Chapter 22

Glue semantics

Ash Asudeh

University of Rochester and Carleton University

Glue Semantics is a general framework for semantic composition and the syntax–semantics interface. It assumes an autonomous syntax and therefore needs to be paired with some syntactic theory. Here the focus is on LFG as the syntactic theory. The Glue logic, a fragment of linear logic, is presented first. This highlights the resource sensitivity of semantic composition in Glue. Second, Glue is presented without reference to LFG or any other syntactic theory. This highlights Glue's property of flexible composition. Third, the syntax–semantics interface is considered. This highlights Glue's autonomy of syntax and serves as a way to compare and constrast Glue with well-known alternatives. Fourth, Glue is paired with LFG (LFG+Glue), which highlights another important property of the theory, syntax/semantics non-isomorphism. Lastly, a number of particular phenomena are briefly reviewed and their analyses sketched: quantifier scope, modification, tense, events, argument structure, multiword expressions, and anaphora.

1 Introduction

The fundamental principle of compositional semantics is the following:

(1) *Principle of Compositionality (PoC)*
The meaning of a whole is a function of the meanings of the parts.
(Partee 1995)

According to the PoC, the meaning of an expression depends on its parts, but also on its syntax. The aspects of syntax that are relevant are standard features like NUM, PERS, TENSE, etc., as well as syntactic predicate-argument relations and local and non-local syntactic dependencies. These are all represented in f(unctional)-structure in LFG, so the relevant syntactic representation for compositional semantics in LFG is f-structure.

Ash Asudeh. Glue semantics. To appear in: Dalrymple, Mary (ed.). Handbook of Lexical Functional Grammar. Berlin: Language Science Press. [preliminary page numbering]

But how are compositionally relevant features and relations obtained from f-structure? This is really a question about the mapping between syntax and semantics, or the nature of the *syntax–semantics interface*.[1] There are two fundamental ways in which different levels in LFG's Correspondence Architecture (Kaplan 1987, 1995) can be related: *description by analysis* and *co-description*. Both methods have been applied to the syntax–semantics interface in LFG.

Halvorsen (1983) developed the initial semantics for LFG, in which an f-structure is analyzed for features, including grammatical functions and other relational dependencies, to obtain a description of the compositional semantics. This is an example of *description by analysis* (Halvorsen & Kaplan 1988; Kaplan 1995) and is similar in spirit to Logical Form (LF) semantics (Heim & Kratzer 1998), even though the input syntactic structures are formally quite different. The description-by-analysis approach to LFG semantics effectively makes the same assumption as LF semantics: the semantic interpretation function applies to an entire syntactic structure — a standard non-tangled tree in LF semantics or an f-structure in description-by-analyses semantics for LFG.

Halvorsen & Kaplan (1988) offered a co-description alternative. According to co-description, a lexical item specifies its c-structural category, which captures its syntactic distribution, and also simultaneously specifies its contributions to f-structure, s(emantic)-structure, and any other grammatical modules. The contribution to f-structure, s-structure, etc., is accomplished through a set of constraints and equalities whose solutions determine the lexical item's non-c-structural contributions.[2] Thus, a syntactic formative on this view simultaneously *co-describes* its contributions to compositional semantics.

Glue Semantics (Glue) further develops and logically systematizes the co-description idea of Halvorsen & Kaplan (1988).[3] In contrast to description by analysis, co-descriptive LFG semantics is in the spirit of the syntax–semantics interface tradition that developed out of the *rule-by-rule* approach of Montague (1973), to use the terminology of Bach (1976). This tradition is standardly exemplified by Categorial Grammar (CG; for a basic overview and foundational references, see

[1]Unfortunately, this term has been somewhat bleached of meaning through overuse in syntactic theory, where the mapping is often not specified in sufficient detail.

[2]See Asudeh (2012: ch. 3) for a basic introduction to one version of the Correspondence Architecture.

[3]The implementation of Glue that was developed for the Xerox Linguistic Environment (XLE) implementation of LFG (Crouch et al. 2011) used description-by-analysis, but out of necessity rather than by design. The co-descriptive version of Glue would have required changes to the underlying XLE implementation, whereas description-by-analysis Glue did not. Also see Andrews (2008) for a consideration of description-by-analysis versus co-description approaches to Glue.

Wood 1993). In fact, Dalrymple, Gupta, et al. (1999) discuss how Glue is strongly related to Categorial Grammar in the type-logical tradition (for overviews and further references, see e.g., Carpenter 1997; Morrill 1994; Morrill 2011; Moortgat 1997).

However, Glue Semantics and Categorial Grammar make distinct assumptions about the relation between the syntax of word order and the syntax of compositional semantics (for discussion and further references on this aspect of CG, see Steedman 2014). LFG's claims about Universal Grammar (Bresnan et al. 2016: ch. 4) serve to highlight the distinction. C-structure, which represents word order, is highly variable cross-linguistically, whereas f-structure, which represents syntactic features and dependencies, is largely invariant cross-linguistically. This is reflected in the fact that although embedding is significant at f-structure, order among features in the same f-structure is not, as shown in (2) and (3):

(2) $$\begin{bmatrix}\textsc{att1} & \begin{bmatrix}\textsc{att2} & \textsc{val}\end{bmatrix}\end{bmatrix}$$
$$\neq \begin{bmatrix}\textsc{att2} & \textsc{val}\end{bmatrix}$$

(3) $$\begin{bmatrix}\textsc{att1} & \textsc{val1} \\ \textsc{att2} & \textsc{val2}\end{bmatrix}$$
$$= \begin{bmatrix}\textsc{att2} & \textsc{val2} \\ \textsc{att1} & \textsc{val1}\end{bmatrix}$$

A language with relatively free word order (e.g., Warlpiri) has quite different c-structures from a language with relatively fixed word order (e.g., English). However, the two languages have similar f-structures, which predicts that they are similar with respect to syntactic features and dependencies (Bresnan et al. 2016: ch. 1). It would be antithetical to the theory for compositional semantics to be computed from c-structure, since the cross-linguistically relevant information for semantics is captured in the unordered f-structure. So Glue Semantics uses a commutative logic for composition, which turns out to yield insights beyond those which originally motivated Glue.[4] This will be explored more carefully below, from a higher level perspective.

The first papers in the Glue Semantics (Glue) framework were published in the mid-nineties (Dalrymple et al. 1993; Dalrymple, Lamping, Pereira, et al. 1995; Dalrymple, Gupta, et al. 1997; Dalrymple, Lamping, Pereira, et al. 1997). The initial major publications on Glue, including revised versions of most of these papers,

[4]Note that the term *logic* here is intended not merely in the sense of a representational language for meaning, but rather a deductive system for deriving formulae from other formulae, i.e., proving conclusions from premises and previously proven conclusions.

appeared in Dalrymple (1999). These publications all assumed some version of LFG syntax. It should be borne in mind, however, that Glue Semantics (Glue) is a general framework for semantic composition and the syntax–semantics interface and is in that sense independent from LFG per se. The key syntactic assumption that Glue makes is headedness, which is universal across formal syntactic theories, even if specifics vary. Glue thus offers a highly flexible and adaptable approach to semantic composition and the syntax–semantics interface. In addition to LFG, Glue Semantics has been defined for a number of syntactic frameworks, including Lexicalized Tree-Adjoining Grammar (Frank & van Genabith 2001), HPSG (Asudeh & Crouch 2002b), Minimalism (Gotham 2018), and Universal Dependencies (Gotham & Haug 2018).

Asudeh (2022: 324) highlights the following high-level properties of Glue Semantics:[5]

1. *Resource-sensitive composition*
   The logic of composition in Glue is *resource-sensitive*: The underlying logic of composition itself requires that all and only the resources/premises instantiated from the syntax are used in semantic composition.

2. *Flexible composition*
   The logic of composition in Glue is *commutative*. Semantic composition is systematically related to and constrained by syntax, but is not determined by syntactic word order. Semantic composition is tightly restricted by resource-sensitive composition.

3. *Autonomy of syntax*
   The logical assumptions of Glue yield a truly autonomous syntax, as a corollary of flexible composition. Semantic composition is commutative, but syntax is not: Syntax is subject to word order constraints that do not apply to semantic composition.

4. *Syntax/semantics non-isomorphism*
   Grammatical formatives, e.g. lexical items, may contribute multiple Glue terms that are all contributed to the semantic proof or no Glue terms at all, as a corollary of autonomy of syntax. There is no requirement that a formative must make exactly one contribution to interpretation.

In Asudeh (2022), I used these properties as organizing themes for a big-picture discussion that mostly backgrounded the combination of LFG in particular with

[5] My thanks to an anonymous reviewer of Asudeh (2022) for suggesting the term *syntax/semantics non-isomorphism.*

Glue (often called LFG+Glue). Here I wish instead to foreground LFG+Glue, but it is nevertheless useful to have these properties in one place, as they will occasionally be referred to below.

I also want to emphasize that these properties are not fully independent, at least as given. The degree of resource sensitivity flows from the particular fragment of *linear logic* (Girard 1987) that one chooses for the Glue logic, but the implicative fragment with universal instantiation is commonly used and this fragment is highly resource sensitive, as explained in the next section. From resource sensitivity flow some automatic constraints on flexible composition such that it's not just 'anything goes.' Flexible composition in turn permits true autonomy of syntax, which fits naturally within LFG's general ethos of allowing mismatches between distinct linguistic modules in the Correspondence Architecture (Kaplan 1987, 1995; Asudeh 2006). Lastly, since syntax is autonomous from semantics, given flexible composition, it does not follow that a compositional analysis is only possible if each formative contributes exactly one meaning to semantic composition. Formatives may contribute nothing to meaning, e.g. expletive subjects or *do*-support *do*, or contribute multiple meanings to semantic composition.

2 The Glue logic: Resource-sensitive composition

The Glue logic is a fragment of *linear logic* (Girard 1987; Crouch & van Genabith 2000). Linear logic can be thought of as a logic of resources: Each premise in a linear logic proof must be used exactly once.[6] This can be usefully understood from the perspective of *substructural logics*. Substructural logics "focus on the behaviour and presence — or more suggestively, the *absence* — of *structural rules*. These are particular rules in a logic which govern the behaviour of collections of information." (Restall 2000: 1–2; emphasis in original). The basic intuition is that the choice of structural rules allows a precise logical characterization of some system of information. Language can be construed as information. For example,

[6]Girard (1987) defines two modal operators for linear logic, ! (*Of course!*) and ? (*Why not?*). These operators prefix particular premises (e.g., $!A$ or $?A$). This allows resource accounting to be turned off for the premise. Some early work in Glue used the ! modal in the analysis of coordination (Kehler et al. 1995, 1999). However, Asudeh (2004, 2005a) argued for a stricter notion of resource sensitivity that results from a simpler modality-free fragment of linear logic. Asudeh & Crouch (2002a) present a polymorphic Glue analysis of coordination (Steedman 1985; Emms 1990, 1992). The Asudeh & Crouch (2002a) approach does not require the modality; also see Dalrymple et al. (2019: ch. 16).

Chomsky (1986, 1995) can be understood as characterizing language as information from a cognitive perspective. Another example is the characterization of language as information from a logical perspective, as in van Benthem (1991).

Three structural rules that are particularly relevant to substructural logics for linguistics are *weakening*, *contraction*, and *commutativity*:[7]

(4) *Weakening* — Intuition: A premise can be *freely added*

$$\frac{\Gamma \vdash B}{\Gamma, A \vdash B}$$

(5) *Contraction* — Intuition: Any additional occurrence of a premise can be *freely discarded*

$$\frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}$$

(6) *Commutativity* — Intuition: Premises can be *freely reordered*

$$\frac{\Gamma, A, B \vdash C}{\Gamma, B, A \vdash C}$$

If a logic *lacks* the rules of weakening and contraction, then premises in the logic cannot be added or discarded and the logic is therefore a *resource logic.*

However, we can also distinguish logics based on commutativity: A resource logic can be commutative or non-commutative. Linear logic is a commutative resource logic. In contrast, the Lambek logic L (Lambek 1958) is a non-commutative resource logic. L is the fundamental logic of the Lambek calculus, the basis for the type-logical approach to Categorial Grammar (see, e.g., van Benthem 1991; Moortgat 1997). The diagram in Figure 1 shows linear logic in a space of related substructural logics.[8]

The appropriate resource logic for semantics alone is a commutative resource logic. Semantic composition is resource-sensitive but does not show evidence of order-sensitivity in its own right (Asudeh 2012: ch. 5). Consider the general case

[7]The notation in these structural rules is understood as follows. Γ denotes a set of terms in the logic, whereas A, B denote particular terms in the logic. The single turnstile denotes a valid derivation/proof from the lefthand side to the righthand side; e.g., $\Gamma \vdash B$ means that B can be proven from Γ. The horizontal line separating the top and bottom of the rule means that the bottom can be derived from the top by the rule in question (i.e., the top *sequent* can be replaced by the bottom one). For example, the weakening rule states that, given $\Gamma \vdash B$, one can conclude $\Gamma, A \vdash B$; i.e., every instance of $\Gamma \vdash B$ can be replaced by $\Gamma, A \vdash B$, given the rule.

[8]Note that the relation between intuitionistic logic and classical logic is characterized by the addition of the law of the excluded middle. However, this law is not strictly a structural rule, hence the dashed rather than solid line in Figure 1.

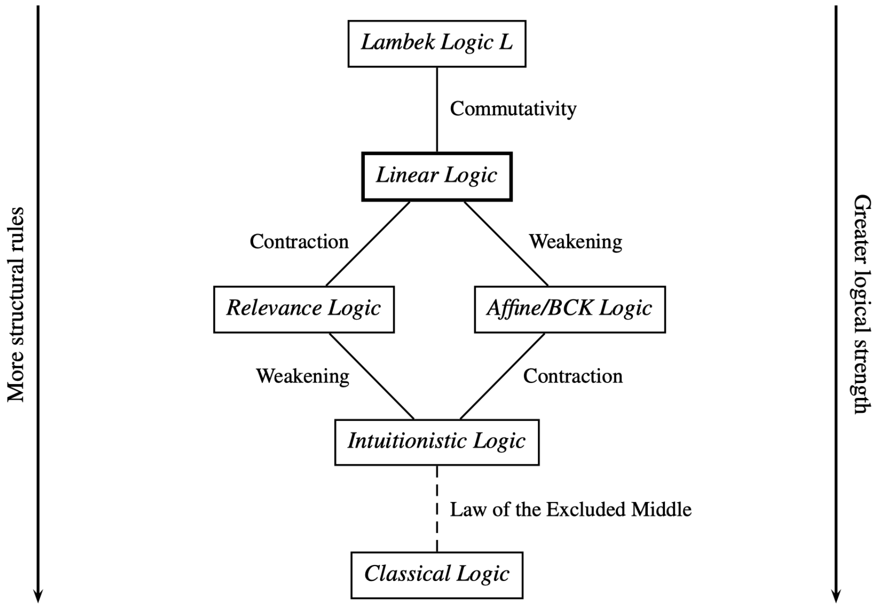


Figure 1: Hierarchy of logics related by structural rules (Asudeh 2012: 103; used with permission)

of some binary structure that is to be interpreted. If one branch denotes a function and the other denotes an argument, the function applies to the argument, whether the function is on the left or right:

(7) $$\left[\!\left[\textbf{function} \quad \textbf{argument} \right]\!\right] = \left[\!\left[\textbf{argument} \quad \textbf{function} \right]\!\right]$$

For example, in English basic word order, the function that is the denotation of a transitive verb takes its argument to the right, but the resulting function that is the denotation of the VP takes its argument to the left.

It is not the *order* of the function and argument that determines their composition, but rather their semantic types (Klein & Sag 1985). This is saliently exemplified by the rule of functional application in the widely familiar system of Heim & Kratzer (1998: 44, 95). It is also exemplified by the equivalent interpretations of the forward and backward slash rules of Combinatory Categorial Grammar (see, e.g., Steedman 1987: 406 or Steedman & Baldridge 2011: 186).

But how is semantics resource-sensitive? The following quote from Klein & Sag (1985: 172) illustrates:

> Translation rules in Montague semantics have the property that the translation of each component of a complex expression occurs exactly once in the translation of the whole. … That is to say, we do not want the set S [of semantic interpretations of a phrase] to contain all meaningful expressions of IL [Intensional Logic] which can be built up from the elements of S, but only those which use each element exactly once.

In other words, Montague's (1973) translation rules are resource-sensitive. However, this is merely coincidental as far as his translation process is concerned. In their generalization of Montague's system, Klein & Sag (1985: 174) need to define an operation of *bounded closure*. This operation ensures that the meaning of each element of semantic composition is indeed used "exactly once."

We can obtain this result in a more general way, if we adopt a resource logic for semantic composition. This rests on the absence of the structural rules of contraction and weakening. The lack of contraction means that the number of occurrences of a premise matters, so a set of linear logic premises is a *multiset* (sometimes called a *bag*). The lack of weakening means that the bag must be emptied in constructing a valid proof. In other words, it follows directly from the absence of contraction and weakening that "each element" must be used "exactly once". Klein & Sag's (1985) bounded closure is effectively an attempt to capture the logical resource sensitivity of linear logic or L (Asudeh 2012: 110–111).

Logical resource sensitivity in turn forms the basis for linguistic resource sensitivity (Asudeh 2012: ch. 4). This is achieved by placing a linguistically motivated goal condition on the Glue logic proof; for example, we can require that the proof of a sentence terminates in a single meaning constructor of type t (Dalrymple, Gupta, et al. 1999). Asudeh (2012: 110–123) argues that resource-sensitive composition not only directly captures bounded closure, it arguably also captures a diverse set of principles across a variety of frameworks. These include Completeness and Coherence (Kaplan & Bresnan 1982), the Theta Criterion (Chomsky 1981), the Projection Principle (Chomsky 1981, 1982, 1986), No Vacuous Quantification (Chomsky 1982, 1995; Kratzer 1995; Kennedy 1997; Heim & Kratzer 1998; Fox 2000), the Principle of Full Interpretation (Chomsky 1986, 1995), and the Inclusiveness Condition (Chomsky 1995).

In addition, it seems that phonology and syntax can equally be considered resource-sensitive, i.e. lack weakening and contraction from a logical perspective, as outlined in Asudeh (2012: 98–99). This allows a deeper generalization about natural language as computation (Steedman 2007), namely that *natural language is resource-sensitive*. The claim is set out in the *Resource Sensitivity Hypothesis* (Asudeh 2012: 95). Where phonology and syntax contrast with semantics is not with respect to weakening and contraction, but rather with respect to commutativity. Phonology is strictly non-commutative, whereas syntax shows commutativity in some circumstances of free word order. This leaves two options. The partial commutativity of syntax can be captured by separating the syntax of structure from the syntax of composition, treating the syntactic module(s) autonomously, as in Glue Semantics. Alternatively, partial commutativity can be captured by not separating structural and compositional syntax and instead introducing a mechanism to the syntax–semantics interface that relaxes commutativity in what is otherwise a non-commutative system. An example of such a mechanism is the categorial modalities of Baldridge (2002).

3 Glue without LFG: Flexible composition

Linguistic meanings in Glue are encoded in *meaning constructors*. Meaning constructors are pairs of terms from two logics. These terms can be represented as $\mathcal{M}$ and G (where $\mathcal{M}$ is mnemonic for *meaning language* and G is mnemonic for *Glue logic*). These could be written in any conventional way for writing pairs, such as $\langle\mathcal{M}, G\rangle$, but most Glue work of the past couple of decades has written the pair using an uninterpreted colon as a pairing symbol, as in (8):

(8) $\mathcal{M} : G$

The meaning language can be anything that supports the lambda calculus, such as the simply typed lambda calculus that is often used in linguistic semantics. However, more specialized lambda languages can be used, as in van Genabith & Crouch (1999), Bary & Haug (2011), and Lowe (2015), which all use Muskens's (1996) Compositional Discourse Representation Theory (CDRT) or Dalrymple et al. (2019: ch. 14), which uses Haug's (2014) partialized version, Partial Compositional DRT (PCDRT). The glue logic is a fragment of *linear logic* (Girard 1987). The glue logic specifies semantic composition based on a syntactic parse that instantiates the general terms in G to a specific syntactic structure. The meaning constructors thus serve as *premises* in a linear logic *proof* of the compositional semantics.

The linear logic implication connective, $\multimap$,[9] is the basis for the fundamental compositional rule of functional application. Functional application corresponds to linear implication elimination in natural deduction style:

(9) Functional application:
Implication elimination — modus ponens

$$\frac{\beta : A \multimap B \qquad \alpha : A}{\beta(\alpha) : B} \multimap_{\mathcal{E}}$$

[9] This is the *multimap* symbol, but it is often referred to in Glue discourse as the *lollipop*.

The implication elimination rule is standard modus ponens. The rule is read as follows: given $\beta : A \multimap B$ and given $\alpha : A$, it is valid to conclude $\beta(\alpha) : B$.

The Curry-Howard Isomorphism (CHI; Curry & Feys 1958; Howard 1980) determines the correspondence between the term to the left of the colon — a term from $\mathcal{M}$ — and the term to right of the colon — a term from G. The CHI puts logical formulas in correspondence with computational types. Here linear logic formulas are in correspondence with types in the lambda calculus.[10] The terms A, B in (9) are schematic for possibly complex formulas; α, β may similarly be complex terms.

The rule for linear implication introduction corresponds to functional abstraction.

(10) Functional abstraction:
Implication introduction — hypothetical reasoning

$$\dfrac{\begin{array}{c}[\alpha : A]^1 \\ \vdots \\ \beta : B\end{array}}{\lambda\alpha.\beta : A \multimap B} \multimap_{\mathcal{I},1}$$

In this schema, a hypothesis is uniquely flagged with a numerical index. The fact that it is a hypothesis — i.e. not a premise encoded by a meaning constructor — is indicated by square brackets. If a conclusion can be derived through some series of proof steps (indicated by the vertical ellipsis), given the hypothesis, then we know that the hypothesis implies the conclusion: the hypothesis is discharged (as the antecedent of an implication with the conclusion as the consequent) and its flag is withdrawn. In the meaning language, this corresponds to abstraction over the variable introduced on the meaning language side of the hypothesis.

Let's turn to a simple linguistic example:

(11) Blake called Alex.

[10] Some early papers in Glue (Dalrymple et al. 1993; Dalrymple, Lamping, Pereira, et al. 1995, 1997; Crouch & van Genabith 1999; van Genabith & Crouch 1999; Fry 1999a; Kehler et al. 1999) used a more ad-hoc method of relating the meaning terms to the Glue logic, but Dalrymple, Gupta, et al. (1997); Dalrymple, Gupta, et al. (1999) introduced the Curry-Howard approach to Glue, which is now standard. Kokkonidis (2008) introduced an alternant called *First-Order Glue* which has also proven influential in subsequent Glue literature (e.g., Bary & Haug 2011, Lowe 2014, Gotham 2018, Gotham & Haug 2018, Findlay 2019; see also Andrews 2010 for a related proposal).

Let us assume the following meaning constructor for the verb *called*, leaving tense aside:[11]

(12) $\lambda y.\lambda x.call(y)(x) : a \multimap b \multimap c$

On the Glue side, c is mnemonic for *called*, a for *Alex*, and b for *Blake*. This meaning constructor would in fact be specified in some general form but instantiated relative to a particular syntactic structure. For now, let us just assume that some instantiation has given us the meaning constructor in (12). In Section 5 below, we'll see how to specify meaning constructors in general terms given LFG's usual f-description language.

Assuming that the lexical entries for *Alex* and *Blake* contribute meaning constructors that are instantiated to $alex : a$ and $blake : b$, we can construct the following proof, given (12); note that $\Rightarrow_\beta$ indicates β-reduction of a lambda term.

(13)
$$\dfrac{blake : b \qquad \dfrac{\lambda y.\lambda x.call(y)(x) : a \multimap b \multimap c \qquad alex : a}{\lambda x.call(alex)(x) : b \multimap c}\;{\multimap_{\mathcal{E}}, \Rightarrow_\beta}}{call(alex)(blake) : c}\;{\multimap_{\mathcal{E}}, \Rightarrow_\beta}$$

The meaning term in the conclusion is equivalent to $call(blake, alex)$ in the commonly used relational notation (Montague 1973).

Note that proofs are abstract mathematical objects that can be written down in various ways. This is quite apart from whatever convention or notation we choose for writing them down. For example, even holding constant our natural deduction notation, what is shown in (13) is just one of four ways to write down the single abstract normal form proof (Prawitz 1965). Writing the proof down imposes an order,[12] but since the Glue logic is commutative (see Section 2 for further details), all four written representations of the proof are equivalent.

Given the commutativity of the Glue logic, the arguments of the function can be freely reordered (re-curried), as in (14) below, but still yield the appropriate meaning:

(14) $\lambda x.\lambda y.call(y)(x) : b \multimap a \multimap l$

Example (15) below is a schematic demonstration of how this argument reordering works in a proof; the example abstracts away from the particular *call* function. The example also shows the implication introduction rule in action.

[11]Note that the lambda term $\lambda y.\lambda x.call(y)(x)$ is equivalent to the function *call* by η-equivalence in the lambda calculus. However, it is useful for the exposition below to present it in non η-reduced form.

[12]The *Alex* meaning constructor/premise must be written either to the right or left of the functional (verb) meaning constructor and similarly for the *Blake* meaning constructor/premise.

(15)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lambda y.\lambda x.f(y)(x) : a \multimap b \multimap c \qquad [v : a]^1}{\lambda x.f(v)(x) : b \multimap c}\multimap_{\mathcal{E}} \qquad [u : b]^2}{f(v)(u) : c}\multimap_{\mathcal{E}}}{\lambda v.f(v)(u) : a \multimap c}\multimap_{\mathcal{I},1}}{\lambda u.\lambda v.f(v)(u) : b \multimap a \multimap c}\multimap_{\mathcal{I},2}}{\lambda x.\lambda y.f(y)(x) : b \multimap a \multimap c}\Rightarrow_{\alpha},\Rightarrow_{\alpha}$$

The result is a reordered form of the original term but without any change in meaning, because the CHI ensures that the function's arguments in the meaning terms are also appropriately reordered. The α-equivalences, in which variables are renamed, are not strictly necessary, but have been added for full transparency. In general, given n arguments in the order $a_1 \dots a_n$, a reverse order $a_n \dots a_1$ can be obtained by a series of hypotheses on the arguments that are discharged in the order they were made. More generally, the arguments can be reordered in any order by mixing the order of hypothesis assumption and discharge.

4 The syntax–semantics interface: Autonomy of syntax

Glue rests on two general assumptions about the syntax–semantics interface:

1. The *logical syntax* of semantic composition (Fenstad et al. 1987) is distinct from the *structural syntax*. The syntax of linear logic proofs captures the logical syntax in Glue. Some separate syntactic framework, such as LFG, captures the structural syntax of categorially determined distribution, constituency, features, and local and non-local dependencies (i.e, *syntax* in the standard sense).

2. Logical syntax and structural syntax are systematically related through the instantiation of Glue meaning constructors.

These assumptions distinguish Glue from both *interpretive theories* of semantic composition and *parallel theories* of semantic composition. A well-known example of interpretive theories is Logical Form semantics (e.g., Heim & Kratzer 1998). The *description-by-analysis* semantics for LFG of Halvorsen (1983) is another example of an interpretive theory. Two well-known examples of parallel theories are Combinatory Categorial Grammar (e.g., Steedman & Baldridge 2011) and Type-Logical Categorial Grammar (e.g., Carpenter 1997).[13]

With respect to LF semantics, Glue's assumption of a separate level of structural syntax is similar. However, in its standard co-descriptive guise, Glue is distinct from LF semantics, because Glue does not assume that the syntactic structure in its entirety is the input to semantic interpretation. With respect to Categorial Grammar, we also see similarity and divergence. We see similarity in Glue's assumption of the pairing of functional application (the fundamental compositional operation) with terms that define complex categories implicationally.[14] However, Glue is also distinct from Categorial Grammar, because Glue does not assume that implicational categories are responsible for word order (hence their lack of directionality), but rather that there is a separate syntactic representation. In sum, Glue is a compositional semantic theory of a third kind. From a big picture perspective, Glue synthesizes certain aspects of LF semantics and Categorial Grammar, yet remains distinct from both these theories.

The assumptions, in 1 and 2 above, that began this section derive a strong notion of syntactic autonomy. Categorial Grammar makes the very strong assumption that syntax and semantics are isomorphic. This assumption entails that any semantic distinction must be the reflection of a syntactic distinction. LF semantics similarly assumes that any interpretive/semantic distinction must be due to an underlying syntactic distinction. In an interpretive semantic theory such as LF semantics, the needs of semantics dictate what's in the syntax, even if the things in question are syntactically questionable. The predicate abstraction/numerical nodes in Heim & Kratzer (1998: 186) are an example, since they require the addition of lambda operators to the syntactic tree. This is surprising from the perspective of semantic theory, since this means that *object* languages, i.e. the natural languages undergoing analysis, must in fact contain these logical operators, for which there is no compelling evidence (such as lexicalization in some language or other).

[13] Parallel theories are often discussed under the rubric of "rule-by-rule composition" (Bach 1976), but the *rule-by-rule* term is no longer accurate. The term originates in the paired syntactic/semantic rules of Montague (1973), which is now deprecated. This kind of theory is also sometimes referred to as "direct compositionality" (Barker & Jacobson 2007; Jacobson 2014), but this raises a number of issues (Asudeh 2006), so I do not favour that term.

[14] Categorial Grammar's slashes are directed implications. For example, X/Y states that one can conclude a category X conditional on there being a category Y to the right; in other words, X/Y means that $Y \rightarrow X$ yields X so long as Y is on the right of X.

Quantifier scope ambiguity offers perhaps the most straightforward demonstration of the distinction between Glue on the one hand, and LF semantics and Categorial Grammar, on the other. Consider the following standard example:

(16) Everybody loves somebody.

The Glue logic computes two readings for this sentence, but *without* imputing a syntactic ambiguity, which seems structurally under-motivated; see s 6.1 for further details. This contrasts with both LF semantics and Categorial Grammar; these theories both require the two readings to be syntactically distinguished.

In the next section, I pair Glue with LFG as the syntactic framework in order to render these general points more specific. Although LFG is the natural syntactic framework to choose, given the present venue and the fact that most Glue work has assumed an LFG syntax, see Section 1 above for a list of other syntactic frameworks that have been paired with Glue.

5 Glue with LFG: Syntax/semantics non-isomorphism

Consider the example in Figure 2, which shows the c-structures and f-structure for the sentence *I drank water* in Finnish and English.[15] The distinct c-structures capture the variation in syntactic realization between the two languages. In particular, they capture the fact that Finnish allows null subjects, unlike English. The f-structure shows that these distinct c-structures encode identical syntactic features and dependencies. Figure 3 shows the same structures with the arrows resolved. One way to solve the equations is to label all c-structure nodes that bear a down arrow with an f-structure variable. Instantiation of the metavariables ↑ and ↓ is arbitrary, barring accidental identity, and resolves the equalities (Bresnan et al. 2016: 54–58).

In both the Finnish and English c-structures, the mapping to OBJECT is contributed structurally by the annotation (↑ OBJ) = ↓ on the NP daughter of V′. In the English c-structure, the mapping to SUBJECT is also contributed structurally, by the annotation (↑ SUBJ) = ↓ on the DP in SpecIP. In contrast, the SUBJECT information is contributed morphologically in the Finnish c-structure. This distinction is reflected in the lexical entries in Table 1. Notice that the f-descriptions in these lexical entries not only describe their lexical contributions to f-structure, but also have appropriate Glue meaning constructors that define the mappings

[15] I assume LFG's theory of *extended heads*, which allows the Finnish verb to be generated in I (Bresnan et al. 2016: ch. 6–7).

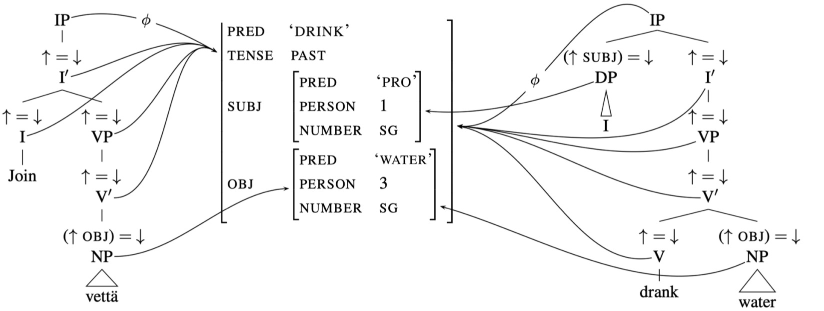


Figure 2: C-structures and f-structure for *I drank water* in Finnish and English (adapted from Asudeh & Toivonen 2015: 27; used with permission)

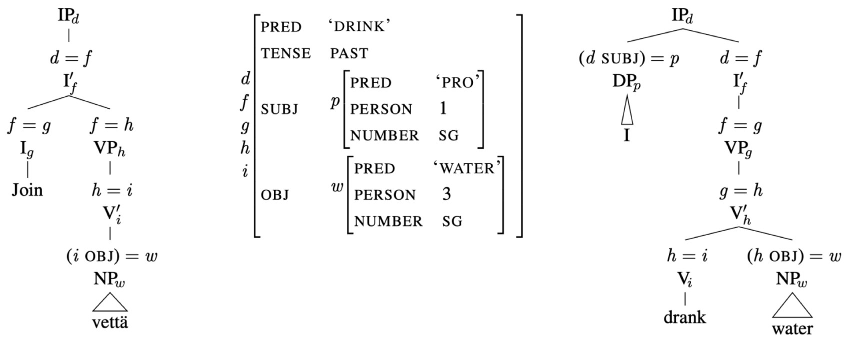


Figure 3: Finnish and English structures with ↑ and ↓ metavariables resolved (Asudeh 2022: 330; used with permission)

to s(emantic)-structure and encode the composition of the head and its dependents as linear implications.[16] I have set tense aside in the semantics, but return to it in Section 6.3 below. The annotation σ on the arrows in the Glue meaning constructors indicates that these are the s-structure correspondents of the relevant f-structures. The σ correspondence function maps from f-structure to s-structure.

The up arrows in Table 1 are instantiated to the f-structure of the relevant pre-terminal node: g (Finnish *join*), w (Finnish *vettä*), p (English *I*), i (English *drank*), and w (English *water*).[17] However, we know from Figure 3 that $g = i = d$.

[16]The asterisk in the term for *vettä*/*water* is the cumulativity operator of Link (1983). It states that *water* is a mass term, although this is not important for our present purposes.

[17]In the case of the abbreviated (triangle) structures, there would be intervening nodes. But there would be a chain of ↑=↓ annotations between the word and the phrase it heads, so this is a harmless simplification.

Table 1: Lexicons for *I drank water* in Finnish and English

Finnish			English		
join	I	$(\uparrow \text{PRED}) = \text{'DRINK'}$ $(\uparrow \text{TENSE}) = \text{PAST}$ $(\uparrow \text{SUBJ PRED}) = \text{'PRO'}$ $(\uparrow \text{SUBJ PERS}) = 1$ $(\uparrow \text{SUBJ NUM}) = \text{SG}$ $\mathit{speaker} : (\uparrow \text{SUBJ})_\sigma$ $\mathit{drink} :$ $(\uparrow \text{OBJ})_\sigma \multimap (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$	*I*	D	$(\uparrow \text{PRED}) = \text{'PRO'}$ $(\uparrow \text{PERS}) = 1$ $(\uparrow \text{NUM}) = \text{SG}$ $\mathit{speaker} : \uparrow_\sigma$
			drank	V	$(\uparrow \text{PRED}) = \text{'DRINK'}$ $(\uparrow \text{TENSE}) = \text{PAST}$ $\mathit{drink} :$ $(\uparrow \text{OBJ})_\sigma \multimap (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$
vettä	N	$(\uparrow \text{PRED}) = \text{'WATER'}$ $(\uparrow \text{PERS}) = 3$ $(\uparrow \text{NUM}) = \text{SG}$ $^*\mathit{water} : \uparrow_\sigma$	*water*	N	$(\uparrow \text{PRED}) = \text{'WATER'}$ $(\uparrow \text{PERS}) = 3$ $(\uparrow \text{NUM}) = \text{SG}$ $^*\mathit{water} : \uparrow_\sigma$

So we can just use the mnemonic label d in all relevant cases. We can also take advantage of the equality $(d\ \text{SUBJ}) = p$. We obtain the following collection of identical instantiated meaning constructors for each language:

(17) $\{\mathit{speaker} : p_\sigma,\ \mathit{drink} : w_\sigma \multimap p_\sigma \multimap d_\sigma,\ ^*\mathit{water} : w_\sigma\}$

This yields a single normal form proof (i.e., minimal proof; Prawitz 1965) for the corresponding Finnish and English sentences, which can be presented in natural deduction format as follows (recall that order of premises on a proof line does not matter, since the Glue logic is commutative):

(18)
$$\dfrac{\mathit{speaker} : p_\sigma \qquad \dfrac{\mathit{drink} : w_\sigma \multimap p_\sigma \multimap d_\sigma \qquad ^*\mathit{water} : w_\sigma}{\mathit{drink}(^*\mathit{water}) : p_\sigma \multimap d_\sigma}\multimap_\mathcal{E}}{\mathit{drink}(^*\mathit{water})(\mathit{speaker}) : d_\sigma}\multimap_\mathcal{E}$$

6 Some applications of glue semantics

6.1 Quantifier scope

Let us return to the quantifier scope example in (16) above, repeated here as (19).

(19) Everybody loves somebody.

Glue's properties of autonomy of syntax and flexible composition allow (19) to be treated as *syntactically unambiguous* but *semantically ambiguous*.

I will not show the c-structure here, as the relevant syntactic representation is the single f-structure for (19) shown here, with mnemonic labels as usual:

(20)
$$ {}_{l}\begin{bmatrix} \textsc{pred} & \text{`}\textsc{love}\text{'} \\ \textsc{tense} & \textsc{pres} \\ \textsc{subj} & {}_{e}\begin{bmatrix} \textsc{pred} & \text{`}\textsc{everybody}\text{'} \\ \textsc{person} & 3 \\ \textsc{number} & \textsc{sg} \end{bmatrix} \\ \textsc{obj} & {}_{s}\begin{bmatrix} \textsc{pred} & \text{`}\textsc{somebody}\text{'} \\ \textsc{person} & 3 \\ \textsc{number} & \textsc{sg} \end{bmatrix} \end{bmatrix} $$

The Glue meaning constructors in the lexical entries are shown in (21). Tense has again been set aside and it is again most transparent for expository purposes to show the meaning term for *loves* in non-η-reduced form (see footnote 11 on η-reduction).

(21)
$$\begin{array}{rll} \textit{everybody} & \mathrm{D} & \lambda Q.\mathit{every}(\mathit{person}, Q) : \forall S.(\uparrow_\sigma \multimap S) \multimap S \\ \textit{somebody} & \mathrm{D} & \lambda Q.\mathit{some}(\mathit{person}, Q) : \forall S.(\uparrow_\sigma \multimap S) \multimap S \\ \textit{loves} & \mathrm{V} & \lambda y.\lambda x.\mathit{love}(y)(x) : (\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma \end{array}$$

When we instantiate the meaning constructors in (21) relative to the f-structure in (20), we get:

(22)
$$\begin{aligned} \Gamma = \{ & \lambda y.\lambda x.\mathit{love}(y)(x) : s \multimap e \multimap l, \\ & \lambda Q.\mathit{every}(\mathit{person}, Q) : \forall S.(e \multimap S) \multimap S, \\ & \lambda Q.\mathit{some}(\mathit{person}, Q) : \forall S.(s \multimap S) \multimap S \} \end{aligned}$$

The functions *every* and *some* are standard quantificational determiners from generalized quantifier theory (Montague 1973; Barwise & Cooper 1981; Keenan & Faltz 1985), with type $\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle$. The function *every* is defined as $\lambda P.\lambda Q.P \subseteq Q$. The function *some* is defined as $\lambda P.\lambda Q.P \cap Q \neq \emptyset$. In these formulas, P is the set of entities that is the determiner's restriction and Q is the set of entities that is its scope. The quantifier $\lambda Q.\mathit{every}(\mathit{person}, Q)$ thus returns true if the set of people is a subset of its scope set. Similarly, the quantifier $\lambda Q.\mathit{some}(\mathit{person}, Q)$ returns true if the intersection of the set of people and its scope set is non-empty.

A comment is in order about the universal quantification symbol $\forall$ in the Glue terms for the quantifiers. This universal ranges over variables in the Glue logic. It allows the quantifier scope over *any* Glue logic dependency on the semantic correspondent of the quantifier. Asudeh (2005b: 393–394) discusses the interpretation of ∀ in linear logic. The key insight is that, given the resource sensitivity of linear logic, the universal means “any one”, not “all”. The function of the linear universal is to define *scope points* and its interpretation is not related to the quantificational force in the meaning language. Observe that *every* and *some* alike are associated with these linear universal scope terms, even though *some* has existential force.

The meaning constructors in (22) yield exactly two normal form/minimal proofs. These can be represented as in Figure 4 and Figure 5.[18] In other theories, quantifier scope ambiguity requires either a syntactic operation such as Quantifier Raising (QR) in Logical Form semantics (May 1977, 1985; Heim & Kratzer 1998) or a type shifting operation and corresponding categorial modification of some kind, as in Combinatory or Type-Logical Categorial Grammar semantics (Partee & Rooth 1983; Hendriks 1993). Thus, interpretive and parallel theories of composition alike impute a syntactic ambiguity to handle quantifier scope ambiguity.[19]

This contrasts with Glue Semantics. The fact that Glue assumes an independent level of syntax (autonomy of syntax) allows composition to be flexible (flexible composition), which in turn allows the theory to derive the two distinct scope readings without positing a syntactic ambiguity or type shift.

$$\dfrac{\lambda y.\lambda x.love(y)(x) : s \multimap e \multimap l \qquad [v : s]^1}{\lambda x.love(v)(x) : e \multimap l}\ {\multimap_{\mathcal{E}}}, \Rightarrow_\beta$$

$$\dfrac{\lambda x.love(v)(x) : e \multimap l \qquad [u : e]^2}{love(v)(u) : l}\ {\multimap_{\mathcal{E}}}, \Rightarrow_\beta$$

$$\dfrac{love(v)(u) : l}{\lambda y.love(y)(u) : s \multimap l}\ {\multimap_{\mathcal{I},1}}, \Rightarrow_\alpha$$

$$\dfrac{\lambda Q.some(person, Q) : \forall S.(s \multimap S) \multimap S \qquad \lambda y.love(y)(u) : s \multimap l}{some(person, \lambda y.love(y)(u)) : l}\ {\multimap_{\mathcal{E}}}, \forall_{\mathcal{E}}[l/S], \Rightarrow_\beta$$

$$\dfrac{some(person, \lambda y.love(y)(u)) : l}{\lambda x.some(person, \lambda y.love(y)(x)) : e \multimap l}\ {\multimap_{\mathcal{I},2}}, \Rightarrow_\alpha$$

$$\dfrac{\lambda Q.every(person, Q) : \forall S.(e \multimap S) \multimap S \qquad \lambda x.some(person, \lambda y.love(y)(x)) : e \multimap l}{every(person, \lambda x.some(person, \lambda y.love(y)(x))) : l}\ {\multimap_{\mathcal{E}}}, \forall_{\mathcal{E}}[l/S], \Rightarrow_\beta$$

Figure 4: Surface scope interpretation of *Everybody loves somebody*

[18]The universal linear instantiation step is trivial, as in classical/intuitionistic logic. I have therefore not shown it explicitly. See Asudeh (2012: 396) for the rule.

[19]Jacobson (2014: ch. 14) offers a textbook comparison of the LF and CG approaches.

$$
\dfrac{\begin{array}{l}\lambda Q.\mathit{some}(\mathit{person},Q) :\\ \forall S.(s \multimap S) \multimap S\end{array} \qquad \dfrac{\dfrac{\begin{array}{l}\lambda Q.\mathit{every}(\mathit{person},Q) :\\ \forall S.(e \multimap S) \multimap S\end{array} \qquad \dfrac{\begin{array}{l}\lambda y.\lambda x.\mathit{love}(y)(x) :\\ s \multimap e \multimap l\end{array} \qquad [v : s]^1}{\begin{array}{l}\lambda x.\mathit{love}(v)(x) :\\ e \multimap l\end{array}} \multimap_{\mathcal{E}}, \Rightarrow_\beta}{\mathit{every}(\mathit{person}, \lambda x.\mathit{love}(v)(x)) : l} \multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[l/S], \Rightarrow_\beta}{\begin{array}{l}\lambda y.\mathit{every}(\mathit{person}, \lambda x.\mathit{love}(y)(x)) :\\ s \multimap l\end{array}} \multimap_{\mathcal{I},1}, \Rightarrow_\alpha}{\mathit{some}(\mathit{person}, \lambda y.\mathit{every}(\mathit{person}, \lambda x.\mathit{love}(y)(x))) : l} \multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[l/S], \Rightarrow_\beta
$$

Figure 5: Inverse scope interpretation of *Everybody loves somebody*

6.2 Modification

Glue is similar to Categorial Grammar in offering an analysis of semantic *modification* such that modifiers are easily identifiable by their formal shape. For example, the nominal modification category in (23) has its Glue logic analog in (24) (leaving the meaning language aside):

(23) N/N

(24) $A_{\langle e,t\rangle} \multimap A_{\langle e,t\rangle}$

A nominal modifier is a functional category/type that takes a nominal category/-type as an input and returns the same category/type as an output. The modificational semantics is captured on the meaning language side.

For example, a Glue meaning constructor for the attributive adjective *Finnish* would look like (25).

(25) $\lambda P \lambda x.P(x) \wedge \mathit{finnish}(x) : (a_e \multimap b_t) \multimap (a_e \multimap b_t)$

Continuing the example, the common noun *city* would provide the $\langle e,t\rangle$ input to the main implication in (25), such that *Finnish city* would correspond to the following (composed) result:

(26) $\lambda x.\mathit{city}(x) \wedge \mathit{finnish}(x) : (a_e \multimap b_t)$

More generally, a modifier of any type corresponds to a meaning constructor with the following form:

(27) $\lambda f.\mathit{mod}(f) : X \multimap X$

The function *mod* is a placeholder for whatever the semantic effect of the modifier is.

The property of syntax-semantics non-isomorphism, which allows a lexical item to contribute multiple meaning constructors, allows a natural and elegant analysis of so-called *recursive modification* (Kasper 1997). In a nominal like the following, the result we want is that it is apparently the case that the city in question is Finnish:

(28) apparently Finnish city

In other words, we somehow want to maintain a consistent semantics for *apparently* as a modifier, while nevertheless allowing it to fulfill this modificational role inside a nominal. This is despite the type clash between the modifier, which expects a proposition-forming type as input, and the adjective *Finnish*, which does not have this type. The adjective instead has the type of a modifier, i.e. a function on the type that the interpretation of *apparently* expects.

The solution in Glue is to associate predicative and attributive adjectives with the property denotation for the adjective, shown in the first line of (29), and to further add a general nominal modification meaning constructor to the lexical entry for the attributive adjective, as in the second line of (29).

(29) *Finnish* $\lambda x.\mathit{finnish}(x) : v \multimap f$
$(\ \lambda Q \lambda P \lambda x.Q(x) \wedge P(x) : (v \multimap f) \multimap (a \multimap b) \multimap (a \multimap b) \ )$

The reader can verify that the combination of these two meaning constructors yields the meaning constructor in (25) above (with types omitted). The second meaning constructor is treated as optional to ensure that predicative uses of the adjective work as expected. Resource-sensitive composition ensures that a predicative occurrence of the adjective cannot use the second meaning constructor whereas an attributive occurence must use it.

The revised analysis allows recursive modification by a modifier like *apparently*, assuming that we have a meaning constructor like the following associated with *apparently*, suitably instantiated to an f-structure where *apparently* is in the ADJ set of *finnish*:

(30) *apparently* $\lambda P \lambda x.\mathit{apparently}(P(x)) : (v \multimap f) \multimap (v \multimap f)$

The combination of this meaning constructor for *apparently* and the first meaning constructor in (29) then yields the following:

(31) $\lambda x.\mathit{apparently}(\mathit{finnish}(x)) : v \multimap f$

This is sufficient for a predicative occurrence, as in *Marimekko is apparently Finnish.*

For attributive occurrences, (31) then combines with the second meaning constructor in (29), which yields the desired result:

(32) $\lambda P \lambda x.apparently(finnish(x)) \wedge P(x) : (a \multimap b) \multimap (a \multimap b)$

This would then combine with the interpretation of *city* to yield the correct interpretation for, e.g., *The apparently Finnish city is nice.*

I leave aside here the natural extension that is necessary to fully capture recursive modification in examples like the following:

(33) apparently obviously Finnish

The extension just involves having two separate meaning constructors for the adverbial modifier *obviously* (and *apparently*, etc.), in order to make the system fully general, much as we have for the adjective *Finnish* in (29).

The first proposal for the extended modificational semantics presented here was in Dalrymple (2001: 255–274), to my knowledge. The most recent version of the LFG+Glue approach to modification, including recursive modification, is the subject matter of Dalrymple et al. (2019: ch. 13).

6.3 Tense

The basic approach to modification that was sketched at the beginning of Section 6.2 supports a simple account of tense as a modifier on a basic verb meaning, provided that we add a tense coordinate to verb meanings (for a review of approaches to tense in compositional semantics, see Grønn & von Stechow 2016).

Let us assume that a basic meaning constructor for a verb now looks like this:

(34) *sigh* $\lambda x \lambda t.sigh(t, x) : subj \multimap tense \multimap verb$

Let's also assume, following Haug 2008, that u stands for utterance time (what Grønn & von Stechow 2016 denote as s^*, for speech time). Then we can capture simple present, past and future tense as follows, with the Glue logic instantiated suitably per the terms in (34):[20]

(35) a. PAST $\lambda P \lambda t.P(t) \wedge t \prec u : (tense \multimap verb) \multimap (tense \multimap verb)$
b. PRESENT $\lambda P \lambda t.P(t) \wedge t = u : (tense \multimap verb) \multimap (tense \multimap verb)$
c. FUTURE $\lambda P \lambda t.P(t) \wedge u \prec t : (tense \multimap verb) \multimap (tense \multimap verb)$

This sort of account is obviously too simple, but it illustrates tense as a modifier. Note that I've presented the tenses "on their own" for maximal perspicuity, but

[20] These sorts of meanings assume a model of time as consisting of points, but it may well be preferable to think of time as consisting of intervals (Dowty 1979). An interval-based semantics poses no problem for tense in Glue Semantics per se, but I've chosen to keep things simple here.

in a lexicalist framework such as LFG, one would normally assume that tense-inflected forms are inserted in the syntax as words, formed morpholexically. That would just mean that, for example, the inflected form *sighed* would contribute both the meaning constructor in (34) and the one in (35a). Note also that I assume some kind of suitable eventual existential closure of the temporal variable.

One could also incorporate grammatical (as opposed to lexical) aspect in a similar, modificational manner. For analyses of tense and grammatical aspect in Glue Semantics, see Haug (2008), Bary & Haug (2011), and Lowe (2014, 2015).[21]

6.4 Events

The first Glue analysis to incorporate event semantics (Davidson 1967; Parsons 1990; Champollion 2017) was never published (Fry 1999b, 2005). To my knowledge, the first major publications to use event semantics were Asudeh & Toivonen (2012) and Asudeh et al. (2013). Much like the analysis of tense sketched above, event semantics for Glue involves adding a dependency on an event variable. Moreover, work in event semantics in Glue has generally taken the Neo-Davidsonian approach of Parsons (1990), in which verbs (and other predicates that take event-arguments) denote functions from events to truth values, such that the arguments of the verb are actually modifiers of the event variable. For example, the sentences in (36) would receive an interpretation like (37), whereas the sentence in (38) would receive an interpretation like (39).[22]

(36) a. Sam hugged Max.

b. Max was hugged by Sam.

(37) $\exists e.hug(e) \wedge agent(e) = sam \wedge patient(e) = max$

(38) Max was hugged.

(39) $\exists e \exists x.hug(e) \wedge agent(e) = x \wedge patient(e) = max$

It can be observed from (37) and (39) that the event variable is eventually existentially closed. This is a standard assumption in event semantics.

Event semantics is a natural meaning language for Glue Semantics, because the event variable permits a highly factorized semantics, using LFG's *template* language (Dalrymple et al. 2004), which is designed to allow generalizations to

[21] Some of this work assumes some version of event semantics (sketched in Section 6.4). However, event semantics is not necessary for a basic treatment of tense, as I've illustrated here.

[22] It is also possible to treat verbs as generalized quantifiers over events (Champollion 2017; Coppock & Champollion 2020), but I'm not aware of any Glue work thus far that has taken this tack and it wouldn't make a difference to the sorts of simple cases sketched here.

be captured across grammatical elements, including meaning constructors. This in turn maximizes the analytic leverage offered by flexible composition and syntax/semantics non-isomorphism.

For example, the lexical entry for the verb *hugged* (again leaving tense aside) can capture its underlying semantic bivalence by encoding a dependency on a SUBJECT and OBJECT (as well as the event variable):[23]

(40) *hugged* $\lambda y \lambda x \lambda e.hug(e) \wedge agent(e) = x \wedge patient(e) = y$:
$obj \multimap subj \multimap event \multimap verb$

We can take advantage of pervasive syncretism in the English passive participle here and assume that the meaning constructor in (40) is associated with the past tense and passive participle alike.

We can then treat the passive voice as contributing a modificational meaning constructor that remaps the arguments, as in (41). I again associate this with an abstract formative to gloss over details of lexicalization (for a related proposal, see Findlay 2019: 185–186).[24]

(41) PASSIVE $\lambda P \lambda x \lambda y.P(x)(y)$: $(obj \multimap subj \multimap verb) \multimap$
$subj \multimap obl \multimap verb$

Note that this entry requires implication elimination on the *event* term in the verb's meaning constructor and then reintroduction of the term (for eventual existential binding of the corresponding variable) after the passive modifier has composed with the verb's meaning constructor. We will shortly add a second meaning constructor to the entry for PASSIVE, but this one suffices to capture the truth-conditional equivalence of (36a–b) (which is not to say that they are information-structurally equivalent).

The result of combining the meaning constructor in (41) with the one in (40) is passive *hugged*:

(42) *hugged*+PASSIVE $\lambda x \lambda y \lambda e.hug(e) \wedge agent(e) = y \wedge patient(e) = x$:
$subj \multimap obl \multimap event \multimap verb$

In other words, the passive voice modifies the meaning of *hugged* such that the passive subject corresponds to the logical object (the patient in this case) and

[23]It has been common in Glue work on event semantics to use e, e', e'', etc., as variables over events, but a common convention in event semantics more generally is to use v, v', v'', etc. (e.g., Champollion 2017; Coppock & Champollion 2020).

[24]Note that the treatment sketched here uses mnemonics for f-structure grammatical functions, like *subj*, in the Glue terms. However, actual Glue work in this vein uses Glue terms defined with respect to argument structure, as sketched in the next section. See Asudeh & Giorgolo (2012: 75–76) and Asudeh et al. (2014: 77ff.) for further details.

passive *by*-phrase corresponds to the logical subject (the agent in this case).[25] Figure 6 shows the proof for (36b). The reader can verify that the result is the same interpretation as that of (36a). The interpretation for (36a,b) is shown in (37) above.

But what of the short passive in (38)? Here we can leverage optionality and the properties of resource-sensitive composition and syntax/semantics non-isomorphism to naturally extend the analysis. We simply add an optional meaning constructor to (41), such that the revised lexical entry is as follows:

(43) PASSIVE $\lambda P \lambda y \lambda x.P(x)(y) : (obj \multimap subj \multimap verb) \multimap subj \multimap obl \multimap verb)$
$(\lambda P \exists x.P(x) : (obl \multimap verb) \multimap verb)$

The optional entry allows the passive to also contribute a second meaning constructor that existentially binds the subject argument. If there is an actual subject resource, though, as in the long passive in (36b), resource sensitivity ensures that the optional meaning constructor cannot be used, because then the actual subject resource would go unused. Figure 7 shows the proof for (36b). The reader can verify that the result is the same interpretation as that of (38), shown in (39) above.

The use of event semantics in LFG+Glue has become especially common in a thread of work on argument structure, the topic that we turn to next.

6.5 Argument structure

There is a prominent strand of work in Glue Semantics on argument structure and mapping theory (i.e., the realization of underlying arguments in the syntax). Representative work in this vein includes Arnold & Sadler (2013), Asudeh & Giorgolo (2012), Asudeh et al. (2014), Asudeh (2021), Findlay (2014, 2016, 2020), Lowe (2016, 2019), Lowe & Birahimani (2019), Przepiórkowski (2017), and Lovestrand (2020).

However, before turning to the Glue approach to argument structure, it is worth presenting some of the background that led to it, because it highlights another issue in Glue Semantics that has concerned some researchers. The substance of the worry can be straightforwardly summarized: What are the identity conditions for empty semantic structures? In other words, if a semantic structure is an attribute value matrix of some kind, as assumed from quite early on

[25] I have implicitly assumed in my choice of mnemonic, *obl*, that the *by*-phrase is an OBLIQUE, but nothing hinges on this. The same term works if it is treated as an ADJUNCT, although its mnemonic function is obscured.

hugged

$$\lambda y\lambda x\lambda e.hug(e) \wedge agent(e) = x \wedge patient(e) = y : obj \multimap subj \multimap event \multimap verb \qquad [u : obj]^1$$

$$\multimap_{\mathcal{E}}$$

$$\lambda x\lambda e.hug(e) \wedge agent(e) = x \wedge patient(e) = u : subj \multimap event \multimap verb \qquad [v : subj]^2$$

$$\multimap_{\mathcal{E}}$$

$$\lambda e.hug(e) \wedge agent(e) = v \wedge patient(e) = u : event \multimap verb \qquad [e' : event]^3$$

$$\multimap_{\mathcal{E}}$$

$$hug(e') \wedge agent(e') = v \wedge patient(e') = u : verb$$

$$\multimap_{\mathcal{I},2}$$

$$\lambda v.hug(e') \wedge agent(e') = v \wedge patient(e') = u : subj \multimap verb$$

$$\multimap_{\mathcal{I},1}$$

$$\lambda u\lambda v.hug(e') \wedge agent(e') = v \wedge patient(e') = u : obj \multimap subj \multimap verb$$

PASSIVE

$$\lambda P\lambda x\lambda y.P(x)(y) : (obj \multimap subj \multimap verb) \multimap subj \multimap obl \multimap verb$$

$$\multimap_{\mathcal{E}}$$

$$(\lambda P\lambda x\lambda y.P(x)(y))(\lambda u\lambda v.hug(e') \wedge agent(e') = v \wedge patient(e') = u) : subj \multimap obl \multimap verb$$

$$\Rightarrow_{\beta}$$

$$\lambda x\lambda y.(\lambda u\lambda v.hug(e') \wedge agent(e') = v \wedge patient(e') = u)(x)(y) : subj \multimap obl \multimap verb$$

$$\Rightarrow_{\beta}$$

$$\lambda x\lambda y.(\lambda v\lambda e'.hug(e') \wedge agent(e') = v \wedge patient(e') = x)(y) : subj \multimap obl \multimap verb$$

$$\Rightarrow_{\beta}$$

$$\lambda x\lambda y\lambda e'.hug(e') \wedge agent(e') = y \wedge patient(e') = x : subj \multimap obl \multimap verb$$

Max

$$max : subj$$

$$\multimap_{\mathcal{E}}$$

$$\lambda y.hug(e') \wedge agent(e') = y \wedge patient(e') = max : obl \multimap verb$$

by Sam

$$sam : obl$$

$$\multimap_{\mathcal{E}}$$

$$hug(e') \wedge agent(e') = sam \wedge patient(e') = max : verb$$

$$\multimap_{\mathcal{I},3}$$

$$\lambda e'.hug(e') \wedge agent(e') = sam \wedge patient(e') = max : event \multimap verb$$

$$\Rightarrow_{\alpha}$$

$$\lambda e.hug(e) \wedge agent(e) = sam \wedge patient(e) = max : event \multimap verb$$

$$\exists_{event}$$

$$\exists e.hug(e) \wedge agent(e) = sam \wedge patient(e) = max : verb$$

Figure 6: Proof for passive (36b), *Max was hugged by Sam*

hugged

$$\lambda y\lambda x\lambda e.hug(e) \wedge agent(e) = x \wedge patient(e) = y : obj \multimap subj \multimap event \multimap verb \qquad [u : obj]^1$$

$$\multimap_{\mathcal{E}}$$

$$\lambda x\lambda e.hug(e) \wedge agent(e) = x \wedge patient(e) = u : subj \multimap event \multimap verb \qquad [v : subj]^2$$

$$\multimap_{\mathcal{E}}$$

$$\lambda e.hug(e) \wedge agent(e) = v \wedge patient(e) = u : event \multimap verb \qquad [e' : event]^3$$

$$\multimap_{\mathcal{E}}$$

$$hug(e') \wedge agent(e') = v \wedge patient(e') = u : verb$$

$$\multimap_{\mathcal{I},2}$$

$$\lambda v.hug(e') \wedge agent(e') = v \wedge patient(e') = u : subj \multimap verb$$

$$\multimap_{\mathcal{I},1}$$

$$\lambda u\lambda v.hug(e') \wedge agent(e') = v \wedge patient(e') = u : obj \multimap subj \multimap verb$$

PASSIVE

$$\lambda P\lambda x\lambda y.P(x)(y) : (obj \multimap subj \multimap verb) \multimap subj \multimap obl \multimap verb$$

$$\multimap_{\mathcal{E}}$$

$$(\lambda P\lambda x\lambda y.P(x)(y))(\lambda u\lambda v.hug(e') \wedge agent(e') = v \wedge patient(e') = u) : subj \multimap obl \multimap verb$$

$$\Rightarrow_{\beta}$$

$$\lambda x\lambda y.(\lambda u\lambda v.hug(e') \wedge agent(e') = v \wedge patient(e') = u)(x)(y) : subj \multimap obl \multimap verb$$

$$\Rightarrow_{\beta}$$

$$\lambda x\lambda y.(\lambda v\lambda e'.hug(e') \wedge agent(e') = v \wedge patient(e') = x)(y) : subj \multimap obl \multimap verb$$

$$\Rightarrow_{\beta}$$

$$\lambda x\lambda y\lambda e'.hug(e') \wedge agent(e') = y \wedge patient(e') = x : subj \multimap obl \multimap verb$$

Max

$$max : subj$$

$$\multimap_{\mathcal{E}}$$

$$\lambda y.hug(e') \wedge agent(e') = y \wedge patient(e') = max : obl \multimap verb$$

PASSIVE

$$\lambda P\exists x.P(x) : (obl \multimap verb) \multimap verb$$

$$\multimap_{\mathcal{E}}$$

$$(\lambda P\exists x.P(x))(\lambda y.hug(e') \wedge agent(e') = y \wedge patient(e') = max) : verb$$

$$\Rightarrow_{\beta}$$

$$\exists x.(\lambda y.hug(e') \wedge agent(e') = y \wedge patient(e') = max)(x) : verb$$

$$\Rightarrow_{\beta}$$

$$\exists x.hug(e') \wedge agent(e') = x \wedge patient(e') = max : verb$$

$$\multimap_{\mathcal{I},3}$$

$$\lambda e'\exists x.hug(e') \wedge agent(e') = x \wedge patient(e') = max : event \multimap verb$$

$$\Rightarrow_{\alpha}$$

$$\lambda e\exists x.hug(e) \wedge agent(e) = x \wedge patient(e) = max : event \multimap verb$$

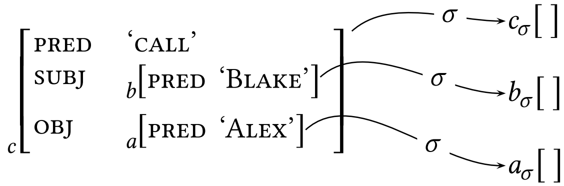
$$\exists_{event}$$

$$\exists e\exists x.hug(e) \wedge agent(e) = x \wedge patient(e) = max : event \multimap verb$$

Figure 7: Proof for short passive (38), *Max was hugged*

in the development of Glue Semantics (Dalrymple, Gupta, et al. 1999; Dalrymple 2001), how can there be *distinct* empty s-structures, since an empty AVM seems to correspond to the empty set, which is unique (Kokkonidis 2008; Findlay 2021)?

(44)



One possible solution to the empty s-structure problem is to make the labelling part of the definition of the structure. In other words, if a standard attribute-value matrix is a finite set of attribute-value pairs (see, e.g., Bresnan et al. 2016: 44), then let us define an s-structure as a finite set of pairs, where the first member of each pair is a string (a unique label) and the second member of each pair is a (possibly empty) AVM. In that case, it's clear that the s-structure $\{\langle a_\sigma, \varnothing\rangle\}$ does not equal the s-structure $\{\langle b_\sigma, \varnothing\rangle\}$, even if both of them have the empty AVM as their second coordinate.

However, another issue with the sort of s-structure in (44) is that it's really not a structure at all, since the parts are not connected. In other words, what we have in (44) is really *three* s-structures, not a single one. This does not make a substantive difference to the kinds of proofs one can do in Glue Semantics, but it is a bit strange from a general LFG-theoretic perspective, as we would expect all the modules in the Correspondence Architecture to be structures and all of the ones that have been proposed, aside from the version of s-structure above, indeed are structures.

Asudeh & Giorgolo (2012) solve this last problem by offering a connected s-structure that also fulfills the role of a(rgument)-structure (Butt et al. 1997) in the Correspondence Architecture. Not only does this eliminate the need for a-structure as a separate module in the architecture, it also relates argument structure and mapping theory more strongly to compositional semantics, as the locus for both is now s-structure. Figure 8 shows the Asudeh & Giorgolo (2012) analysis for (45).

(45) Kim ate at noon.

The verb *ate* is semantically bivalent, since it entails that there is something that has been eaten, but it can nevertheless be syntactically intransitive (Asudeh & Giorgolo 2012: 71). This is reflected in the analysis in Figure 8. There is no OBJECT in the f-structure, but there are two arguments in the connected s-structure, which also serves as a representation of argument structure.

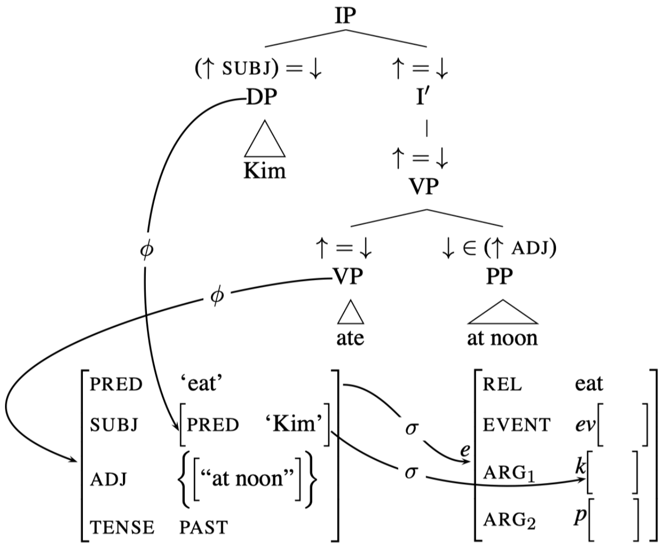


Figure 8: *Kim ate at noon* (Asudeh & Giorgolo 2012: 72; used with permission)

The solution to the syntax/semantics mismatch for the verb *ate* is to allow the verb itself to contribute an optional second meaning constructor that existentially closes the dependency on the second argument:[26]

(46) *ate* $\lambda y \lambda x \lambda e.eat(e) \wedge agent(e) = x \wedge patient(e) = y$:
$obj \multimap subj \multimap event \multimap verb$
$(\ \lambda P \exists x.P(x)\ :\ (obj \multimap verb) \multimap verb\ )$

This treatment is similar to the one for the passive in (43). Note that this is a simplification of the actual approach in Asudeh & Giorgolo (2012) and Asudeh et al. (2014), because in those approaches the Glue logic terms are defined using ARG features at s-structure, which allows the analysis to more naturally interact properly with argument alternations.

6.6 Multiword expressions

Multiword expressions (MWEs) are a challenge to a lexicalist theory like LFG, because they show a mixture of idiomaticity and productivity in both their syntax

[26]Intransitive uses of semantically bivalent verbs also trigger presuppositions about the implicit argument (Fillmore 1986); e.g., *Kim ate at noon* presupposes that what Kim ate is food (for Kim). I do not attempt to model this here, but see Asudeh & Giorgolo (2012) and Asudeh (2021) for some further discussion.

and semantics (Findlay 2019: ch. 1). On the one hand, we find expressions like *by and large* which are idiosyncratic in both their syntax (apparently a coordination of a preposition and an adjective) and semantics (the expression means something similar to the adverb *mostly*, but this can't be compositionally obtained from the usual meanings of its parts). On the other hand, we find expressions like *spill the beans*, which are syntactically unexceptional and possibly yield to a kind of transpositional semantic analysis in which $[\![spill]\!] = [\![reveal]\!]$ and $[\![the\ beans]\!] = [\![the\ secret]\!]$. Nevertheless, even with this MWE we see evidence of particular syntactic and semantic restrictions. For example, the object is necessarily the definite plural *beans* and other forms are either excluded entirely (e.g., #*a bean* or #*the peas*) or else seem at best like metalinguistic word-play (e.g., *the legumes*).

In short, MWEs are challenging because they are like words in the sense that they seem to be lexically stored expressions but are like phrases in having syntactic parts and, in some cases, these parts seem to be visible to syntactic operations. For example, in *It's too late: the beans have already been spilled*, the MWE has been passivized and one part is modified by an adverbial. For a lexicalist theory, simultaneously capturing these lexical and non-lexical properties of MWEs is difficult. Indeed, in order to account for this mixture of lexical and syntactic properties, Findlay (2019) replaces the c-structural part of standard LFG with Tree-Adjoining Grammar (TAG; Joshi et al. 1975; Abeillé & Rambow 2000), which allows expressions to be associated with trees in the lexicon, rather than with a simple category. TAG allows these trees to then be inserted or adjoined in the phrasal syntax. Findlay (2019) calls the resulting theory *Lexicalised LFG*, in a nod to Lexicalized TAG (Schabes et al. 1988), because it allows lexicalization of syntactic structures as TAG trees while maintaining LFG's standard separate level of f-structure and a mapping between the TAG-based c-structures and the f-structures.

No matter how one captures the syntax of MWEs, the syntax/semantics non-isomorphism of Glue Semantics naturally captures their syntax/semantics mismatches and idiomaticity. For example, Figure 9 shows Findlay's (2019) *lexical* entry for *by and large*. It is an adjunct tree, since this is a modifier. The meaning of *by-and-large* is captured by the call to a template, @By-And-Large-Meaning, but we can simplify things as in (47).

(47) *by and large* $\lambda P \lambda x.mostly(P(x)) : (subj \multimap verb) \multimap (subj \multimap verb)$

This is a relatively straightforward example. For more complex examples, see Findlay (2019).

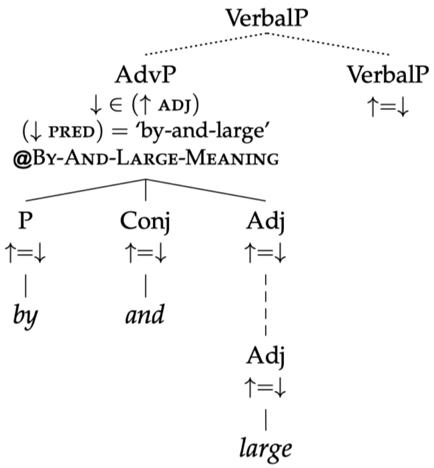


Figure 9: Lexical entry for *by and large* (Findlay 2019: 265; used with permission)

In more recent work, Findlay (2021) has adopted a different formalization of Glue in order to account for MWEs that show form flexibility as long as some kind of core meaning is maintained, like in the following:

(48) a kick up the bum/backside/bottom/buttocks/ass/heinie/keister/booty/…

In this MWE, any word that denotes ⟦*bum*⟧ would seem to do, no matter its form, but anything that doesn't denote ⟦*bum*⟧ doesn't seem to have the idiomatic 'motivational' reading (e.g., #*crotch*).

6.7 Anaphora

Anaphora has been a topic of long-standing interest in Glue Semantics. A recent LFG+Glue treatment and overview of previous literature is given in Dalrymple et al. (2019: ch. 14). Their treatment is a fairly sophisticated one that builds on recent work by Haug (2014) and Dalrymple et al. (2018). Here I present a simpler overview that summarizes the approach in Dalrymple, Lamping, Pereira, et al. (1997) and Asudeh (2004, 2012).

The property of flexible composition means that Glue can provide a variable-free treatment of anaphora, but without requiring that the anaphoric dependency be passed through all intervening material between the anaphor and its antecedent (in the intra-sentential case), as in non-commutative Categorial Grammar approaches (Jacobson 1999 et seq.). The simplest way to capture this would be through an implicational meaning constructor as in (49). I again associate the meaning constructor with an abstract formative to leave aside other aspects of particular personal pronouns, such as person, number, gender.

(49) ANAPHOR $\lambda y.y : antecedent \multimap anaphor$

However, there is a problem with a treatment this simple, because of resource-sensitive composition. If the anaphor consumes the antecedent resource, then the antecedent would no longer be available for composition. This means that whatever function takes the antecedent's denotation as an actual argument can no longer have its resource-sensitive compositional requirements satisfied. There would be no valid proof.

In order to remedy this, a simple solution is to slightly expand the fragment of linear logic that serves as the Glue logic. We add the *multiplicative conjunction* operator, ⊗, which does tensor/pair formation. The meaning constructor in (49) is then revised as follows:

(50) ANAPHOR $\lambda y.y \times y : antecedent \multimap (antecedent \otimes anaphor)$

The anaphor is still a function on its antecedent, but it now returns both its own resource and the antecedent resource.

On this sort of approach to anaphora, the multiplicative conjunction ⊗ is only ever introduced lexically (much as is the linear logic universal for scope points; see above). Therefore we just need to add the elimination rule for this connective, which is the following:

(51) Structured functional application:
Multiplicative conjunction elimination pairwise substitution

$$\dfrac{\begin{array}{ccc} & [\beta : A]^1 & [\gamma : B]^2 \\ \vdots & \vdots & \vdots \\ \alpha : A \otimes B & & \delta : C \end{array}}{\text{let } \alpha \text{ be } \beta \times \gamma \text{ in } \delta : C} \otimes_{\mathcal{E},1,2}$$

The *let* type constructor performs pairwise substitution for the variables x, y in the result.

This is still quite abstract, so it is probably helpful to look at example (52) and its accompanying proof in Figure 10, both from Asudeh (2012: 84).[27]

[27]Note that I have left out the $\multimap_{\mathcal{E}}$ annotations in the proof to reduce clutter. Also, the following mnemonic Glue terms are used: t for the term contributed by *Thora* (which is both the antecedent of the pronoun and the subject of the sentence), p for pronoun, g for *giggle*, and s for *said*.

(52) Thora said she giggled.

$$\dfrac{\dfrac{\begin{matrix}\textbf{\textit{Thora}}\\ thora : \\ t\end{matrix} \quad \begin{matrix}\textbf{\textit{she}}\\ \lambda z.z \times z : \\ t \multimap (t \otimes p)\end{matrix}}{thora \times thora : t \otimes p}\multimap_{\mathcal{E}} \quad \dfrac{\dfrac{[x : t]^1 \quad \begin{matrix}\textbf{\textit{said}}\\ \lambda u \lambda q.say(u,q) : \\ t \multimap g \multimap s\end{matrix}}{\lambda q.say(x,q) : g \multimap s}\multimap_{\mathcal{E}} \quad \dfrac{[y : p]^2 \quad \begin{matrix}\textbf{\textit{giggled}}\\ \lambda x.giggle(x) : \\ p \multimap g\end{matrix}}{giggle(y) : g}\multimap_{\mathcal{E}}}{say(x, giggle(y)) : s}\multimap_{\mathcal{E}}}{\dfrac{\text{let } thora \times thora \text{ be } x \times y \text{ in } say(x, giggle(y)) : s}{say(thora, giggle(thora)) : s}\Rightarrow_{\beta}}\otimes_{\mathcal{E},1,2}$$

Figure 10: Proof for intra-sentential anaphoric reading of (52), *Thora said she giggled*

For a fuller treatment of anaphora that extends to inter-sentential cases, see Haug & Dalrymple (2020) and Dalrymple & Haug (2022).

7 Conclusion

Glue Semantics is a general framework for semantic composition and the syntax–semantics interface. The focus here has been on Glue for LFG, typically known as LFG+Glue. Four key properties of Glue Semantics are resource-sensitive composition, flexible composition, autonomy of syntax, and syntax/semantics non-isomorphism (Asudeh 2022: 324). Analyses in Glue Semantics are highly constrained by the resource logic *linear logic*, a fragment of which serves as the Glue logic for semantic composition. Although resource-sensitive composition constrains semantic composition, it allows composition to be commutative. This yields the property of flexible composition: The logical syntax of composition is not identical to the structural syntax. From this we can derive the property of autonomy of syntax: Syntax and semantics are separate levels. From this we lastly derive the property of syntax/semantics non-isomorphism: Whatever the basic elements of structural syntax are taken to be (words in the case of standard LFG), these elements may make multiple or no contributions to semantic composition. The best source for further details about Glue analyses of particular phenomena and further Glue references is Dalrymple et al. (2019). However, I've listed a representative sample of Glue work by topic in the appendix.

Acknowledgements

Many thanks to Mary Dalrymple for editing this volume, but also for many useful discussions about Glue Semantics over the last more than two decades. Similarly, thank you to Avery Andrews, Doug Arnold, Dick Crouch, Jamie Findlay, Gianluca Giorgolo, Matthew Gotham, Dag Haug, Miltos Kokkonidis, John Lowe, Agnieszka Patejuk, Adam Przepiórkowski, Louisa Sadler, and Vijay Saraswat. Last but not least, a big thank you to the three anonymous reviewers. All remaining errors and misconceptions are my own.

List of Glue work by topic

Here is a representative sample of work in Glue Semantics, organized alphabetically by topic:[28]

- Anaphora
  Dalrymple, Lamping, Pereira, et al. (1997), Asudeh (2004, 2005b, 2012), Bary & Haug (2011), Belyaev & Haug (2014), Dalrymple et al. (2019: ch. 14), Haug & Dalrymple (2020), Dalrymple & Haug (2022)

- Argument structure and argument realization
  Asudeh & Giorgolo (2012), Asudeh et al. (2014), Asudeh (2021), Arnold & Sadler (2013), Findlay (2014, 2016, 2020), Lowe (2016, 2019), Lowe & Birahimani (2019), Przepiórkowski (2017), Lovestrand (2020)

- Category theory for natural language semantics
  Giorgolo & Asudeh (2012a,b), Giorgolo & Asudeh (2014a,b), Asudeh & Giorgolo (2016, 2020)

- Complex predicates
  Andrews (2007, 2018), Lowe (2016, 2019), Lowe & Birahimani (2019), Lovestrand (2020)

- Computational applications and tools (open source)[29]
  Crouch et al. (1986), Lev (2007), Meßmer & Zymla (2018), Dalrymple et al. (2020), Zymla (2021a,b,c)

- Concomitance
  Haug (2009)

- Constructions
  Asudeh et al. (2008, 2013), Asudeh & Toivonen (2014)

[28] My apologies to anyone whose work I have inadvertently omitted.
[29] Zymla's (2021c) tool goes with the XLE tools for computational implementation and testing of LFG grammars (Crouch, Dalrymple, Kaplan, King, Maxwell & Newman 2011).

- Control/equi and raising
  Asudeh (2005a), Haug (2013), Dalrymple et al. (2019: ch. 15)

- Conventional implicature
  Asudeh (2004: ch. 4), Potts (2005), Arnold & Sadler (2010, 2011), Giorgolo & Asudeh (2012a)

- Coordination
  Kehler et al. (1999), Asudeh & Crouch (2002a), Dalrymple et al. (2019: ch. 16)

- Copy raising
  Asudeh (2004, 2012), Asudeh & Toivonen (2007, 2012)

- Distance distributivity
  Przepiórkowski (2014a,b, 2015)

- Dynamic semantics
  Crouch & van Genabith (1999), van Genabith & Crouch (1999), Dalrymple et al. (2019: ch. 14)

- Event semantics
  Fry (2005), Asudeh & Giorgolo (2012), Asudeh & Toivonen (2012), Asudeh et al. (2013), Asudeh et al. (2014)

- Evidentiality
  Asudeh & Toivonen (2017)

- Formal foundations
  Dalrymple, Gupta, et al. (1999), Dalrymple, Lamping, Pereira, et al. (1999), Dalrymple, Lamping, Pereira, et al. (1997); Asudeh (2004, 2012: ch. 5) Kokkonidis (2008), Andrews (2008, 2010), Findlay (2021))

- Fragments
  Asudeh (2012: ch. 11)

- Idioms and multiword expressions
  Findlay (2019, 2021)

- Incorporation
  Asudeh (2007), Baker et al. (2010)

- Information structure
  Dalrymple & Nikolaeva (2011), Mycock (2006), Morrison (2017)

- Intensionality
  Dalrymple, Lamping, Pereira, et al. (1997), Asudeh & Toivonen (2012)

- Modification
  Dalrymple et al. (1993), Dalrymple, Lamping, Pereira, et al. (1999), Dalrymple (2001), Asudeh & Crouch (2002b), Andrews (2018), Dalrymple et al. (2019: ch. 13)

- Negative polarity items
  Fry (1999a)

- Perception verbs
  Asudeh & Toivonen (2007, 2012), Asudeh (2012), Camilleri et al. (2014)

- Predication
  Dalrymple (2001), Asudeh & Crouch (2002b), Dalrymple et al. (2019: ch. 13)

- Quantification and scope
  Dalrymple, Lamping, Pereira, et al. (1997), Dalrymple et al. (2019: ch. 8)

- Relational nouns
  Asudeh (2005b)

- Resumptive pronouns
  Asudeh (2004, 2005b, 2011, 2012), Camilleri & Sadler (2011)

- Split nominals
  Kuhn (2001)

- Tense and aspect
  Haug (2008), Bary & Haug (2011), Lowe (2014, 2015), Belyaev (2020)

- Unbounded dependencies
  Asudeh (2012), Dalrymple et al. (2019: ch. 17)

References

Abeillé, Anne & Owen Rambow (eds.). 2000. *Tree Adjoining Grammars: Formalisms, linguistic analysis and processing*. Stanford: CSLI Publications.

Andrews, Avery D. 2007. Projections and glue for clause union complex predicates. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '07 conference*, 44–65. Stanford: CSLI Publications.

Andrews, Avery D. 2008. The role of PRED in LFG+Glue. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '08 conference*, 46–67. Stanford: CSLI Publications.

Andrews, Avery D. 2010. Propositional glue and the architecture of LFG. *Linguistics and Philosophy* 33(3). 141–170.

Andrews, Avery D. 2018. Sets, heads and spreading in LFG. *Journal of Language Modelling* 6(1). 131–174. DOI: 10.15398/jlm.v6i1.175.

Arnold, Doug & Louisa Sadler. 2010. Pottsian LFG. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '10 conference*, 43–63. Stanford: CSLI Publications.

Arnold, Doug & Louisa Sadler. 2011. Resource splitting and reintegration with supplementals. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '11 conference*, 26–46. Stanford: CSLI Publications.

Arnold, Doug & Louisa Sadler. 2013. Affected experiencers and mixed semantics in LFG/Glue. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '12 conference*, 44–63. Stanford: CSLI Publications.

Asudeh, Ash. 2004. *Resumption as resource management*. Stanford: Stanford University. (Doctoral dissertation).

Asudeh, Ash. 2005a. Control and semantic resource sensitivity. *Journal of Linguistics* 41(3). 465–511. DOI: 10.1017/s0022226705003427.

Asudeh, Ash. 2005b. Relational nouns, pronouns, and resumption. *Linguistics and Philosophy* 28. 375–446. DOI: 10.1007/s10988-005-2656-7.

Asudeh, Ash. 2006. Direct compositionality and the architecture of LFG. In Miriam Butt, Mary Dalrymple & Tracy Holloway King (eds.), *Intelligent linguistic architectures: Variations on themes by Ronald M. Kaplan*, 363–387. Stanford: CSLI Publications.

Asudeh, Ash. 2007. Some notes on pseudo noun incorporation in Niuean. Unpublished manuscript, Carleton University. sas.rochester.edu/lin/sites/asudeh/pdf/niuean.pdf.

Asudeh, Ash. 2011. Towards a unified theory of resumption. In Alain Rouveret (ed.), *Resumptive pronouns at the interfaces*. Amsterdam: John Benjamins. DOI: 10.1075/lfab.5.03asu.

Asudeh, Ash. 2012. *The logic of pronominal resumption* (Oxford Studies in Theoretical Linguistics). Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780199206421.001.0001.

Asudeh, Ash. 2021. The unrealized and the unheard. In I Wayan Arka, Ash Asudeh & Tracy Holloway King (eds.), *Modular design of grammar: Linguistics on the edge*, 22–44. Oxford: Oxford University Press. DOI: 10.1093/oso/9780192844842.003.0003.

Asudeh, Ash. 2022. Glue semantics. *Annual Review of Linguistics* 8. 321–341. DOI: 10.1146/annurev-linguistics-032521-053835.

Asudeh, Ash & Richard Crouch. 2002a. Coordination and parallelism in Glue Semantics: Integrating discourse cohesion and the element constraint. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '02 conference*, 19–39. Stanford.

Asudeh, Ash & Richard Crouch. 2002b. Glue semantics for HPSG. In Frank Van Eynde, Lars Hellan & Dorothee Beermann (eds.), *Proceedings of the 8th international conference on Head-Driven Phrase Structure Grammar*, 1–19. Stanford: CSLI Publications. DOI: 10.21248/hpsg.2001.1.

Asudeh, Ash, Mary Dalrymple & Ida Toivonen. 2008. Constructions with lexical integrity: Templates as the lexicon–syntax interface. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '08 conference*, 68–88. Stanford: CSLI Publications.

Asudeh, Ash, Mary Dalrymple & Ida Toivonen. 2013. Constructions with Lexical Integrity. *Journal of Language Modelling* 1(1). 1–54. DOI: 10.15398/jlm.v1i1.56.

Asudeh, Ash & Gianluca Giorgolo. 2012. Flexible composition for optional and derived arguments. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '12 conference*, 64–84. Stanford: CSLI Publications.

Asudeh, Ash & Gianluca Giorgolo. 2016. Perspectives. *Semantics & Pragmatics* 9(21). 1–57. DOI: 10.3765/sp.9.21.

Asudeh, Ash & Gianluca Giorgolo. 2020. *Enriched meanings: natural language semantics with category theory*. Oxford: Oxford University Press. DOI: 10.1093/oso/9780198847854.001.0001.

Asudeh, Ash, Gianluca Giorgolo & Ida Toivonen. 2014. Meaning and valency. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '14 conference*, 68–88. Stanford: CSLI Publications.

Asudeh, Ash & Ida Toivonen. 2007. Copy raising and its consequences for perception reports. In Annie Zaenen, Jane Simpson, Tracy Holloway King, Jane Grimshaw, Joan Maling & Chris Manning (eds.), *Architectures, rules, and preferences: Variations on themes by Joan W. Bresnan*, 49–67. Stanford: CSLI Publications.

Asudeh, Ash & Ida Toivonen. 2012. Copy raising and perception. *Natural Language & Linguistic Theory* 30(2). 321–380. DOI: 10.1007/s11049-012-9168-2.

Asudeh, Ash & Ida Toivonen. 2014. *With* Lexical Integrity. *Theoretical Linguistics* 40(1–2). 175–186.

Asudeh, Ash & Ida Toivonen. 2015. Lexical-Functional Grammar. In Bernd Heine & Heiko Narrog (eds.), *The Oxford handbook of linguistic analysis*, 2nd edn., 373–406. Oxford: Oxford University Press. DOI: 10.1093/oxfordhb/9780199677078.013.0017.

Asudeh, Ash & Ida Toivonen. 2017. A modular approach to evidentiality. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '17 conference*, 45–65. Stanford: CSLI Publications.

Bach, Emmon W. 1976. An extension of classical transformational grammar. In Jerrold M. Sadock, David J. Dwyer, Seok C. Song & Emmon Bach (eds.), *Problems in linguistic metatheory: Proceedings of the 1976 conference*, 183–224. East Lansing, MI: Michigan State University.

Baker, Brett, Kate Horrack, Rachel Nordlinger & Louisa Sadler. 2010. Putting it all together: Agreement, incorporation, coordination and external possession in Wubuy (Australia). In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '10 conference*, 65–84. Stanford: CSLI Publications.

Baldridge, Jason. 2002. *Lexically specified derivational control in Combinatory Categorial Grammar*. University of Edinburgh. (Doctoral dissertation).

Barker, Chris & Pauline Jacobson (eds.). 2007. *Direct compositionality*. Oxford: Oxford University Press.

Barwise, Jon & Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4(2). 159–219.

Bary, Corien & Dag Haug. 2011. Temporal anaphora across and inside sentences: the function of participles. *Semantics & Pragmatics* 4(8). 1–56. DOI: 10.3765/sp.4.8.

Belyaev, Oleg. 2020. "Verbal case" in Ashti Dargwa. In Miriam Butt & Ida Toivonen (eds.), *Proceedings of the LFG '20 conference*, 26–46. Stanford: CSLI Publications.

Belyaev, Oleg & Dag Haug. 2014. Pronominal coreference in Ossetic correlatives and the syntax-semantics interface. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '14 conference*, 89–109. Stanford: CSLI Publications.

Bresnan, Joan, Ash Asudeh, Ida Toivonen & Stephen Wechsler. 2016. *Lexical-Functional Syntax*. 2nd edn. (Blackwell Textbooks in Linguistics 16). Malden, MA: Wiley-Blackwell.

Butt, Miriam, Mary Dalrymple & Anette Frank. 1997. An architecture for linking theory in LFG. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '97 conference*, 1–16. Stanford: CSLI Publications.

Camilleri, Maris, Shaimaa ElSadek & Louisa Sadler. 2014. Perceptual reports in (dialects of) Arabic. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '14 conference*, 179–199. Stanford: CSLI Publications.

Camilleri, Maris & Louisa Sadler. 2011. Restrictive relative clauses in Maltese and resumption. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '11 conference*, 110–130. Stanford: CSLI Publications.

Carpenter, Bob. 1997. *Type-logical semantics*. Cambridge, MA: The MIT Press. DOI: 10.7551/mitpress/6945.001.0001.

Champollion, Lucas. 2017. *Parts of a whole: distributivity as a bridge between aspect and measurement*. Oxford: Oxford University Press.

Chomsky, Noam. 1981. *Lectures on government and binding*. Dordrecht: Foris Publications. DOI: 10.1515/9783110884166.

Chomsky, Noam. 1982. *Some concepts and consequences of the theory of government and binding* (Linguistic Inquiry Monographs). Cambridge, MA: The MIT Press.

Chomsky, Noam. 1986. *Knowledge of language: its nature, origin, and use*. New York: Praeger.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: The MIT Press. DOI: 10.7551/mitpress/9780262527347.001.0001.

Coppock, Elizabeth & Lucas Champollion. 2020. Invitation to formal semantics. Unpublished manuscript, Boston University and New York University. http://eecoppock.info/bootcamp/semantics-boot-camp.pdf.

Crouch, Richard, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III & Paula Newman. 2011. *XLE Documentation*. Xerox Palo Alto Research Center. Palo Alto, CA. https://ling.sprachwiss.uni-konstanz.de/pages/xle/doc/xle_toc.html.

Crouch, Richard, Anette Frank & Josef van Genabith. 1986. Glue, underspecification and translation. In Harry Bunt, Reinhard Muskens & Elias Thijsse (eds.), *Computing meaning*, vol. 2 (Studies in Linguistics and Philosophy 77), 165–184. Dordrecht: Springer.

Crouch, Richard & Josef van Genabith. 1999. Context change, underspecification, and the structure of Glue Language derivations. In Mary Dalrymple (ed.), *Semantics and syntax in Lexical Functional Grammar: The resource logic approach* (Language, Speech, and Communication), 117–189. Cambridge, MA: The MIT Press.

Crouch, Richard & Joseph van Genabith. 2000. Linear logic for linguists. Lecture notes, ESSLLI 2000.

Curry, Haskell B. & Robert Feys. 1958. *Combinatory logic*. Vol. 1. Amsterdam: North-Holland.

Dalrymple, Mary (ed.). 1999. *Semantics and syntax in Lexical Functional Grammar: The resource logic approach* (Language, Speech, and Communication). Cambridge, MA: The MIT Press. DOI: 10.7551/mitpress/6169.001.0001.

Dalrymple, Mary. 2001. *Lexical Functional Grammar* (Syntax and Semantics 34). New York: Academic Press. DOI: 10.1163/9781849500104.

Dalrymple, Mary, Vaneet Gupta, John Lamping & Vijay Saraswat. 1999. Relating resource-based semantics to categorial semantics. In Mary Dalrymple (ed.), *Semantics and syntax in Lexical Functional Grammar: The resource logic approach* (Language, Speech, and Communication), 261–280. Earlier version published as Dalrymple, Gupta, Lamping & Saraswat (1997). Cambridge, MA: The MIT Press.

Dalrymple, Mary, Vineet Gupta, John Lamping & Vijay Saraswat. 1997. Relating resource-based semantics to categorial semantics. In *Proceedings of the Fifth Meeting on Mathematics of Language (MOL5)*, vol. 97-02 (DFKI Documents (D)), 22–29. Revised version published in Dalrymple (1999: 261–280).

Dalrymple, Mary & Dag Haug. 2022. Reciprocal scope at the syntax-semantics interface. In Miriam Butt, Jamie Y. Findlay & Ida Toivonen (eds.), *Proceedings of the LFG '22 conference*. Stanford: CSLI Publications.

Dalrymple, Mary, Dag T. Haug & John J. Lowe. 2018. Integrating LFG's binding theory with PCDRT. *Journal of Language Modelling* 6(1). 87–129. DOI: 10.15398/jlm.v6i1.204.

Dalrymple, Mary, Ronald M. Kaplan & Tracy Holloway King. 2004. Linguistic generalizations over descriptions. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '04 conference*, 199–208. Stanford: CSLI Publications.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell III & Annie Zaenen (eds.). 1995. *Formal issues in Lexical-Functional Grammar*. Stanford: CSLI Publications.

Dalrymple, Mary, John Lamping, Fernando Pereira & Vijay Saraswat. 1995. Linear logic for meaning assembly. In Suresh Manandhar, Gabriel Pereira Lopes & Werner Nutt (eds.), *Proceedings of Computational Logic for Natural Language Processing*. Edinburgh.

Dalrymple, Mary, John Lamping, Fernando Pereira & Vijay Saraswat. 1997. Quantification, anaphora, and intensionality. *Journal of Logic, Language, and Information* 6(3). Revised version published in Dalrymple (1999: 39–89), 219–273.

Dalrymple, Mary, John Lamping, Fernando Pereira & Vijay Saraswat. 1999. Overview and introduction. In Mary Dalrymple (ed.), *Semantics and syntax in Lexical Functional Grammar: The resource logic approach*. Cambridge, MA: The MIT Press.

Dalrymple, Mary, John Lamping & Vijay Saraswat. 1993. LFG semantics via constraints. In *Proceedings of the 6th conference of the European chapter of the ACL*

(EACL 1993), 97–105. Association for Computational Linguistics. DOI: 10.3115/976744.976757.

Dalrymple, Mary, John J. Lowe & Louise Mycock. 2019. *The Oxford reference guide to Lexical Functional Grammar*. Oxford: Oxford University Press. DOI: 10.1093/oso/9780198733300.001.0001.

Dalrymple, Mary & Irina Nikolaeva. 2011. *Objects and information structure* (Cambridge Studies in Linguistics). Cambridge, UK: Cambridge University Press. DOI: 10.1017/cbo9780511993473.

Dalrymple, Mary, Agnieszka Patejuk & Mark-Matthias Zymla. 2020. XLE+Glue – A new tool for integrating semantic analysis in XLE. In Miriam Butt & Ida Toivonen (eds.), *Proceedings of the LFG '20 conference*, 89–108. Stanford: CSLI Publications.

Davidson, Donald. 1967. Truth and meaning. *Synthèse*. 304–323.

de Groote, Philippe (ed.). 1995. *The Curry-Howard isomorphism* (Cahiers du Centre de Logique 8). Louvain-la-Neuve: Academia.

Dowty, David R. 1979. *Word meaning and Montague grammar: The semantics of verbs and times in generative semantics and in Montague's PTQ* (Synthese Language Library 7). Dordrecht: Reidel.

Emms, Martin. 1990. Polymorphic quantifiers. In *Studies in Categorial Grammar*, 65–111. Edinburgh: Centre for Cognitive Science.

Emms, Martin. 1992. *Logical ambiguity*. University of Edinburgh. (Doctoral dissertation).

Fenstad, Jens Erik, Per-Kristian Halvorsen, Tore Langhold & Johan van Benthem. 1987. *Situations, language and logic*. Dordrecht: Reidel. DOI: 10.1007/978-94-009-1335-6.

Fillmore, Charles J. 1986. Pragmatically controlled zero anaphora. In *Proceedings of the twelfth annual meeting of the Berkeley Linguistics Society*, 95–107. Berkeley: Berkeley Linguistics Society. DOI: 10.3765/bls.v12i0.1866.

Findlay, Jamie Y. 2014. *The prepositional passive: a Lexical Functional account*. University of Oxford. (M.Phil. Thesis).

Findlay, Jamie Y. 2016. Mapping theory without argument structure. *Journal of Language Modelling* 4(2). 293–338. DOI: 10.15398/jlm.v4i2.171.

Findlay, Jamie Y. 2019. *Multiword expressions and the lexicon*. Oxford: University of Oxford. (D.Phil. Thesis).

Findlay, Jamie Y. 2020. Mapping Theory and the anatomy of a verbal lexical entry. In Miriam Butt & Ida Toivonen (eds.), *Proceedings of the LFG '20 conference*, 127–147. Stanford: CSLI Publications.

Findlay, Jamie Y. 2021. Meaning in LFG. In I Wayan Arka, Ash Asudeh & Tracy Holloway King (eds.), *Modular design of grammar: Linguistics on the edge*, 340–374. Oxford: Oxford University Press. DOI: 10.1093/oso/9780192844842.003.0020.

Fox, Danny. 2000. *Economy and semantic interpretation*. Cambridge, MA: The MIT Press.

Frank, Anette & Josef van Genabith. 2001. GlueTag: Linear logic based semantics for LTAG – and what it teaches us about LFG and LTAG. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '01 conference*, 104–126. Stanford: CSLI Publications.

Fry, John. 1999a. Proof nets and negative polarity licensing. In Mary Dalrymple (ed.), *Semantics and syntax in Lexical Functional Grammar: The resource logic approach* (Language, Speech, and Communication), 91–116. Cambridge, MA: The MIT Press.

Fry, John. 1999b. Resource-logical event semantics for LFG. Presented at the LFG '99 Conference.

Fry, John. 2005. Resource-logical event semantics for LFG. Unpublished manuscript, Stanford University.

Giorgolo, Gianluca & Ash Asudeh. 2012a. $\langle M, \eta, \star \rangle$ Monads for Conventional Implicatures. In Ana Aguilar Guevara, Anna Chernilovskaya & Rick Nouwen (eds.), *Proceedings of Sinn und Bedeutung 16*, vol. 1, 265–278. MIT Working Papers in Linguistics.

Giorgolo, Gianluca & Ash Asudeh. 2012b. Missing resources in a resource-sensitive semantics. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '12 conference*, 219–239. Stanford: CSLI Publications.

Giorgolo, Gianluca & Ash Asudeh. 2014a. Monads as a solution for generalized opacity. In *Proceedings of the EACL 2014 workshop on type theory and natural language semantics (TTNLS)*, 19–27. Stroudsburg, PA: Association for Computational Linguistics. DOI: 10.3115/v1/w14-1403.

Giorgolo, Gianluca & Ash Asudeh. 2014b. One semiring to rule them all. In *Proceedings of the 36th annual meeting of the cognitive science society*, 116–121.

Girard, Jean-Yves. 1987. Linear logic. *Theoretical Computer Science* (50). 1–102. DOI: 10.1016/0304-3975(87)90045-4.

Gotham, Matthew. 2018. Making Logical Form type-logical: Glue semantics for Minimalist syntax. *Linguistics and Philosophy* 41(5). 511–556. DOI: 10.1007/s10988-018-9229-z.

Gotham, Matthew & Dag Haug. 2018. Glue semantics for Universal Dependencies. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '18 conference*, 208–226. Stanford: CSLI Publications.

Grønn, Atle & Arnim von Stechow. 2016. Tense. In Maria Aloni & Paul Dekker (eds.), *The Cambridge handbook of formal semantics*, 313–341. Cambridge, UK: Cambridge University Press. DOI: 10.1017/cbo9781139236157.012.

Halvorsen, Per-Kristian. 1983. Semantics for Lexical-Functional Grammar. *Linguistic Inquiry* 14. 567–615.

Halvorsen, Per-Kristian & Ronald M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, 1116–1122. Tokyo. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 279–292).

Haug, Dag. 2008. Tense and aspect for Glue Semantics: the case of participial XADJ's. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '08 conference*, 291–311. Stanford: CSLI Publications.

Haug, Dag. 2009. 'med' – The syntax and semantics of concomitance in Norwegian. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '09 conference*, 338–356. Stanford: CSLI Publications.

Haug, Dag. 2013. Partial control and anaphoric control in LFG. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '13 conference*, 274–294. Stanford: CSLI Publications.

Haug, Dag. 2014. Partial dynamic semantics for anaphora: Compositionality without syntactic coindexation. *Journal of Semantics* 31(4). 457–511. DOI: 10.1093/jos/fft008.

Haug, Dag & Mary Dalrymple. 2020. Reciprocity: Anaphora, scope, and quantification. *Semantics and Pragmatics* 13(10). DOI: 10.3765/sp.13.10.

Heim, Irene & Angelika Kratzer. 1998. *Semantics in generative grammar*. Malden, MA: Blackwell.

Hendriks, Herman. 1993. *Studied flexibility: categories and types in syntax and semantics*. University of Amsterdam. (Doctoral dissertation).

Howard, William A. 1980. The formulae-as-types notion of construction. In Jonathan P. Seldin & J. Roger Hindley (eds.), *To H.B. Curry: essays on combinatory logic, lambda calculus and formalism*, 479–490. Circulated in unpublished form from 1969. Reprinted in de Groote (1995: 15–26). London: Academic Press.

Jacobson, Pauline. 1999. Towards a variable-free semantics. *Linguistics and Philosophy* 22(2). 117–184.

Jacobson, Pauline. 2014. *Compositional semantics: an introduction to the syntax/semantics interface*. Oxford: Oxford University Press.

Joshi, Aravind K., Leon S. Levy & Masako Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences* 10(1). 136–163. DOI: 10.1016/S0022-0000(75)80019-5.

Kaplan, Ronald M. 1987. Three seductions of computational psycholinguistics. In Peter Whitelock, Mary McGee Wood, Harold L. Somers, Rod Johnson & Paul Bennett (eds.), *Linguistic theory and computer applications*, 149–188. London: Academic Press. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 339–367).

Kaplan, Ronald M. 1989. The formal architecture of Lexical-Functional Grammar. *Journal of Information Science and Engineering* 5. 305–322. Revised version published in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 7–27).

Kaplan, Ronald M. 1995. The formal architecture of Lexical-Functional Grammar. In Mary Dalrymple, Ronald M. Kaplan, John T. Maxwell III & Annie Zaenen (eds.), *Formal issues in Lexical-Functional Grammar*, 7–27. Stanford: CSLI Publications. Earlier version published as Kaplan (1989).

Kaplan, Ronald M. & Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan (ed.), *The mental representation of grammatical relations*, 173–281. Cambridge, MA: The MIT Press. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 29–130).

Kasper, Robert. 1997. Semantics of recursive modification. Unpublished manuscript, Ohio State University.

Keenan, Edward L. & Leonard M. Faltz. 1985. *Boolean semantics for natural language*. Dordrecht: Reidel.

Kehler, Andrew, Mary Dalrymple, John Lamping & Vijay Saraswat. 1995. The semantics of resource-sharing in Lexical-Functional Grammar. In *Proceedings of the 1995 Meeting of the European Chapter of the Association for Computational Linguistics*, 31–38. Revised version published in Dalrymple (1999: 191–208). Dublin. DOI: 10.3115/976973.976979.

Kehler, Andrew, Mary Dalrymple, John Lamping & Vijay Saraswat. 1999. Resource sharing in Glue Language Semantics. In Mary Dalrymple (ed.), *Semantics and syntax in Lexical Functional Grammar: The resource logic approach* (Language, Speech, and Communication), 191–208. Earlier version published as Kehler, Dalrymple, Lamping & Saraswat (1995). Cambridge, MA: The MIT Press.

Kennedy, Christopher. 1997. Antecedent contained deletion and the syntax of quantification. *Linguistic Inquiry* 28. 662–688.

Klein, Ewan & Ivan A. Sag. 1985. Type-driven translation. *Linguistics and Philosophy* 8(2). 163–201. DOI: 10.1007/bf00632365.

Kokkonidis, Miltiadis. 2008. First-order glue. *Journal of Logic, Language and Information* 17(1). 43–68.

Kratzer, Angelika. 1995. Stage-level and individual-level predicates. In Gregory N. Carlson & Francis Jeffry Pelletier (eds.), *The generic book*, 125–175. Chicago: University of Chicago Press.

Kuhn, Jonas. 2001. Resource sensitivity in the syntax-semantics interface: Evidence from the German split NP construction. In W. Detmar Meurers & Tibor Kiss (eds.), *Constraint-based approaches to Germanic syntax* (Studies in Constraint-Based Lexicalism), 177–216. Stanford: CSLI Publications.

Lambek, Joachim. 1958. The mathematics of sentence structure. *The American Mathematical Monthly* 65(3). 154–170. DOI: 10.2307/2310058.

Lev, Iddo. 2007. *Packed computation of exact meaning representations*. Stanford: Stanford University. (Doctoral dissertation).

Link, Godehard. 1983. The logical analysis of plurals and mass terms: a lattice-theoretical approach. In Rainer Bäuerle, Christoph Schwarze & Arnim von Stechow (eds.), *Meaning, use, and interpretation of language*, 302–323. Reprinted in Link (1997: 11–34). Berlin: Walter de Gruyter. DOI: 10.1515/9783110852820.302.

Link, Godehard. 1997. *Algebraic semantics in language and philosophy*. Stanford: CSLI Publications.

Lovestrand, Joseph. 2020. F-structure and s-structure of Urdu complex predicates. In Miriam Butt & Ida Toivonen (eds.), *Proceedings of the LFG '20 conference*, 232–249. Stanford: CSLI Publications.

Lowe, John J. 2014. Gluing meanings and semantic structures. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '14 conference*, 387–407. Stanford: CSLI Publications.

Lowe, John J. 2015. *Participles in Rigvedic Sanskrit: the syntax and semantics of adjectival verb forms*. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780198701361.001.0001.

Lowe, John J. 2016. Complex predicates: An LFG+glue analysis. *Journal of Language Modelling* 3. 413–462. DOI: 10.15398/jlm.v3i2.125.

Lowe, John J. 2019. Argument alternations in complex predicates: an LFG+glue perspective. *Linguistic Issues in Language Technology* 17(2). 1–16. DOI: 10.33011/lilt.v17i.1421.

Lowe, John J. & Ali H. Birahimani. 2019. The argument structure of Siraiki causatives. In Miriam Butt, Tracy Holloway King & Ida Toivonen (eds.), *Proceedings of the LFG '19 conference*, 191–211. Stanford: CSLI Publications.

May, Robert. 1977. *The grammar of quantification*. MIT. (Doctoral dissertation).

May, Robert. 1985. *Logical form: its structure and derivation*. Cambridge, MA: The MIT Press.

Meßmer, Moritz & Mark-Matthias Zymla. 2018. The glue semantics workbench: A modular toolkit for exploring linear logic and glue semantics. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '18 conference*, 268–282. Stanford: CSLI Publications.

Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In Jaakko Hintikka, Julian Moravcsik & Patrick Suppes (eds.), *Approaches to language*, 221–242. Reprinted in Montague (1974: 247–270). Dordrecht: Reidel. DOI: 10.1007/978-94-010-2506-5_10.

Montague, Richard. 1974. *Formal philosophy*. New Haven: Yale University Press.

Moortgat, Michael. 1997. Categorial type logics. In Johan van Benthem & Alice ter Meulen (eds.), *Handbook of logic and language*, 1st edn., 93–177. Amsterdam: North-Holland.

Morrill, Glyn. 1994. *Type logical grammar: Categorial logic of signs*. Dordrecht: Kluwer Academic Publishers. DOI: 10.1007/978-94-011-1042-6.

Morrill, Glyn V. 2011. *Categorial grammar: logical syntax, semantics, and processing*. Oxford: Oxford University Press.

Morrison, Andrew. 2017. Adjectival focus as a granularity problem for LFG–Glue. Unpublished manuscript, University of Oxford.

Muskens, Reinhard. 1996. Combining Montague Semantics and Discourse Representation Theory. *Linguistics and Philosophy* 19(2). 143–186.

Mycock, Louise. 2006. *The typology of wh-questions*. Manchester: University of Manchester. (Doctoral dissertation).

Parsons, Terence. 1990. *Events in the semantics of English: A study in subatomic semantics* (Current Studies in Linguistics 19). Cambridge, MA: The MIT Press.

Partee, Barbara H. 1995. Lexical semantics and compositionality. In Lila R. Gleitman & Mark Liberman (eds.), *An invitation to cognitive science: language*, 2nd edn., vol. 1, 311–360. Cambridge, MA: The MIT Press. DOI: 10.7551/mitpress/3964.001.0001.

Partee, Barbara H. & Mats Rooth. 1983. Generalized conjunction and type ambiguity. In Rainer Bäuerle, Christoph Schwarze & Arnim von Stechow (eds.), *Meaning, use, and interpretation of language*, 361–383. Berlin: Walter de Gruyter. DOI: 10.1515/9783110852820.

Potts, Christopher. 2005. *The logic of conventional implicatures*. Oxford: Oxford University Press.

Prawitz, Dag. 1965. *Natural deduction: A proof-theoretical study*. Stockholm: Almquist & Wiksell.

Przepiórkowski, Adam. 2014a. Distance distributivity in Polish: Towards a Glue Semantics approach. In Christopher Piñón (ed.), *Empirical issues in syntax and semantics 10*, 107–124.

Przepiórkowski, Adam. 2014b. Locality constraints in distance distributivity: a propositional glue approach. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '14 conference*, 402–502. Stanford: CSLI Publications.

Przepiórkowski, Adam. 2015. A weakly compositional analysis of distance distributivity in Polish. In Małgorzata Szajbel-Keck, Roslyn Burns & Darya Kavitskaya (eds.), *Formal approaches to Slavic linguistics: The first Berkeley meeting*, 262–281. Ann Arbor: Michigan Slavic Publications.

Przepiórkowski, Adam. 2017. Hierarchical lexicon and the argument/adjunct distinction. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '17 conference*, 348–367. Stanford: CSLI Publications.

Restall, Greg. 2000. *An introduction to substructural logics*. London: Routledge. DOI: 10.4324/9780203016244.

Schabes, Yves, Anne Abeillé & Aravind K. Joshi. 1988. Parsing strategies with 'lexicalized' grammars: application to Tree Adjoining Grammars. In *COLING '88: Proceedings of the 12th Conference on Computational Linguistics*, 578–583. Stroudsburg, PA: Association for Computational Linguistics. DOI: 10.3115/991719.991757.

Steedman, Mark. 1985. Dependency and coördination in the grammar of Dutch and English. *Language* 61(3). 523–568. DOI: 10.2307/414385.

Steedman, Mark. 1987. Combinatory grammars and parasitic gaps. *Natural Language & Linguistic Theory* 5(3). 403–440. DOI: 10.1007/bf00134555.

Steedman, Mark. 2007. On "the computation". In Gillian Ramchand & Charles Reiss (eds.), *The Oxford handbook of linguistic interfaces*, 575–611. Oxford: Oxford University Press. DOI: 10.1093/oxfordhb/9780199247455.001.0001.

Steedman, Mark. 2014. Categorial grammar. In Andrew Carnie, Yosuke Sato & Daniel Siddiqi (eds.), *The Routledge handbook of syntax*, 670–701. New York: Routledge.

Steedman, Mark & Jason Baldridge. 2011. Combinatory categorial grammar. In Robert D. Borsley & Kersti Börjars (eds.), *Non-transformational syntax: Formal and explicit models of grammar*, 181–224. Oxford: Wiley-Blackwell.

van Benthem, Johan. 1991. *Language in action: categories, lambdas, and dynamic logic*. Reprinted in 1995 by the MIT Press, Cambridge, MA. Amsterdam: North-Holland.

van Genabith, Josef & Richard Crouch. 1999. Dynamic and underspecified semantics for LFG. In Mary Dalrymple (ed.), *Semantics and syntax in Lexical Functional Grammar: The resource logic approach* (Language, Speech, and Communication), 209–260. Cambridge, MA: The MIT Press.

Wood, Mary McGee. 1993. *Categorial grammars*. London: Routledge.

Zymla, Mark-Matthias. 2021a. Abstract syntax annotator. Project, University of Konstanz. github.com/Mmaz1988/abstract-syntax-annotator-web.

Zymla, Mark-Matthias. 2021b. Glue semantics workbench, v.2. Project, University of Konstanz. github.com/Mmaz1988/GlueSemWorkbench_v2.

Zymla, Mark-Matthias. 2021c. XLE interface for glue semantics workbench. Project, University of Konstanz. github.com/Mmaz1988/xle-glueworkbench-interface.