 Propositional and property theories of control

Two major theories provide different semantic analyses for control verbs. One approach (Chierchia, 1984; Dowty, 1985) argues that the control verb's complement denotes a property, and therefore is called the *property approach*. According to this approach, the meaning of (13) is (14):

(13) *John tried to yawn*

[6] Example (11) is example (114) from Jackendoff and Culicover (2003).

(14) $\mathsf{try}(\mathsf{john}, \lambda x.\mathsf{yawn}(x))$

This analysis allows correct inference patterns as described, among other sources, in Asudeh (2002). The second approach is the *propositional approach* (Sag and Pollard, 1991; Pollard and Sag, 1994) that argues that the verb's complement is a proposition, and this approach is also adopted by Dalrymple (2001). In the propositional theory of control the meaning of (13) is

(15) $\mathsf{try}(\mathsf{john}, \mathsf{yawn}(\mathsf{john}))$

Our work was performed in the context of the propositional approach to control.

2.3.2 Current Glue analysis of control

Resource management issues arise when a functional control analysis of equi verbs is combined with the propositional theory of control. Both Dalrymple (2001) and Asudeh (to appear) propose (for different languages) a high order meaning constructor to derive the correct semantics. The lexical entry of a control verb *try* looks like:

(16) *try* V (↑ PRED) = 'TRY<SUBJ,XCOMP>'
(↑ XCOMP SUBJ)=(↑ SUBJ)
$\lambda P.\lambda x.\mathsf{try}(x, P(x))$: $((\uparrow \text{XCOMP SUBJ})_\sigma \multimap (\uparrow \text{XCOMP})_\sigma) \multimap (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$

This approach provides a flat semantic derivation, in which the control verb is responsible for explicitly substituting the meaning of the understood subject into the semantic predicate.

In section 5 we propose a more structured analysis, in which the understood subject is specified once only at the f-structure level, through f-structure sharing.

2.4 Plural entities

Plural entities are required to describe collective and cumulative quantification (Scha, 1981) and mass terms (Link, 1983). Plural entities are also needed to represent the meaning of simple sentences like (17)

(17) *John and Mary met.*

which cannot be represented by

(18) * $\mathsf{meet}(\mathsf{john}) \wedge \mathsf{meet}(\mathsf{mary})$

Scha (1981) suggests that entities that represent more than one element should be represented as a set of elements. Each element is of semantic type e, and therefore a set of elements would be of type (e, t). Verb predicates can then operate on atomic elements of type e or on sets of atomic elements of type (e, t). While proposing a solution to the problem of collective and cumulative predication, it requires a model in which the semantic type of plural entities is the same as of nouns and intransitive verbs, which in turn must be raised in order to operate on the plural entities.

Link (1983) proposes a semi-lattice structure to represent the elements of the model. In this approach all elements are of type e, but while some elements represent single entities (like the person John), other represent plural entities (like the two persons, John and Mary). The main advantage of this work highlighted by Link is the ability to correctly represent mass terms (e.g., water) and its ability to correctly describe the "part-of" relation (e.g., the diamond is part of the ring). Other approaches exist apart from these two and Schwarzschild (1996); Landman (2004) as well as other sources provide a comprehensive discussion on the various approaches to the representation of plural entities.

Our account is indifferent to the theory of plurality used. Link's notation is used and no distinction between singular and plural entities is made (both are of the same semantic type 'e'). This allows us to keep the usual type system in which entities (whether they are plural or mass entities) are of the same semantic type (e) and nouns and intransitive verbs are of type (e, t). Additionally, in our approach as it is presented in section 3, there is no need for an atom/set distinction, discussed in Winter (2001, Ch. 5.3), because the validation of the semantic number is performed at the f-structure level, and at the semantic level the lattice entities are all of the same kind and may be either plural or singular.

Disregarding the preferred choice of plural entry representation it is convenient to organize plural entities in a semi-lattice. The $\perp$ entity (corresponding to the empty set if sets are used) and all singular entries are the atoms of the semi-lattice. Plural entities are created using the least upper bound operator of the semi-lattice that is represented by the $\oplus$ symbol. For example the plural entity that represents both John and Mary is $\mathsf{john} \oplus \mathsf{mary}$. In section 7 the $\oplus$ operator is used to derive the plural semantics of multi-functional control sentences.

3 The semantic number

In section 1 we have briefly mentioned that the controlled verb may introduce restrictions on the understood subject. One such restriction is the semantic number restriction (compare the two sentences in (3)). These restrictions are unique in the fact that they cross the boundaries of the controlled sentence and influence directly the matrix sentence. In this section we show that functional control provides a correct analysis of the phenomena. In section 5 we present the multi-functional control approach that provides an analysis of split control sentences that also exhibit the semantic number restriction.

3.1 Linguistic data

The semantic number is a distinct property of noun phrases. For example, a semantic number mismatch is responsible for the ungrammaticality of (19b) and (19c):

(19) (a) *The boys gathered in the old building*
(b) * *Bob met in the park*
(c) * *The girl gathered elsewhere*

All four combinations of syntactic and semantic number values are possible. Syntactic number agreement is still required, as shown in (20b):[7]

(20) (a) *The committee gathers this afternoon*
(b) * *The committee gather this afternoon*
(c) * *The eyeglasses are similar*

The subject of (20a) is syntactically singular, and it agrees with the verb, the form of which is 3$^{\text{rd}}$ person, singular. However it is also semantically plural, which is exactly what the verb *gathers* requires as a subject. On the other hand (20b) is ungrammatical because although the noun *committee* is semantically plural, it is syntactically singular, and therefore is does not agree with the verb *gather*. To complete the picture, some verb phrases require a semantically singular subject. Consider for example the difference between the next two sentences:

(21) (a) *The boy is a writer*
(b) * *The committee is a writer*

As mentioned above, the semantic number restriction crosses the boundaries of the controlled sentence. Consider for example the differences between (a) and (b) in:

(22) (a) *The boys planned to gather elsewhere*
(b) * *The girl planned to gather elsewhere*

(23) (a) *The committee seems to meet in the conference room*
(b) * *Bob seems to meet in the conference room*

Despite its name, the semantic number is a syntactic property and no knowledge of the actual number of the committee members is needed in order to determine that (20a) is grammatical. Similar syntactic property was also proposed by Wechsler and Zlatic (2003); Heycock and Zamparelli (to appear).

3.2 The "SEMNUM" feature and functional control

It is possible to model the observations of the previous section by introducing a new feature in the f-structure, the SEMNUM feature. This new feature describes the semantic number of nouns and noun phrases and has two possible values — "SG" for semantically singular nouns, and "PL" for semantically plural nouns. For example, the noun *committee* that is considered syntactically singular and semantically plural will have (NUM: SG) and (SEMNUM: PL), while the word *eyeglasses* which is considered syntactically plural and semantically singular will have (NUM: PL) and (SEMNUM: SG). Heycock and Zamparelli (to appear) propose a similar distinction between syntactic ad semantic number for nouns and noun phrases. The paper proposes two boolean properties: LATT that describes semantic and PLUR that denotes syntactic plurality.

Only a few verbs like the intransitive *meet* and *gather* pose semantical number restrictions on their subject. Therefore, only these verbs require the additional functional equation constraining the semantic number of the subject, and the rest of the verbs in the lexicon remain unchanged. Additionally, as it has been noted in the previous section, the SEMNUM feature is not part of the usual agreement restrictions between the subject and the verb in a sentence.

[7] Example (20c) is inspired by Winter 2001, p. 192.

The functional control analysis accounts for the semantic restriction posed on the understood subject by a controlled verb. Functional control ensures that the local restrictions posed by the controlled verb on its understood subject propagate through f-structure sharing to the controlling grammatical function in the matrix sentence. Consider for example the following control sentence:

(24) *The committee plans to gather (in the afternoon)*

The appropriate lexical entries for the controlled infinitive verb *to gather* and the controlling verb *plans* are presented in (25).[8] Figure 5 presents the full f-structure and the s-structure of the sentence.[9]

(25) *to gather*^{it} $\overline{\text{VP}}$ $(\uparrow$ PRED$)$ = 'GATHER<SUBJ>'
$(\uparrow$ SUBJ SEMNUM$)$ = PL

plans V $(\uparrow$ PRED$)$ = 'PLAN<XCOMP,SUBJ>'
$(\uparrow$ SUBJ NUM$)$ = SG
$(\uparrow$ XCOMP SUBJ$)$ = $(\uparrow$ SUBJ$)$

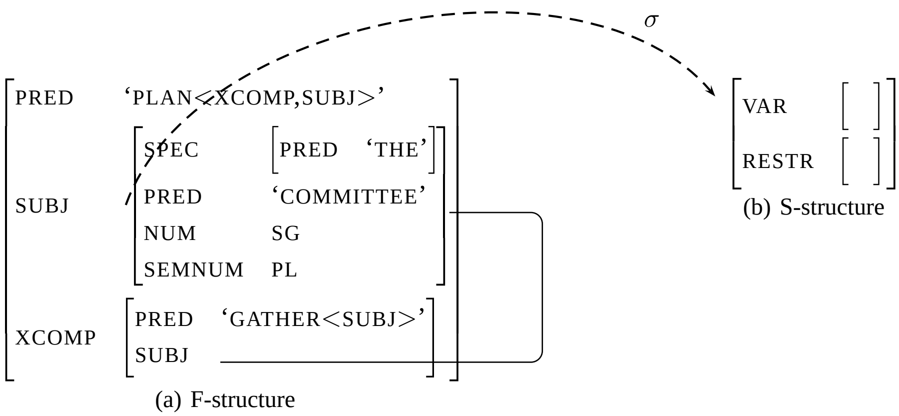

Figure 5: The f-structure and the s-structure of *The committee plans to gather*

Since the semantic number is in fact a syntactic feature, the meaning constructors of the affected nouns and verbs don't have to change. Lexical entries of representative nouns and a verb appear in (26). Notice that the premises on the linear logic side of the meaning constructor involve internal elements of the s-structures, which themselves are σ-projections of the corresponding f-structures. These meaning constructors are described in detail in Dalrymple (2001, p. 251).

(26) *committee* N $(\uparrow$ PRED$)$ = 'COMMITTEE'
$(\uparrow$ NUM$)$ = SG
$(\uparrow$ SEMNUM$)$ = PL
$\lambda x.\mathsf{committee}(x) : (\uparrow_\sigma \text{ VAR}) \multimap (\uparrow_\sigma \text{ RESTR})$

eyeglasses N $(\uparrow$ PRED$)$ = 'EYEGLASSES'
$(\uparrow$ NUM$)$ = PL
$(\uparrow$ SEMNUM$)$ = SG
$\lambda x.\mathsf{eyeglasses}(x) : (\uparrow_\sigma \text{ VAR}) \multimap (\uparrow_\sigma \text{ RESTR})$

met V $(\uparrow$ PRED$)$ = 'MEET<SUBJ>'
$(\uparrow$ SUBJ SEMNUM$)$ = PL
$\lambda x.\mathsf{meet}(x) : (\uparrow \text{ SUBJ})_\sigma \multimap \uparrow_\sigma$

meets V $(\uparrow$ PRED$)$ = 'MEET<SUBJ>'
$(\uparrow$ SUBJ NUM$)$ = SG
$(\uparrow$ SUBJ SEMNUM$)$ = PL
$\lambda x.\mathsf{meet}(x) : (\uparrow \text{ SUBJ})_\sigma \multimap \uparrow_\sigma$

[8] See section 4 for the description of the $\overline{\text{VP}}$ notation.
[9] Figure 5 only shows the outline of the s-structure. In-depth discussion can be found in Dalrymple (2001).

The analysis presented in this section has yet to be extended to account for determiner agreement (Dalrymple and King, 2004) and quantification in order to correctly analyze sentences such as *All committees met* and *Every committee met* (Winter, 2001, p. 202, ex. 34).

4 The c-structure rules

This section presents the main c-structure rules that will be used in the rest of this work. First there is the basic sentence rule, the noun phrase creation rule (Dalrymple, 2001, pg. 156) and the noun phrase coordination rule that will be used in section 6.1.

(27)
$$\begin{array}{lll} \mathrm{S} & \longrightarrow & \underset{(\uparrow \textsc{subj})=\downarrow}{\mathrm{NP}} \quad \underset{\uparrow=\downarrow}{\mathrm{VP}} \\[2ex] \mathrm{NP} & \longrightarrow & \underset{(\uparrow \textsc{spec})=\downarrow}{\mathrm{Det}} \quad \underset{\uparrow=\downarrow}{\mathrm{N}} \\[2ex] \mathrm{NP} & \longrightarrow & \underset{\downarrow \in \uparrow}{\mathrm{NP}} \quad \underset{\uparrow=\downarrow}{\mathrm{Cnj}} \quad \underset{\downarrow \in \uparrow}{\mathrm{NP}} \end{array}$$

The last rule that should be considered is the VP rule. Following Falk (2001) we use the informal notation of $\overline{\mathrm{VP}}$ to describe the category of the *to* constructions.[10] The rules that handle the most relevant VP complements are:

(28)
$$\begin{array}{lll} \mathrm{VP} & \longrightarrow & \underset{\uparrow=\downarrow}{\mathrm{V}} \left(\underset{(\uparrow \textsc{obj})=\downarrow}{\mathrm{NP}} \right) \left(\underset{(\uparrow \textsc{comp}|\textsc{xcomp})=\downarrow}{\overline{\mathrm{VP}}} \right) \underset{\downarrow \in (\uparrow \textsc{adj})}{\mathrm{PP}^*} \\[2ex] \overline{\mathrm{VP}} & \longrightarrow & \textit{to} \quad \underset{\uparrow=\downarrow}{\mathrm{VP}} \end{array}$$

5 Multi-functional control

Consider the plural reading of sentence (1), repeated here for convenience

(29) *John persuaded Mary to meet in the restaurant*

In this plural reading that talks about John and Mary meeting together in the restaurant, the understood subject of the controlled sentence is no longer controlled by a single grammatical function of the matrix sentence. Instead, it is controlled jointly by two grammatical functions of the matrix sentence, its subject and its object.

We propose to describe this phenomenon by representing the understood subject of the controlled sentence by an f-structure that represents both contributors. The f-structure representing the understood subject becomes a set, the elements of which are the matrix subject and object that appear inside the understood subject f-structure via the f-structure sharing mechanism (figure 6). This way, the f-structure of the understood subject can have properties of its own, as is the case with coordination. For example, the syntactic and the semantic number of the understood subject may differ from the corresponding number of its constituents.

The f-structure that represents a multi-functional control analysis of (29) is presented in figure 6. In the f-structure the number properties of the understood subject correctly describe the syntactic and the semantic number of the understood subject.

The multi-functional control structure is created by the functional equations in the control verb's lexical entry. The lexical entry is also responsible for assigning the number feature.[11] We propose the following lexical entry for the verb *persuaded*, that is responsible for creating the f-structure in figure 6:

(30) *persuaded* V
$$\begin{array}{l} (\uparrow \textsc{pred}) = \text{‘}\textsc{persuade}\langle\textsc{subj,obj,xcomp}\rangle\text{’} \\ (\uparrow \textsc{subj}) \in (\uparrow \textsc{xcomp subj}) \\ (\uparrow \textsc{obj}) \in (\uparrow \textsc{xcomp subj}) \\ (\uparrow \textsc{xcomp subj num}) = \textsc{pl} \\ (\uparrow \textsc{xcomp subj semnum}) = \textsc{pl} \end{array}$$

[10] Refer to Falk (in preparation) for a discussion on the categorization of the infinitival *to*.

[11] An approach that would solve the problem of non-distributive features in noun phrase coordination can be applied to compute the number features of the understood subject. To simplify the analysis multi-functional control assigns plural semantic number to the understood subject.

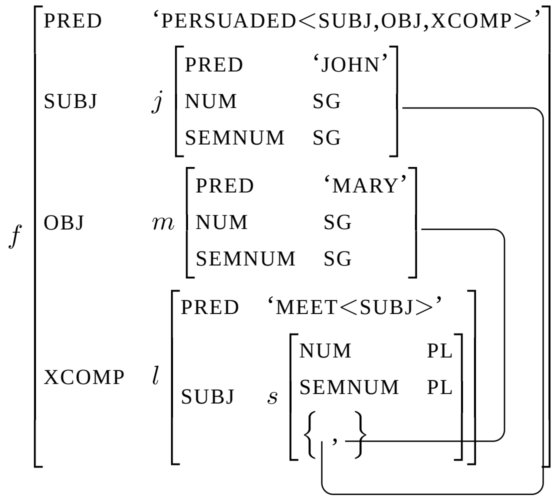

Figure 6: A multi-functional control analysis of *John persuaded Mary to meet*

The proposed lexical entry for the verb *persuaded* does not replace the existing conventional functional analysis that is suitable for sentences that exhibit simple subject or object control, such as (10b), repeated here for convenience as (31):

(31) *John persuaded Mary to become a writer*

The verb *persuaded* has two lexical entries — the multi-functional entry and the conventional entry. These lexical entries share the common properties of the verb and only the functional equations that define the control features differ between the two.[12] When the LFG analysis is performed both lexical entries are considered. If the semantic number restriction permits conventional functional control it will be used (for example, it is possible for (31), but not for (29)). Additionally, a multi-functional control analysis is attempted that succeeds for (29) but fails for (31), because the multi-functional understood subject disagrees in its semantic number with the verb phrase *to become a writer*:

(32) *become a writer* VP ($\uparrow$ PRED) = ‘BECOME-A-WRITER<SUBJ>’
($\uparrow$ SUBJ SEMNUM) = SG

For sentences such as (2), both analyses will succeed, and the result is two possible f-structures that describe the sentence. As a direction for future research, it should be possible to generalize the multi-functional analysis to correctly analyze conventional functional control sentences as having a single element in the understood subject set, assuming the ability to compute the semantic number of such a set.

6 Plural meaning of sets

In this section we present our approach to meaning derivation of sets. In this paper we are mostly interested with the application to multi-functional control, but the most immediate application is probably to coordination. While our approach is formulated in such a way that it can suit other types of coordination as well, we focus on the issue of noun-phrase coordination, since it is most close to the problem of deriving the meaning of multi-functional control sentences.

6.1 The case of noun-phrase coordination

6.1.1 Current analysis

Consider for example the following sentence that involves noun phrase coordination:

(33) *John, Philip, Mary and Susan walked.*

The corresponding f-structure (based on Dalrymple 2001) is shown in figure 7.

Now that an f-structure has been created, the semantic meaning can be derived using the Glue framework. We would like to build a meaning constructor that consumes all set elements’ semantic projections and produces the semantic projection of the subject, with the appropriate meaning achieved through the Curry-Howard isomorphism. However, the number of set

[12] Dalrymple et al. (2004) propose notational mechanism that will allow sharing of the common properties between the two alternative entries.

$$f\begin{bmatrix} \text{PRED} & \text{`WALK<SUBJ>'} \\ \text{SUBJ} & g\begin{bmatrix} \text{CONJ} & \text{AND} \\ & \left\{\begin{matrix} j\begin{bmatrix}\text{PRED} & \text{`JOHN'}\end{bmatrix} \\ p\begin{bmatrix}\text{PRED} & \text{`PHILIP'}\end{bmatrix} \\ m\begin{bmatrix}\text{PRED} & \text{`MARY'}\end{bmatrix} \\ s\begin{bmatrix}\text{PRED} & \text{`SUSAN'}\end{bmatrix} \end{matrix}\right\} \end{bmatrix} \end{bmatrix}$$

Figure 7: The f-structure of *John, Philip, Mary and Susan walked*

elements is unknown at the time the lexicon is designed and meaning constructors that appear in the lexicon cannot deal with a variable number of Glue premises.

Some techniques have been proposed to cope with this problem. Dalrymple (2001, Ch. 13) proposes to attach two meaning constructors to the lexical entry of *and*. The first (**[g-and]**) is responsible for picking two arbitrary elements from the set and computing their conjunction. The second meaning constructor (**[g-and2]**) is responsible for consuming the rest of the set elements and producing the final meaning of the subject. This meaning constructor (**[g-and2]**) is defined with the '!' linear logic operator that suspends resource sensitivity limitations of this meaning constructor and allows to use it an unlimited number of times. This is a major drawback of this approach, since it complicates the Glue derivation, while the actual number of uses is known, and it is the number of set elements. Another approach proposed by Asudeh and Crouch (2002a) attempts to solve the problem by presenting meaning constructors in c-structure rules, and allow c-structure operators, such as the Kleene star to duplicate them as well. A drawback of this approach is having (complex) meaning constructors in c-structures, while most meaning constructors are provided by the lexicon.

6.1.2 Our analysis

We propose a different approach that handles both the noun-phrase coordination and multi-functional control (section 5). We take advantage of the fact that the semantic meaning derivation can be done in three consecutive steps. First, the c-structure and the f-structure are created. Next, the semantic projections are computed and meaning constructors are instantiated. As a last step, Glue logic rules are applied to the meaning constructors.

While the first and the last steps are performed in the frameworks external to Glue (LFG itself and linear logic), the second step can be extended to enrich the expressive power of the meaning constructors.

We propose to allow the creation of meaning constructors using constructs similar to templates or macros[13] in programming languages. Such a template, which we call a *meta-meaning constructor* would allow us to create a meaning constructor, based on the knowledge of the whole f-structure, as opposed to specifying the meaning constructors in the lexicon, where the f-structure context information is, of course, not available. Specifically, when the meta-meaning constructor is instantiated, the number of set elements is already known, while information of that kind is usually not available at the time the lexicon is created. After instantiation, the meta-meaning constructor becomes a regular meaning constructor that is used in derivation as usual.

The meta-meaning constructor for noun phrase coordination of some set S is defined as:

$$Coord(S) = \lambda x_1 . \dots . \lambda x_n . x_1 \oplus \dots \oplus x_n : (S \in)_{\sigma_e} \multimap \dots \multimap (S \in)_{\sigma_e} \multimap S_{\sigma e}$$

The set S is the parameter of the noun phrase coordination meta-meaning constructor. When the meta-meaning constructor appears in the lexical entry of *and*, the mother node meta-variable appears as the parameter. After the f-structure has been instantiated the meta-variable is substituted by the actual f-structure set. The lexical entry of the noun-phrase coordinating *and* is:

(34)	*and*$^{\text{NP}}$	Cnj	($\uparrow$ CONJ) = AND
			$Coord(\uparrow)$

[13] Asudeh and Crouch (2002a) mentions the possible application of *macros* to provide linguistic generalizations, but no details are provided.

After the f-structure has been created (see figure 7) the actual f-structure set g is substituted as the parameter of the meaning constructor. The result, that reflects the number of g's set elements is

$$Coord(g) = \lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.x_1 \oplus x_2 \oplus x_3 \oplus x_4 : (S \in)_{\sigma e} \multimap (S \in)_{\sigma e} \multimap (S \in)_{\sigma e} \multimap (S \in)_{\sigma e} \multimap S\ _e$$
$$Coord(g) = \lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.x_1 \oplus x_2 \oplus x_3 \oplus x_4 : j_e \multimap p_e \multimap m_e \multimap s_e \multimap g_e$$

It should be mentioned that if the set g consisted of only two f-structures, the result would be the same meaning constructor as in the lexical entry of *and*, proposed by (Dalrymple, 2001).

6.1.3 Coordinating quantified nouns phrases

With meta-meaning constructors, quantified noun phrases are analyzed in same way as before. Consider, for example, the following sentence:

(35) *John, Mary, a professor and Susan walked.*

The corresponding f-structure is presented in figure 8:

$$f \begin{bmatrix} \text{PRED} & \text{`WALK<SUBJ>'} \\ \text{SUBJ} & g \begin{bmatrix} \text{CONJ} & \text{AND} \\ \multicolumn{2}{l}{\left\{ \begin{array}{l} j \begin{bmatrix} \text{PRED} & \text{`JOHN'} \end{bmatrix} \\ m \begin{bmatrix} \text{PRED} & \text{`MARY'} \end{bmatrix} \\ p \begin{bmatrix} \text{PRED} & \text{`PROFESSOR'} \\ \text{SPEC} & \text{`A'} \end{bmatrix} \\ s \begin{bmatrix} \text{PRED} & \text{`SUSAN'} \end{bmatrix} \end{array} \right\}} \end{bmatrix} \end{bmatrix}$$

Figure 8: The f-structure of *John, Mary, a professor and Susan walked*

In order to instantiate the meaning constructor for *and* the $Coord$ meta-meaning constructor is applied on the f-structure set g. The result is the meaning constructor **[and]** that appears in (36)[14] along with the complete list of all instantiated meaning constructors.

(36)
$$\begin{array}{llr@{\;:\;}l} & \textbf{[John]} & \mathsf{john} & j_\sigma \\ & \textbf{[Mary]} & \mathsf{mary} & m_\sigma \\ & \textbf{[a professor]} & \lambda Q.\mathsf{a}(x, \mathsf{professor}(x), Q(x)) & \forall H.[p_\sigma \multimap H] \multimap H \\ & \textbf{[Susan]} & \mathsf{susan} & s_\sigma \\ & \textbf{[and]} & \lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.x_1 \oplus x_2 \oplus x_3 \oplus x_4 & m_\sigma \multimap p_\sigma \multimap j_\sigma \multimap s_\sigma \multimap g_\sigma \\ & \textbf{[walked]} & \lambda x.\mathsf{walked}(x) & g_\sigma \multimap f_\sigma \end{array}$$

Figure 9 shows the complete meaning derivation of (35). Note that if more than one quantified noun phrase occurred in the sentence, the semantic analysis would become naturally ambiguous.

6.2 Set coordination for other phrase types

The approach introduced in section 6.1 can be applied to sentence level (S) coordination with the only differences being the semantic type (which is now t), the coordination operator (propositional $\wedge$) and the f-structure label. Consider sentence (37) with figure 10 showing the appropriate f-structure.

(37) *John smiled, Mary laughed and Susan giggled.*

In order to accommodate additional semantic types and coordination operators, we generalize the coordination meta-meaning constructor. The new constructor is generic with respect to three parameters: the f-structure set being coordinated, the semantic coordination operator and the semantic types of the set elements' semantic projections. The generic set coordination

[14]We do not analyze the internal structure of the quantified noun phrase. A detailed analysis can be found in Dalrymple (2001).

$$
\dfrac{\dfrac{\textbf{[a professor]}}{\lambda Q.\mathsf{a}(x, \mathsf{professor}(x), Q(x)) : \forall H.[p_\sigma \multimap H] \multimap H} \quad \dfrac{\dfrac{\dfrac{\textbf{[walked]}}{\lambda x.\mathsf{walked}(x) : g_\sigma \multimap f_\sigma} \quad \dfrac{\dfrac{\textbf{[Susan]}}{\mathsf{susan} : s_\sigma} \quad \dfrac{\dfrac{\textbf{[John]}}{\mathsf{john} : j_\sigma} \quad \dfrac{z : [p_\sigma] \quad \dfrac{\dfrac{\textbf{[Mary]}}{\mathsf{mary} : m_\sigma} \quad \dfrac{\textbf{[and]}}{\lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.x_1 \oplus x_2 \oplus x_3 \oplus x_4 : m_\sigma \multimap p_\sigma \multimap j_\sigma \multimap s_\sigma \multimap g_\sigma}}{\lambda x_2.\lambda x_3.\lambda x_4.\mathsf{mary} \oplus x_2 \oplus x_3 \oplus x_4 : p_\sigma \multimap j_\sigma \multimap s_\sigma \multimap g_\sigma}}{\lambda x_3.\lambda x_4.\mathsf{mary} \oplus z \oplus x_3 \oplus x_4 : j_\sigma \multimap s_\sigma \multimap g_\sigma}}{\lambda x_4.\mathsf{mary} \oplus z \oplus \mathsf{john} \oplus x_4 : s_\sigma \multimap g_\sigma}}{\mathsf{mary} \oplus z \oplus \mathsf{john} \oplus \mathsf{susan} : g_\sigma}}{\mathsf{walked}(\mathsf{mary} \oplus z \oplus \mathsf{john} \oplus \mathsf{susan}) : f_\sigma}}{\lambda z.\mathsf{walked}(\mathsf{mary} \oplus z \oplus \mathsf{john} \oplus \mathsf{susan}) : p_\sigma \multimap f_\sigma}\ (I)}{\mathsf{a}(x, \mathsf{professor}(x), \mathsf{walked}(\mathsf{mary} \oplus x \oplus \mathsf{john} \oplus \mathsf{susan})) : f_\sigma}\ (H \Rightarrow f_\sigma, \multimap_{\mathcal{E}})
$$

Figure 9: The semantic derivation of *John, Mary, a professor and Susan walked*

$$f\left[\begin{array}{l}\text{CONJ}\quad\text{AND}\\ \left\{\begin{array}{l} s\left[\begin{array}{ll}\text{PRED} & \text{‘SMILE<SUBJ>’}\\ \text{SUBJ} & \left[\text{PRED}\quad\text{‘JOHN’}\right]\end{array}\right]\\ l\left[\begin{array}{ll}\text{PRED} & \text{‘LAUGH<SUBJ>’}\\ \text{SUBJ} & \left[\text{PRED}\quad\text{‘MARY’}\right]\end{array}\right]\\ g\left[\begin{array}{ll}\text{PRED} & \text{‘GIGGLE<SUBJ>’}\\ \text{SUBJ} & \left[\text{PRED}\quad\text{‘SUSAN’}\right]\end{array}\right]\end{array}\right\}\end{array}\right]$$

Figure 10: The f-structure of *John smiled, Mary laughed and Susan giggled.*

meta-meaning constructor for some set f, elements of which have semantic projection type τ that uses the ‘$\odot$’ coordination operator is defined as:

$$Coord(f, \odot, \tau) = \lambda x_1. \ldots .\lambda x_n.x_1 \odot \ldots \odot x_n : (f\in)_{\sigma_\tau} \multimap \ldots \multimap (f\in)_{\sigma_\tau} \multimap f_{\sigma_\tau}.$$

For sentence (37) the meta-meaning constructor is instantiated with set f, semantic type t and semantic coordination operator $\wedge_t$ and can be now used in the meaning derivation:

$$\lambda x_1.\lambda x_2.\lambda x_3.x_1 \wedge x_2 \wedge x_3 : s_{\sigma_t} \multimap (l_{\sigma_t} \multimap (g_{\sigma_t} \multimap f_{\sigma_t}))$$

While meta-meaning constructors may not be the ultimate answer to the issue of coordination in natural languages, they can help the lexicon designer to better organize the (structured) lexicon, and assist in the meaning derivation of control sentences.

7 The semantics of multi-functional control

7.1 The semantics of the understood subject

In order to derive the meaning of the understood subject of the controlled sentence in (29) (the f-structure appears in figure 6) we use the $Coord$ meta-meaning constructor introduced in section 6.2. In order to use the meta-meaning constructor we add the following element to the multi-functional lexical entry of *persuade* in (30):

(38) $Coord((\uparrow \text{XCOMP SUBJ}), \oplus_e, e)$

After instantiation (recall that j and m are elements of s), it becomes:

(39) **[coord]** $\lambda x_1.\lambda x_2.x_1 \oplus x_2 : j_\sigma \multimap (m_\sigma \multimap s_\sigma)$

When the coordination meaning constructor is combined with the meaning constructors of *John* and *Mary*

(40) **[John]** $\text{john} : j_\sigma$
[Mary] $\text{mary} : m_\sigma$

the meaning of the understood subject can be derived, as shown in figure 11.

$$\dfrac{\dfrac{\textbf{[Mary]}}{\text{mary} : m_\sigma} \qquad \dfrac{\dfrac{\textbf{[John]}}{\text{john} : j_\sigma} \quad \dfrac{\textbf{[coord]}}{\lambda x_1.\lambda x_2.x_1 \oplus x_2 : j_\sigma \multimap (m_\sigma \multimap s_\sigma)}}{\lambda x_2.\text{john} \oplus x_2 : m_\sigma \multimap s_\sigma}}{\text{john} \oplus \text{mary} : s_\sigma}$$

Figure 11: The derivation of the understood subject meaning

The meaning derivation of the understood subject is only the first step in the meaning derivation process. In the next steps the meaning of the understood subject is used to derive the meaning of the controlled sentence, which in turn is used to derive the meaning of the whole sentence. The complete derivation appears in figure 15.

7.2 Multi-functional control and resource sensitivity

7.2.1 Control sentences and resource deficit

The meaning derivation of the understood subject, presented in the previous section, has consumed the resources that represented the meanings of the controllers *John* and *Mary*. Since a resource that was consumed cannot be used again for the derivation of the meaning of the whole sentence, the derivation of the whole sentence will fail. Such a situation is called a *resource deficit*.

Two approaches have been proposed to deal with such a resource scarcity. The first is the "paths as resources" approach (Kehler et al., 1999). In this approach it is possible for a verb to consume a path that leads to the semantic projection, and not the projection itself. This way, shared resources can be consumed the number of times they are shared in the f-structure, and solve the resource deficit problem. However this approach appears to fail when it comes to some control verbs. For example, the f-structure (Dalrymple, 2001; Asudeh, to appear) for a sentence like

(41) *David seems to leave*

shares the subject of the main sentence with the subject of the controlled sentence. However, the semantic projection of the shared subject is consumed only once, by the controlled verb. If the "paths as resources" approach would be applied here, there would be an unused resource which would cause the whole derivation to fail.

When it comes to equi, similar problem exists in property theory of control.[15] The subject of the main sentence is shared with the subject of the controlled sentence, and according to this theory two resources are introduced. Yet, the semantic subject of the controlled sentence is never consumed in the property theory, which again, leaves an unused resource.

The second approach, proposed by Asudeh (2002, to appear) is to "get in charge" of the problem, by using high order types. This approach claims that if a control verb is responsible for establishing the sharing via its functional equations, it should be responsible for dealing with it in its meaning constructor. That is, the control verb's meaning constructor expects to receive all its grammatical functions that contribute to its meaning. The meaning side of the meaning constructor will take care of substituting the right meaning into the understood subject's position. According to this approach, the meaning constructor of the multi-functional version of *persuaded*, would look like

$$\lambda x.\lambda.y.\lambda P.\mathsf{persuade}(x, y, P(x \oplus y)) : ((\uparrow \text{XCOMP SUBJ})_\sigma \multimap (\uparrow \text{XCOMP})_\sigma) \multimap ((\uparrow \text{SUBJ})_\sigma \multimap ((\uparrow \text{OBJ})_\sigma \multimap \ \uparrow_\sigma))$$

The disadvantage of this approach is that it proposes a flat analysis of the sentence that ignores the complex f-structure almost completely. For example, a meaning constructor, the right side of which corresponds to the understood subject $(\uparrow \text{XCOMP SUBJ})_\sigma$ does not occur in the derivation.

7.2.2 Resource management

In order to handle these resource sensitivity issues we step outside the implicational fragment of the linear logic. In this section we propose the means to produce additional copies of Glue premises in a controlled manner. The resource management rules which are based on the rules in Asudeh (2004), use the linear conjunction '$\otimes$' operator, with the corresponding use of the '$\times$' operator on the meaning side. For example, the following meaning constructor, when added to the lexical entry of *persuaded* will create another copy of the matrix subject:

(42) $\lambda a.a \times a : (\uparrow \text{SUBJ})_\sigma \multimap ((\uparrow \text{SUBJ})_\sigma \otimes (\uparrow \text{SUBJ})_\sigma)$

This way, the matrix subject can contribute to the meaning of the understood subject of the controlled sentence and still be used as the subject of the matrix verb. The instantiated version of this meaning constructor for sentence (29), based on the f-structure in figure 6, is presented below along with the meaning constructor of the subject. Figure 12 presents the derivation of the duplicated subject meaning.

(43) **[John]** $\mathsf{john} : j_\sigma$
[subjdup] $\lambda a.a \times a : j_\sigma \multimap (j_\sigma \otimes j_\sigma)$

$$\dfrac{\dfrac{\textbf{[john]}}{\mathsf{john} : j_\sigma} \qquad \dfrac{\textbf{[subjdup]}}{\lambda a.a \times a : j_\sigma \multimap (j_\sigma \otimes j_\sigma)}}{\mathsf{john} \times \mathsf{john} : j_\sigma \otimes j_\sigma}$$

Figure 12: The duplication of the matrix subject meaning constructor

The newly created meaning constructor cannot be simply split into two meaning constructors. The conjunction elimination rule (Asudeh, 2004) shown in figure 13 must be used.[16] The result can be β-reduced using the rule in (44).

[15] We mention it here although the work focuses on the propositional theory of control.
[16] All linear logic rules used in this work appear in Appendix A.

$$\dfrac{\begin{array}{cc} & [x:A]^1\ [y:B]^2 \\ \vdots & \vdots \\ a : A \otimes B & f : C \end{array}}{\text{let } a \text{ be } x \times y \text{ in } f : C} \otimes_{\varepsilon,1,2}$$

Figure 13: The linear conjunction elimination rule

(44) let $a \times b$ be $x \times y$ in $f \quad \Rightarrow_\beta \quad f[a/x, b/y]$

In the resource management approach we use the fact that systematic mismatches between representation levels are possible in Glue. Hence, resource duplication is induced by the f-structure sharing rules, but is not mandatory. This allows a fine grained control over resource duplication, as opposed to the "paths as resources" approach of Kehler et al. (1999) discussed earlier.

In order to derive the meaning of (29), both f-structure sharings must be realized with resource duplication constructors. Therefore another meaning constructor, similar to (42) that duplicates the object's meaning is required in the lexical entry of *persuaded*. The full derivation of a multi-functional control sentence is presented in the next section.

7.3 Complete derivation

In this section we present the complete meaning derivation of the sentence *John persuaded Mary to meet (in the restaurant)*. For this sentence, the plural reading (i.e., both John and Mary will be meeting) is the only possible reading. The derivation here is presented from the basic lexical entries, up to the full propositional meaning derivation.

7.3.1 The lexical entries

These are the lexical entries that belong to the above sentence:

John	NP	$(\uparrow \text{PRED}) = \text{'JOHN'}$ $(\uparrow \text{NUM}) = \text{SG}$ $(\uparrow \text{SEMNUM}) = \text{SG}$ $\mathsf{john} : \uparrow_\sigma$
Mary	NP	$(\uparrow \text{PRED}) = \text{'MARY'}$ $(\uparrow \text{NUM}) = \text{SG}$ $(\uparrow \text{SEMNUM}) = \text{SG}$ $\mathsf{mary} : \uparrow_\sigma$
meet[it]	V	$(\uparrow \text{PRED}) = \text{'MEET<SUBJ>'}$ $(\uparrow \text{SUBJ SEMNUM}) = \text{PL}$ $\lambda x.\mathsf{meet}(x) : (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$
persuade	V	$(\uparrow \text{PRED}) = \text{'PERSUADE<XCOMP,SUBJ,OBJ>'}$ $\left.\begin{array}{l}(\uparrow \text{XCOMP SUBJ SEMNUM}) = \text{PL} \\ (\uparrow \text{XCOMP SUBJ NUM}) = \text{PL} \\ (\uparrow \text{SUBJ}) \in (\uparrow \text{XCOMP SUBJ}) \\ (\uparrow \text{OBJ}) \in (\uparrow \text{XCOMP SUBJ})\end{array}\right\\}$ *set creating constraints* $\left.\begin{array}{l}\lambda a.a \times a : (\uparrow \text{SUBJ})_\sigma \multimap ((\uparrow \text{SUBJ})_\sigma \otimes (\uparrow \text{SUBJ})_\sigma) \\ \lambda a.a \times a : (\uparrow \text{OBJ})_\sigma \multimap ((\uparrow \text{OBJ})_\sigma \otimes (\uparrow \text{OBJ})_\sigma)\end{array}\right\\}$ *resource management* $Coord((\uparrow \text{XCOMP SUBJ}), \oplus, e)$ $\lambda P.\lambda y.\lambda x.\mathsf{persuade}(x,y,P) : (\uparrow \text{XCOMP})_\sigma \multimap [(\uparrow \text{OBJ})_\sigma \multimap [(\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma]]$

Recall that the $Coord$ meaning meta-constructor is defined by:

$$Coord(f, \odot, \tau) = \lambda x_1 \ldots \lambda x_n . x_1 \odot \ldots \odot x_n : \underbrace{(f\in)_{\sigma_\tau} \multimap \ldots \multimap (f\in)_{\sigma_\tau}}_{|f| \textit{ times}} \multimap f_{\sigma_\tau}$$

7.3.2 The c/f-structure correspondence

As the first step of the derivation the c-structure tree is built using the rules described in section 4 and the ϕ-projection is computed to create the f-structure. Figure 14 shows both structures and the ϕ-projection function.

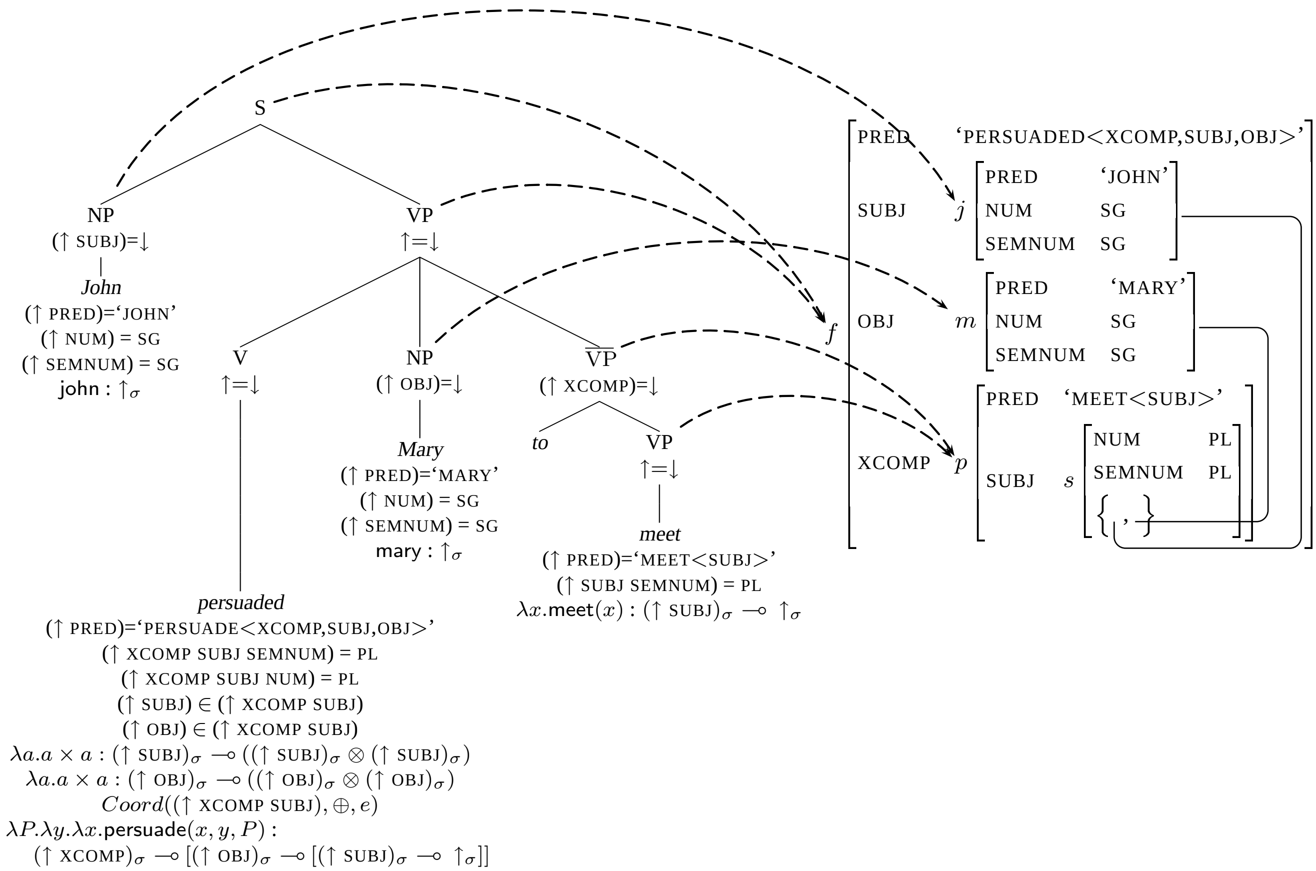

Figure 14: The c/f-structures of *John persuaded Mary to meet*

7.3.3 Instantiating the meaning constructors

After the f-structure has been created, meaning constructors are instantiated. The meaning constructors introduced by *John*, *Mary* and *meet* are instantiated in the same way they were instantiated in regular functional control, because they indeed didn't change:

(45) **[john]** $\textsf{john} : j_\sigma$
[mary] $\textsf{mary} : m_\sigma$
[meet] $\lambda x.\textsf{meet}(x) : s_\sigma \multimap p_\sigma$

The control verb *persuade* introduces several meaning constructors. First, the uninstantiated resource management meaning constructors in (46) are instantiated to become the meaning constructors in (47)

(46) $\lambda a.a \times a : (\uparrow \textsc{subj})_\sigma \multimap ((\uparrow \textsc{subj})_\sigma \otimes (\uparrow \textsc{subj})_\sigma)$
$\lambda a.a \times a : (\uparrow \textsc{obj})_\sigma \multimap ((\uparrow \textsc{obj})_\sigma \otimes (\uparrow \textsc{obj})_\sigma)$

(47) **[subjdup]** $\lambda a.a \times a : j_\sigma \multimap (j_\sigma \otimes j_\sigma)$
[objdup] $\lambda a.a \times a : m_\sigma \multimap (m_\sigma \otimes m_\sigma)$

Next, the internal coordination meaning constructor is instantiated. After substituting the actual parameters into the $Coord$ meta-meaning constructor and picking the set elements for the corresponding positions, it becomes:

(48) **[coord]** $\lambda x_1.\lambda x_2.x_1 \oplus x_2 : j_\sigma \multimap (m_\sigma \multimap s_\sigma)$

Last but not least is the meaning constructor that generates the meaning of the whole sentence. It is instantiated as usual, and the result is

(49) **[persuade]** $\lambda P.\lambda y.\lambda x.\mathsf{persuade}(x, y, P) : p_\sigma \multimap (m_\sigma \multimap (j_\sigma \multimap f_\sigma))$

To summarize this step, all 7 instantiated meaning constructors are listed again for reference:

(50)

[john]	$\mathsf{john} : j_\sigma$
[mary]	$\mathsf{mary} : m_\sigma$
[meet]	$\lambda x.\mathsf{meet}(x) : s_\sigma \multimap p_\sigma$
[subjdup]	$\lambda a.a \times a : j_\sigma \multimap (j_\sigma \otimes j_\sigma)$
[objdup]	$\lambda a.a \times a : m_\sigma \multimap (m_\sigma \otimes m_\sigma)$
[coord]	$\lambda x_1.\lambda x_2.x_1 \oplus x_2 : j_\sigma \multimap (m_\sigma \multimap s_\sigma)$
[persuade]	$\lambda P.\lambda y.\lambda x.\mathsf{persuade}(x, y, P) : p_\sigma \multimap (m_\sigma \multimap (j_\sigma \multimap f_\sigma))$

7.4 Glue derivation

After all meaning constructors have been instantiated, Glue logic rules are applied to derive the meaning of the sentence. Due to lack of space, the derivation is presented in parts that are eventually combined into the complete linear logic proof. Following that, the full derivation tree is presented in a smaller font size in figure 16.

$$\dfrac{\dfrac{[B : m_\sigma]^2 \qquad \dfrac{[A : j_\sigma]^1 \qquad \dfrac{\textbf{[coord]}}{\lambda x_1.\lambda x_2.x_1 \oplus x_2 : j_\sigma \multimap (m_\sigma \multimap s_\sigma)}}{\lambda x_2.A \oplus x_2 : m_\sigma \multimap s_\sigma}}{A \oplus B : s_\sigma} \qquad \dfrac{\textbf{[meet]}}{\lambda x.\mathsf{meet}(x) : s_\sigma \multimap p_\sigma}}{\mathsf{meet}(A \oplus B) : p_\sigma}$$

$$\dfrac{[C : j_\sigma]^3 \qquad \dfrac{[D : m_\sigma]^4 \qquad \dfrac{\mathsf{meet}(A \oplus B) : p_\sigma \qquad \dfrac{\textbf{[persuade]}}{\lambda P.\lambda y.\lambda x.\mathsf{persuade}(x, y, P) : p_\sigma \multimap (m_\sigma \multimap (j_\sigma \multimap f_\sigma))}}{\lambda y.\lambda x.\mathsf{persuade}(x, y, \mathsf{meet}(A \oplus B)) : m_\sigma \multimap (j_\sigma \multimap f_\sigma)}}{\lambda x.\mathsf{persuade}(x, D, \mathsf{meet}(A \oplus B)) : j_\sigma \multimap f_\sigma}}{\mathsf{persuade}(C, D, \mathsf{meet}(A \oplus B)) : f_\sigma}$$

$$\dfrac{\dfrac{\dfrac{\textbf{[john]}}{\mathsf{john} : j_\sigma} \qquad \dfrac{\textbf{[subjdup]}}{\lambda a.a \times a : j_\sigma \multimap (j_\sigma \otimes j_\sigma)}}{\mathsf{john} \times \mathsf{john} : j_\sigma \otimes j_\sigma} \qquad \dfrac{\dfrac{\mathsf{persuade}(C, D, \mathsf{meet}(A \oplus B)) : f_\sigma \qquad \dfrac{\dfrac{\textbf{[mary]}}{\mathsf{mary} : m_\sigma} \qquad \dfrac{\textbf{[objdup]}}{\lambda a.a \times a : m_\sigma \multimap (m_\sigma \otimes m_\sigma)}}{\mathsf{mary} \times \mathsf{mary} : m_\sigma \otimes m_\sigma}}{\mathsf{let\ mary} \times \mathsf{mary\ be}\ B \times D\ \mathsf{in\ persuade}(C, D, \mathsf{meet}(A \oplus B))}\otimes_{\varepsilon,2,4}}{\mathsf{persuade}(C, \mathsf{mary}, \mathsf{meet}(A \oplus \mathsf{mary})) : f_\sigma}\Rightarrow_\beta}{\dfrac{\mathsf{let\ john} \times \mathsf{john\ be}\ A \times C\ \mathsf{in\ persuade}(C, \mathsf{mary}, \mathsf{meet}(A \oplus \mathsf{mary}))}{\mathsf{persuade}(\mathsf{john}, \mathsf{mary}, \mathsf{meet}(\mathsf{john} \oplus \mathsf{mary})) : f_\sigma}\Rightarrow_\beta}\otimes_{\varepsilon,1,3}$$

Figure 15: Glue derivation of *John persuaded Mary to meet*

8 Conclusion

In this paper we have presented the multi-functional control analysis of split control sentences such as *John persuaded Mary to meet in the restaurant*. The multi-functional control analysis represents the understood subject of the controlled sentence at the f-structure level as a set, that consists of the grammatical functions that contribute to the meaning of the understood subject. In order to derive the meaning of the understood subject, meaning meta-constructors, that allow the template-like specification of meaning constructors, were presented. We have also presented how meaning meta-constructors can be used in the analysis of noun-phrase and sentence level coordination.

While both extensions to the LFG Glue formalism were presented in order to solve the problem of the lost plural reading, we believe that these extensions can be applicable to the semantic derivation of other constructs as well.

$$\dfrac{}{\lambda x_1.\lambda x_2.x_1 \oplus x_2 : j_\sigma \multimap (m_\sigma \multimap s_\sigma)}\,\textbf{[coord]}$$

$$\dfrac{[A : j_\sigma]^1 \qquad \lambda x_1.\lambda x_2.x_1 \oplus x_2 : j_\sigma \multimap (m_\sigma \multimap s_\sigma)}{\lambda x_2.A \oplus x_2 : m_\sigma \multimap s_\sigma}$$

$$\dfrac{[B : m_\sigma]^2 \qquad \lambda x_2.A \oplus x_2 : m_\sigma \multimap s_\sigma}{A \oplus B : s_\sigma}$$

$$\dfrac{}{\lambda x.\mathsf{meet}(x) : s_\sigma \multimap p_\sigma}\,\textbf{[meet]}$$

$$\dfrac{A \oplus B : s_\sigma \qquad \lambda x.\mathsf{meet}(x) : s_\sigma \multimap p_\sigma}{\mathsf{meet}(A \oplus B) : p_\sigma}$$

$$\dfrac{}{\lambda P.\lambda y.\lambda x.\mathsf{persuade}(x, y, P) : p_\sigma \multimap (m_\sigma \multimap (j_\sigma \multimap f_\sigma))}\,\textbf{[persuade]}$$

$$\dfrac{\mathsf{meet}(A \oplus B) : p_\sigma \qquad \lambda P.\lambda y.\lambda x.\mathsf{persuade}(x, y, P) : p_\sigma \multimap (m_\sigma \multimap (j_\sigma \multimap f_\sigma))}{\lambda y.\lambda x.\mathsf{persuade}(x, y, \mathsf{meet}(A \oplus B)) : m_\sigma \multimap (j_\sigma \multimap f_\sigma)}$$

$$\dfrac{[D : m_\sigma]^4 \qquad \lambda y.\lambda x.\mathsf{persuade}(x, y, \mathsf{meet}(A \oplus B)) : m_\sigma \multimap (j_\sigma \multimap f_\sigma)}{\lambda x.\mathsf{persuade}(x, D, \mathsf{meet}(A \oplus B)) : j_\sigma \multimap f_\sigma}$$

$$\dfrac{[C : j_\sigma]^3 \qquad \lambda x.\mathsf{persuade}(x, D, \mathsf{meet}(A \oplus B)) : j_\sigma \multimap f_\sigma}{\mathsf{persuade}(C, D, \mathsf{meet}(A \oplus B)) : f_\sigma}$$

$$\dfrac{}{\mathsf{mary} : m_\sigma}\,\textbf{[mary]} \qquad \dfrac{}{\lambda a.a \times a : m_\sigma \multimap (m_\sigma \otimes m_\sigma)}\,\textbf{[objdup]}$$

$$\dfrac{\mathsf{mary} : m_\sigma \qquad \lambda a.a \times a : m_\sigma \multimap (m_\sigma \otimes m_\sigma)}{\mathsf{mary} \times \mathsf{mary} : m_\sigma \otimes m_\sigma}$$

$$\dfrac{\mathsf{mary} \times \mathsf{mary} : m_\sigma \otimes m_\sigma \qquad \mathsf{persuade}(C, D, \mathsf{meet}(A \oplus B)) : f_\sigma}{\mathsf{let\ mary} \times \mathsf{mary\ be}\ B \times D\ \mathsf{in\ persuade}(C, D, \mathsf{meet}(A \oplus B))}\,\otimes_{\varepsilon,2,4}$$

$$\dfrac{\mathsf{let\ mary} \times \mathsf{mary\ be}\ B \times D\ \mathsf{in\ persuade}(C, D, \mathsf{meet}(A \oplus B))}{\mathsf{persuade}(C, \mathsf{mary}, \mathsf{meet}(A \oplus \mathsf{mary})) : f_\sigma}\,\Rightarrow_\beta$$

$$\dfrac{}{\mathsf{john} : j_\sigma}\,\textbf{[john]} \qquad \dfrac{}{\lambda a.a \times a : j_\sigma \multimap (j_\sigma \otimes j_\sigma)}\,\textbf{[subjdup]}$$

$$\dfrac{\mathsf{john} : j_\sigma \qquad \lambda a.a \times a : j_\sigma \multimap (j_\sigma \otimes j_\sigma)}{\mathsf{john} \times \mathsf{john} : j_\sigma \otimes j_\sigma}$$

$$\dfrac{\mathsf{john} \times \mathsf{john} : j_\sigma \otimes j_\sigma \qquad \mathsf{persuade}(C, \mathsf{mary}, \mathsf{meet}(A \oplus \mathsf{mary})) : f_\sigma}{\mathsf{let\ john} \times \mathsf{john\ be}\ A \times C\ \mathsf{in\ persuade}(C, \mathsf{mary}, \mathsf{meet}(A \oplus \mathsf{mary}))}\,\otimes_{\varepsilon,1,3}$$

$$\dfrac{\mathsf{let\ john} \times \mathsf{john\ be}\ A \times C\ \mathsf{in\ persuade}(C, \mathsf{mary}, \mathsf{meet}(A \oplus \mathsf{mary}))}{\mathsf{persuade}(\mathsf{john}, \mathsf{mary}, \mathsf{meet}(\mathsf{john} \oplus \mathsf{mary})) : f_\sigma}\,\Rightarrow_\beta$$

Figure 16: Compact Glue derivation of *John persuaded Mary to meet*

A Glue logic rules

Glue logic rules used in this work were adopted from Asudeh (2004).

- Implication Elimination Introduction

$$\dfrac{\begin{matrix}\vdots\\ a : A\end{matrix} \qquad \begin{matrix}\vdots\\ f : A \multimap B\end{matrix}}{f(a) : B}\multimap_{\mathcal{E}} \qquad\qquad \dfrac{\begin{matrix}[x : A]^1\\ \vdots\\ f : B\end{matrix}}{\lambda x.f : A \multimap B}\multimap_{\mathcal{I},1}$$

- Conjunction Elimination

$$\dfrac{\begin{matrix}\vdots\\ a : A \otimes B\end{matrix} \qquad \begin{matrix}[x : A]^1\ [y : B]^2\\ \vdots\\ f : C\end{matrix}}{\text{let } a \text{ be } x \times y \text{ in } f : C}\otimes_{\mathcal{E},1,2}$$

- Universal Elimination

$$\dfrac{\begin{matrix}\vdots\\ x : \forall H.A\end{matrix}}{x : A[G/H]}\forall_{\mathcal{E}}$$

References

Avery Andrews. Glue logic vs. spreading architecture in LFG. In Christo Moskovsky, editor, *Proceedings of the 2003 Conference of the Australian Linguistics Society*, 2003.

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

Ash Asudeh. *Resumption as resource management*. PhD thesis, Stanford University, 2004.

Ash Asudeh. Control and semantic resource sensitivity. *Journal of Linguistics*, to appear.

Ash Asudeh and Richard Crouch. Coordination and parallelism in glue semantics. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG02 Conference*. CSLI Publications, 2002a.

Ash Asudeh and Richard Crouch. Glue semantics for HPSG. In F. van Eynde, L. Hellan, and D. Beermann, editors, *Proceedings of the 8th. International HPSG Conference*, Stanford, CA, 2002b. CSLI Publications.

Joan Bresnan. Control and complementation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 282–390. MIT Press, Cambridge, MA, 1982.

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

Gennaro Chierchia. Anaphoric properties of infinitives and gerunds. In Mark Cobler, Susannah MacKaye, and Michael Wescoat, editors, *Proceedings of the Third West Coast Conference on Formal Linguistics*, pages 28–39. Stanford Linguistics Association, Stanford, CA, 1984.

Noam Chomsky. *Lectures on government and binding*. Foris, Dordrecht, 1981.

Haskell Brookes Curry and Robert Feys. *Combinatory Logic, Volume I*. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam, 1958. Second printing 1968.

Mary Dalrymple, editor. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. The MIT Press, Cambridge, MA, 1999.

Mary Dalrymple. *Syntax and Semantics*, volume 31 of *Lexical Functional Grammar*. Academic Press, 2001.

Mary Dalrymple, Ron Kaplan, and Tracy Holloway King. Lexical structures as generalizations over descriptions. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG2004 Conference*. CSLI Publications, 2004.

Mary Dalrymple and Tracy Holloway King. Determiner agreement and noun conjunction. *Journal of Linguistics*, 2004.

David Dowty. On recent analyses of the semantics of control. *Linguistics and Philosophy*, 1985.

Jan Engh and Kristian Emil Kristoffersen. The bibliography of control. *Norskrift*, 1996. URL `http://folk.uio.no/janengh/KONTROLL/`. Updated version exists online.

Yehuda N. Falk. *Lexical-Functional Grammar*. Number 126 in CSLI Lecture Notes. CSLI Publications, Stanford, CA, 2001.

Yehuda N. Falk. Infinitival *to*. URL `http://pluto.mscc.huji.ac.il/~msyfalk/Infinitival.pdf`. in preparation.

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

Caroline Heycock and Roberto Zamparelli. Friends and colleagues: Plurality, coordination, and the structure of DP. *Natural Language Semantics*, to appear.

J. Roger Hindley. *Basic simple type theory*. Cambridge University Press, 1997.

W. A. Howard. The formulae-as-types notion of construction. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic Press, New York, N.Y., 1980.

Ray Jackendoff and Peter W. Culicover. The semantic basis of control in english. *Language*, 79(3):517–556, 2003.

Hans Kamp. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof, editors, *Formal Methods in the Study of Language*, pages 277–321. Mathematical Centre Tracts, Amsterdam, 1981.

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

Andrew Kehler, Mary Dalrymple, John Lamping, and Saraswat Vijay. Resource sharing in glue language semantics. In Mary Dalrymple, editor, *Semantics and Syntax in Lexical Functional Grammar*, pages 191–208. MIT Press, 1999.

Idan Landau. *Elements of Control: Structure and Meaning in Infinitival Constructions*. Studies in Natural Language & Linguistic Theory. Kluwer, Dordrecht, 2000.

Fred Landman. *Indefinites and the Type of Sets*. Explorations in Semantics. Blackwell Publishing, 2004.

Godehard Link. The logical analysis of plurals and mass terms: a lattice theoretical approach. In R. Bauerle, C. Schwarze, and A. von Stechow, editors, *Meaning, Use and Interpretation of Language*. De Gruyter, Berlin, 1983.

Richard Montague. English as a formal language. In Bruno Visentini et al., editors, *Linguaggi nella Societa e nella Tecnica*. Edizioni di Comunita, Milan, 1970.

Richard Montague. The Proper Treatment of Quantification in Ordinary English. In Jaakko Hintikka, Julian Moravcsik, and Patrick Suppes, editors, *Approaches to Natural Language*, pages 221–242. Reidel, Dordrecht, 1973.

Michael Moortgat. Categorial type logics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 93–177. Elsevier, Amsterdam, 1997.

Carl Pollard and Ivan Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, 1994.

Ivan Sag and Carl Pollard. An integrated theory of complement control. *Language*, 67(1):63–113, 1991.

Remko J.H. Scha. Distributive, collective and cumulative quantification. In J. Groenendijk, M. Stokhof, and T. M. V. Janssen, editors, *Formal Methods in the Study of Language*. Mathematisch Centrum, Amsterdam, 1981.

Roger Schwarzschild. *Pluralities*, volume 61 of *Studies in Linguistics and Philosophy*. Kluwer, Dordrecht, 1996.

Stephen Wechsler and Larisa Zlatic. *The Many Faces of Agreement*. CSLI Publications, 2003.

Yoad Winter. *Flexibility Principles in Boolean Semantics*, volume 37 of *Current studies in linguistics*. MIT Press, 2001.