Missing Resources in a Resource-Sensitive Semantics[*]

Gianluca Giorgolo
Gianluca_Giorgolo@carleton.ca
Carleton University

Ash Asudeh
ash.asudeh@ling-phil.ox.ac.uk
Carleton University &
University of Oxford

June 29, 2012
LFG 2012 · Bali

1 Introduction

(1) Alice ate yesterday afternoon. ⇔ Alice ate something yesterday afternoon.[1]

(2) Bob drank last night. ⇔ Bob drank something last night / Bob drank something alcoholic last night.

(3) (?) Bob loves drinking. ⇔ Bob loves drinking alcohol.

(4) Yesterday, Alice debugged for three hours. ⇔ Yesterday, Alice debugged some code/some programs for three hours. *(in a context in which it is known that Alice finds debugging annoying)*

(5) Silvio was accused of tax fraud. ⇔ Silvio was accused of tax fraud by someone. / Someone accused Silvio of tax fraud.

(6) Silvio was accused. ⇔ Silvio was accused of something by someone.

Generalization: Semantic roles that qualify as (derived) arguments according to Needham and Toivonen (2011) are in these cases considered optional. However the equivalences show that at the level of semantic representation their slot is filled by an existentially bound variable.

But notice that there are cases in which a missing argument is interpreted universally:

(7) W.H.O. warns against homeopathy use. ⇔ W.H.O. warns everyone against homeopathy use.

[*]This research is supported by an Early Researcher Award from the Ontario Ministry of Research and Innovation and NSERC Discovery Grant #371969.

[1]But "Alice devoured yesterday" is ungrammatical.

and in other cases deictically/indexically:

(8) Bob arrived yesterday. $\Leftrightarrow$ Bob arrived from somewhere yesterday to the contextually relevant location.[2]

Question: how can we capture the semantic contribution of missing resources and therefore account for these equivalences (or just entailments) in a resource sensitive semantics? The fact that we assume that our semantic process is sensitive to use of resources makes this question even more central as the phenomena we investigate seem to contradict the assumption of resource sensitivity.

1.1 Prior work

- Bresnan (1978) proposes a *lexical solution* for the cases of transitive verbs used intransitively:

  eat: V, [___ NP], NP_1 $\mathbf{eat}$ NP_2
  [___], $(\exists$ y) NP_1 $\mathbf{eat}$ y

  The idea is to transform the binary predicate $\mathbf{eat}$ into a unary predicate by *binding* with an *existential quantifier* the argument corresponding to the object.

  Dowty (1982) proposes basically the same analysis in the context of Montague Grammar by introducing so-called *Relation-Reducing Rules*. For instance, one rule transforms a transitive verb like “eat” into an intransitive one and at the same time changes the semantics of the verb by binding its second argument to an existential quantifier.

  Rules like those proposed by Dowty (1982) have no clear translation in the context of LFG so an approach similar to the one of Bresnan (1978) may be preferable. The solution we propose in Asudeh and Giorgolo (2012) is basically an extension of Bresnan (1978).

- In response to Bresnan (1978), Fodor and Fodor (1980) notice that in case of a quantifier in subject position, the existential binding the second argument of $\mathbf{eat}$ must take narrow scope:

  (10) Every boy ate. $\Rightarrow$ Every boy ate something.

  The entailment holds only between the first sentence and the second when interpreted as follows:

  (11) $\forall x \exists y(\mathbf{boy}(x) \rightarrow \mathbf{eat}(x, y))$

  (Notice that this generalization is already captured in the adaptation of Bresnan (1978) in Asudeh and Giorgolo (2012))

  To capture this fact they propose a different approach:

  – At the semantic representation level we have two different $\mathbf{eat}$ predicates, a unary $\mathbf{eat}_1$ and a binary $\mathbf{eat}_2$ version.

[2]However notice that Stanley (2000) proposes an analysis of this unexpressed arguments in which they are considered bound by a linguistic operator, based on examples like the following

(9) Everywhere Bob goes, it rains.

where the raining event location co-varies with the locations quantified over by “everywhere”.

– These predicates are related through *meaning postulates*:

(12) $\mathbf{eat}(c) \leftrightarrow \exists y(\mathbf{eat}(c, y))$ with c a constant term

(13) $Qx(\mathbf{eat}(x)) \leftrightarrow Qx\exists y(\mathbf{eat}(x, y))$ with Q a quantifier

– The relative order of the quantifier in (13) is explained on the basis of (12) and of general logical principles:

$$\begin{tikzcd}
\forall x(P(x)) \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \forall x\exists y(P(x, y)) \arrow[d, leftrightarrow] \\
P(x_1) \wedge \ldots \wedge P(x_n) \arrow[r, leftrightarrow] & \exists y(P(x_1, y)) \wedge \ldots \wedge \exists y(P(x_n, y))
\end{tikzcd}$$

– The solution is therefore composed by a mix of lexical ambiguity and axioms specific to the meaning representation format. The narrow scope of the existential quantifier can actually be captured also in the “closure” approach by assigning to the intransitive use *eat* the meaning:

(14) $\lambda x\exists y(\mathbf{eat}(x, y))$

Drawbacks: If we want to extend this approach to other constructions (e.g. passives) we face an explosion of postulates that need to be added to the lexicon.

Lexical postulates like (13) have problems also in capturing the totality of the possible cases. For example to account for implicit existential quantification in sentences like (15) we would need to list all the possible scopal readings for the quantifiers binding the various arguments of the predicate $\mathbf{accuse}$.

(15) Most politicians were accused of at least two crimes

- Carlson (1984) and Lasersohn (1993) analyse these phenomena according to a neo-davidsonian perspective. In particular Lasersohn (1993) approach is motivated by a problem that approaches based on an existential closure have with distributive readings:

(16) The papers were graded.

(17) $\forall y\exists x(y \in \mathbf{paper}^* \rightarrow \mathbf{grade}(x, y))$

However if we assume that (20)[3] represents the distributive reading of *grade* then the existential closure procedure gives us the incorrect wide scope reading for the unspecified agent

[3]Lasersohn (1993) actually uses a meaning postulate

(18) $\alpha(X) \leftrightarrow \forall y(y \in X \rightarrow \alpha(y))$

which makes its argument less compelling as it would be easy to modify this postulate in the case of a binary (or in general n-ary predicate) predicate to obtain the desired result:

(19) $Qx(R(x, Y)) \leftrightarrow \forall yQx(y \in Y \rightarrow R(x, y))$ where Q is a quantifier and Y a plural entity.

This approach would however suffer the same drawbacks described for Fodor and Fodor (1980).

reported in (21).

(20) $\lambda y \lambda x \forall z(z \in y \rightarrow \mathbf{grade}(x, z))$

(21) $(\lambda y \lambda x \forall z(z \in y \rightarrow \mathbf{grade}(x, z)))\ \mathbf{paper}^* \rightsquigarrow \exists x \forall z(z \in \mathbf{paper}^* \rightarrow \mathbf{grade}(x, z))$

The solution proposed by Lasersohn (1993) is based on *event semantics*. A sentence like (16) is interpreted as (22), where no mention to an agent is made.

(22) $\exists e(\mathbf{grade}(e) \wedge \textsc{patient}(\mathbf{paper}^*, e))$

According to this solution, the agent is introduced via an ontological postulate that states that every *atomic* “grading” event must have an agent (see (24)).[4] The restriction to atomic events is crucial.

(24) $\forall e(\textsc{atom}(\mathbf{grade}, e) \rightarrow \exists x(\textsc{agent}(x, e)))$

Drawbacks:

– Unaccusative uses of verbs supporting distributive readings:

(25) The mirrors were broken $\Rightarrow$ Each mirror was broken by someone / something

(26) My TV broke $\not\Rightarrow$ Someone / something broke my TV

(27) 19.6% of consumers contacted believed business conditions will improve in the coming six months $\not\Rightarrow$ 19.6% of consumers contacted believed someone/something will improve business conditions in the coming six months

In the case of (26) the ontological postulate that states that atomic “breaking” events must have an agent gives the wrong interpretation.[5]

– In some cases a reading with a wide scope for the existentially bound implicit argument is preferred

(29) The numbers were summed

The preferred reading for sentence (29) is one where a single entity sums the numbers (or where at least this entity performs the final addition that gives the final result).

– This approach still requires lexically specified rules to capture the fact that “John ate” is grammatical while “John devoured” is not.

• Blom et al. (2011) recently proposed, in the context of Abstract Categorial Grammar, a formal semantics for the phenomena illustrated in (1) - (6). The approach is fully *lexical* and it is based on the ideas of *option or sum types* and *choice*. The uniqueness of this approach is represented by the fact that they do not need to analyze a verb like “eat” as semantically ambiguous.

[4] ATOM is defined by Lasersohn (1993) as follows:

(23) $\textsc{atom}(\alpha, e) \leftrightarrow (\alpha(e) \wedge \neg\exists e'(e' < e \wedge \alpha(e')))$

[5] Some languages use a reflexive-like construction that could suggest an implicit agent co-referring with the patient. However this is not the case:

(28) Ieri, alle tre, la porta si è chiusa. È stato Marco. (*Yesterday, at three, the door closed. Marco did it.*)

– Their first assumption is an extension to the type system: for each type τ they create a new type τ^o by adding to the original type a distinguished element $*$

– They also assume a lambda calculus for meaning creation and composition that includes a simplified *if - then - else* construct in the form of an operator `option`:

(30) $$\texttt{option}(x, f, d) = \begin{cases} d & \text{if } x = * \\ f(x) & \text{otherwise} \end{cases}$$

With these two extensions in place they proceed by modifying the familiar standard lexical entry for a verb like "eat" in the following way:

(31) $$\lambda o \lambda s(\texttt{option}(o, \lambda u(\mathbf{eat}(s, u)), \exists x(\mathbf{eat}(s, x)))) : e^o \rightarrow e \rightarrow t$$

In this way they *do not need* to analyze "eat" as *ambiguous* between two meanings. Instead they change its type, by letting it take as its object argument an element of the special type e^o. Blom (2012) proposes a solution based on the introduction of an optionalization and a de-optionalization operation. We show that they can be implemented in terms of monads, an extension we have already independently introduced for Glue Semantics in Giorgolo and Asudeh (2011).

2 Main claims

1. Verbs like *eat* are not ambiguous between two meanings; nor should the passive form be analyzed as being ambiguous (we assume here that agentive *by*-phrases are not adjuncts).

2. Certain lexical items and certain grammatical constructions have instead a "*safety mechanism*" that allows them to function also when a *specific* resource is missing.

3. To draw a parallel with computer science, certain expressions have the capability of dealing with exceptional situations via a mechanism similar to a *throw-catch* construct (more on this similarity below).

4. We propose to adapt the proposal by Blom et al. (2011) to the LFG Correspondence Architecture by assuming that the exceptional situation is signalled during the construction of the Glue term and is handled in the compositional process through lexically specified mechanisms.

5. We show that the our adaptation of Blom et al. (2011) is an instance of a *monad* which means that we can re-use the extension of Glue Semantics we presented in Giorgolo and Asudeh (2011).

6. The advantages:

   - No need to introduce lexical postulates. "Exceptional" behavior is already encoded in the lexicon. This means that we can also capture generalizations like the optionality of agents in passive constructions via a lexical entry for certain morpho-syntactic features.
   - Avoiding an event-based approach we stay clear from ontological issues that are *not* directly *related* to linguistic phenomena.

- The extension to Glue Logic we use has already been proposed for an unrelated phenomenon (Giorgolo and Asudeh 2011). This supports the hypothesis that this augmented form of composition is part of the compositional toolkit of natural language.
- Compared with the approach we present in Asudeh and Giorgolo (2012) we do not need to resort to the θ mapping which introduces some form of (possibly) undesirable non-determinism in the grammatical architecture (we can go from the a-structure to the s-structure following two different paths that are not equal).

3 Overview

1. *Introduction*
2. *Main claims*
3. *Overview*
4. Missing resources
5. Analysis
6. Comparing Giorgolo and Asudeh with Asudeh and Giorgolo
7. Conclusions

4 Missing resources

The resource sensisitive perspective of Glue Semantics prompts us to be more specific about the details of how the existential closure is used to fill the missing argument slot.

We assume that the denotation of a passive verb or of a verb that supports implicit objects is a function from two entities to a truth value. These expressions are however special as they are capable of handling the absence of an expression denoting one of the two arguments of this function.

According to Blom et al. (2011)'s interpretation these arguments are considered optional. Yet they need to be represente in the derivation. Therefore the approach they take is to allow them to be inserted freely in the derivation. However this procedure goes against the resource sensitive assumptions of Glue Semantics. We propose therefore a different interpretation of optional arguments that help us understand how such an object may enter the derivation.

Instead of considering the first argument of the denotation of a verb like "eat" an optional argument we will take it to be an obligatory argument (from the perspective of functional application it is obligatory). However instead of being a value of type e we will take it to be the result of a computation that yields values of type e. What is crucial is that the computation may *fail*. A verb like "eat" is capable of recovering from such a failure. The kind of machinery we have in mind is similar to a *throw-catch* construct of many programming languages.

When during a computation an error occurs (the error is said to be *thrown*) we can have mechanisms in place, higher up in the chain of control that directs the computation, that provide an alternative way to continue the computation (they *catch* the error). If such a mechanism is not present the error is simply propagated back to the user and the system does not return any usable value. Similarly when we attempt to access a lexical resource that is not there (e.g. the object of a verb) we can either have a mechanism that fixes the problem (we bind the empty slot with an existential quantifier, as we can do with verbs like "eat" or "sing") or we end in an illegal configuration (the ungrammaticality of "*John devoured").

The computation that may throw the error (i.e. not produce the required value) corresponds to the procedure that construct the Glue terms that guide the compositional process from the s-structure projected by the f-structure of an expression.

We show how a *throw-catch*-like mechanism can be implemented in Glue Semantics on the basis of Blom et al. (2011)'s `option` function and the *monadic extension* we proposed in Giorgolo and Asudeh (2011).

4.1 Monads to the rescue

A short monad primer:

- Monads where first used to give a unified analysis of various semantic phenomena by Shan (2001)
- The main intuition behind monads is that they are a way to reproduce the structure of a space of values and functions in a richer setting that carries more information, in the sense that we can specify more things about the values and functions.
- We can move from the information-poor space to the information-rich space as follows:
  - A value or function in the poor space is mapped to an information-enriched counterpart by associating the value or function with some sort of default information. In this way, we get an object of the right information-rich type, without committing to any particular enriched information.
  - For example, in the case of multidimensionality, the values and functions that contribute only to at-issue material can be mapped to a richer space where they have a vacuous side-issue component.
- A more operational way to look at monads is to consider them as computations that yield values.
- A monad is defined by a triple $\langle M, \eta, \star \rangle$.
- η ('unit') is the mapping from the information-poor space to the information-rich space.
- $\star$ ('bind') is the mechanism for extracting values from computations and creating new computations using these values. $\star$ also allows ordering for side-effects of computation.
- M is the label for the information-rich counterpart of the original, information-poor types.

In this case the monad we want to use is the $Option$ or $Maybe$ monad:

(32) $\eta(x) = x$

(33) $m \star k = \begin{cases} * & m = * \\ k(m) & \text{otherwise} \end{cases}$

Instead of the extension based on a second implication that we proposed in Giorgolo and Asudeh (2011), we will present here a superior and more elegant system that Avery Andrews (p.c.) suggested and that is based on the logical system proposed by Benton et al. (1998).

In this system monadic values are marked with a sort of modality operator $\Diamond$ (which will play the role of M). In the case under consideration we can intuitively associate a resource with type $\Diamond a$ to something that possibly yields a value of type a.

The two rules for the introduction and elimination of the $\Diamond$ unary connective correspond directly to the η operation (if we have a value we can produce it with a computation that does nothing but produce the value) and the $\star$ operation (the intuitive interpretation of the rule is that if we can construct a computation yielding a value of type b by assuming a value of type a and we have a computation that produces a value of type a then we can plug the result of this computation in the body of the b-yielding computation).

$$\frac{x : a}{\eta(x) : \Diamond a}\,\Diamond I \qquad \frac{m : \Diamond a \qquad \begin{array}{c}[x : a]_i \\ \vdots \\ f : \Diamond b\end{array}}{m \star \lambda x.f : \Diamond b}\,\Diamond E_i$$

In this framework verbs like “eat”, “read” or “sing” subcategorize for an object but they consume it only when wrapped in a monad:[6]

(34) eat V $(\uparrow \textsc{pred}) = \text{‘}\mathbf{eat}\text{'}$
$\lambda o \lambda s(\texttt{option}(o, \lambda u(\mathbf{eat}(s,u)), \exists x(\mathbf{eat}(s,x))))$
$\Diamond(\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$

The λ-term in (34) represents the semantic contribution of “eat”. It is a function of two arguments, o and s, the former a *computation* returning a value of type e while the latter is a pure value of type e (produced possibly by a computation at a different level). The body of function uses the `option` procedure to test the result of o: if it is a value of type e then the term $\lambda u(\mathbf{eat}(s,u))$ is applied to the result and the result is used as the second argument of the relation $\mathbf{eat}$, otherwise `option` returns its third argument $\exists x(\mathbf{eat}(s,x))$ were the second argument of $\mathbf{eat}$ is bound by the existential quantifier.

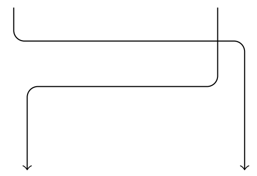
[6]For the sake of simplicity we do not require anything of the implicit object but to exist. A more realistic lexical entry would require the bound variable to be something that is food for the referent. For example if Dr. McCoy from Star Trek utters “Every subject ate” referring to a group of alien beings in his lab, we expect that each subject ate something compatible with its biology. Notice that the lexical entry in (34) allows us to make the value of the bound variable x dependent on the value of the variable s as required.

In the case of a passive construction we can derive its denotation from the active form using the function `passivize` defined in (35) that takes as argument a function of type $e \to e \to t$ and returns a new function of type $\mathit{Option}\, e \to e \to t$.[7]

(35) $\texttt{passivize}(f) = \lambda a \lambda p(\texttt{option}(a, \lambda a(f(a,p)), \exists x(f(x,p))))$

At the level of Glue terms this corresponds to remapping the template in (37) into the one in (38).

(37) $(\uparrow \text{OBJ})_\sigma \multimap (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$



(38) $\Diamond(\uparrow \text{OBL})_\sigma \multimap (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$

5 Analysis

We present here some examples of how we propose to analyze the phenomena we are interested in. We assume the toy lexicon in table 1.

5.1 Implicit objects

The first example we analyze is the case of an implicit object (39).

(39) John ate

The simplified f-structure associated with the (39) is show in (40).

(40) $$e\begin{bmatrix} \text{PRED} & \textbf{`eat'} \\ \text{SUBJ} & j\begin{bmatrix} \text{PRED} & \textbf{`john'} \end{bmatrix} \end{bmatrix}$$

The f-structure in (40) projects an s-structure that is used to construct the premises for the glue proof.[8] In Glue Semantics the s-structure is then used together with the lexicon as the input for

[7]Alternatively we could move the `option` outside the second lambda abstraction:

(36) $\texttt{passivize}(f) = \lambda a(\texttt{option}(a, \lambda a \lambda p(f(a,p)), \lambda p \exists x(f(x,p))))$

The two definitions are equivalent.

[8]We assume here that subcategorization of grammatical functions such as SUBJ and OBJ is not represented in the f-structure but is rather encoded in the linear terms that control meaning composition (Kuhn 2001, Asudeh 2012).

Word	Category	Constraints
John	N	$(\uparrow\ \textsc{pred}) = \text{‘}\mathbf{john}\text{’}$ $\mathbf{john} : \uparrow_\sigma$
ate	V	$(\uparrow\ \textsc{pred}) = \text{‘}\mathbf{eat}\text{’}$ $\lambda o \lambda s(\mathtt{option}(o, \lambda u(\mathbf{eat}(s,u)), \exists x(\mathbf{eat}(s,x))))$ $\Diamond(\uparrow\ \textsc{obj})_\sigma \multimap (\uparrow\ \textsc{subj})_\sigma \multimap \uparrow_\sigma$
something	N	$(\uparrow\ \textsc{pred}) = \text{‘}\exists\text{’}$ $\lambda P \exists x(P(x))$ $(\uparrow_\sigma \multimap X) \multimap X$
kiss	V	$(\uparrow\ \textsc{pred}) = \text{‘}\mathbf{kiss}\text{’}$ $\lambda o \lambda s(\mathbf{kiss}(s,o))$ $(\uparrow\ \textsc{obj})_\sigma \multimap (\uparrow\ \textsc{subj})_\sigma \multimap \uparrow_\sigma$
kissed$_{\text{pass}}$	V	$(\uparrow\ \textsc{pred}) = \text{‘}\mathbf{kiss}\text{’}$ $\mathtt{passivize}(\lambda o \lambda s(\mathbf{kiss}(s,o))) \rightsquigarrow$ $\lambda a \lambda p(\mathtt{option}(a, \lambda a(\mathbf{kiss}(a,p)), \exists x(\mathbf{kiss}(x,p))))$ $\Diamond(\uparrow\ \textsc{obl})_\sigma \multimap (\uparrow\ \textsc{subj})_\sigma \multimap \uparrow_\sigma$

Table 1: Toy lexicon

the procedure that generates *resources* (i.e. premises) for the semantic derivation. This procedure is normally understood as producing a set of resources/premises. What we make explicit is the possibility that this procedure encounters an exceptional situation, such as when attempting to instantiate the linear formula template for the verb “ate”. In that case, there is no linking with the s-structure projected by the OBJ feature as no such feature is present. We assume that the procedure signals this error and link it to the rest of the template formula which can instead be instantied. The error becomes therefore a (faulty) premise for the semantics derivation.

Alternatively we can reuse some of the intuitions of the second analysis we presented in Asudeh and Giorgolo (2012). If we posit that the lexical entry of a verb like “eat” introduces in the s-structure both an AGENT and a PATIENT feature whose values are determined on the basis of the f-structure by the σ projection we can understand the presence in our derivations of an error premise in two (roughly equivalent) ways:

1. the values of the features of the s-structure may all be initialized to $*$ signaling by default that no resource, corresponding to that semantic feature, has been explicitly introduced yet. The σ projection fills the values of the features that have a corresponding f-structural counterpart. In the case of an implicit object the PATIENT feature receives no value.
2. σ by default attempts to fill the values of all s-structural features. If a feature cannot be assigned a value an error is raised and registered in the s-structure using the special value $*$.

If we choose this approach we have to change the lexicon accordingly to make direct reference to the s-structural features. The changes are straightforward. In the rest of the presentation we will use the first implementation of the monadic approach.

In the derivation the error premise is given the name $\Diamond n$ as it is not a pure value but a computational object, and is represented by $*$. The resulting proof is shown below and consists of two simple functional applications/$\multimap$-eliminations.

$$\dfrac{\overset{\text{John}}{\mathbf{john} : j} \qquad \dfrac{\overset{\text{ate}}{\lambda o \lambda s(\mathtt{option}(o, \lambda u(\mathbf{eat}(s,u)), \exists x(\mathbf{eat}(s,x)))) : \Diamond n \multimap j \multimap e} \qquad \overset{\mathtt{error}}{* : \Diamond n}}{\lambda s \exists x(\mathbf{eat}(s,x)) : j \multimap e}\multimap E}{\exists x(\mathbf{eat}(\mathbf{john}, x)) : e}\multimap E$$

The resulting interpretation corresponds to the intuitive meaning associated with (39).

5.2 Explicit objects

The second example we consider shows how the same lexical entry for the verb "ate" generates the correct interpretation when the object is explicitly realized:

(41) John ate something

Based on the f-structure in (42) we associate with the sentence the semantic derivation in figure 1.

(42)
$$e\begin{bmatrix} \text{PRED} & \text{'}\mathbf{eat}\text{'} \\ \text{SUBJ} & j\begin{bmatrix}\text{PRED} & \text{'}\mathbf{john}\text{'}\end{bmatrix} \\ \text{OBJ} & st\begin{bmatrix}\text{PRED} & \text{'}\exists\text{'}\end{bmatrix} \end{bmatrix}$$

The crucial step in the proof is the "lowering" of the type of the denotation of "ate" from the type $\mathit{Option}\ e \rightarrow e \rightarrow t$ to the type $e \rightarrow e \rightarrow t$. The shape of the proof steps that correspond to this operation is the following:

$$\dfrac{\dfrac{\dfrac{[x : a]_1}{\eta(x) : \Diamond a}\Diamond I \qquad f : \Diamond a \multimap b}{f(\eta(x)) : b}\multimap E}{\lambda x.f(\eta(x)) : a \multimap b}\multimap I$$

At the level of meaning terms we simply create a new function that wraps its first argument in a monad using η, generating therefore a computation that does nothing besides returning the value passed as argument.

$$
\dfrac{
  \dfrac{
    \begin{matrix}\text{John}\\ John : j\end{matrix}
    \qquad
    \dfrac{
      \dfrac{
        \dfrac{
          \begin{matrix}\text{ate}\\ \lambda o \lambda s(\texttt{option}(o, \lambda u(\mathbf{eat}(s,u)), \exists x(\mathbf{eat}(s,x)))) : \Diamond t \multimap j \multimap e\end{matrix}
          \qquad
          \dfrac{[z : t]_1}{\eta(z) : \Diamond t}\;\Diamond I
        }{\lambda s(\mathbf{eat}(s,z)) : j \multimap e}\;\multimap E
      }{\lambda z \lambda s(\mathbf{eat}(s,z)) : t \multimap j \multimap e}\;\multimap I_1
      \qquad
      [w : t]_2
    }{\lambda s(\mathbf{eat}(s,w)) : j \multimap e}\;\multimap E
  }{
    \dfrac{\mathbf{eat}(John, w)) : e}{\lambda w(\mathbf{eat}(John, w)) : t \multimap e}\;\multimap I_2
  }\;\multimap E
  \qquad
  \begin{matrix}\text{something}\\ \lambda P \exists y(P(y)) : (t \multimap X) \multimap X\end{matrix}
}{\exists y(\mathbf{eat}(John, y)) : e}\;\multimap E
$$

Figure 1: Proof for “John ate something”

5.3 Passives

Finally, we show how a passive construction without a *by*-phrase gets an existential interpretation. The example sentence and the associated f-structure are shown respectively in (43) and (44). The proof has exactly the same shape as the one for the case of an implicit object. What is interesting is how the Glue and the meaning terms for the passive form of “kiss” are structured on the basis of their active counterparts (see table 1). The resulting denotation corresponds to a function that is capable of providing an existential closure in case the *agent* is not expressed phonologically.

As it was the case for the analysis of implicit objects, the procedure that instantiate the linear formula governing the compositional behavior of “kissed$_{\text{pass}}$” fails as there is no projection of a OBL feature in the s-structure. The error is added to the premises that guide the semantic composition reasoning and is linked to the resource corresponding to the passive verb.

(43) John was kissed

(44)
$$k\left[\begin{array}{ll}\text{PRED} & \textbf{`kiss'}\\ \text{SUBJ} & j\left[\begin{array}{ll}\text{PRED} & \textbf{`john'}\end{array}\right]\end{array}\right]$$

$$\dfrac{\begin{array}{c}\text{John}\\ \mathbf{john} : j\end{array} \qquad \dfrac{\begin{array}{c}\text{kissed}_{\text{pass}}\\ \lambda a\lambda p(\texttt{option}(a, \lambda a(\mathbf{kiss}(a,p)), \exists x(\mathbf{kiss}(x,p)))) : \Diamond n \multimap j \multimap k\end{array} \qquad \begin{array}{c}\texttt{error}\\ * : \Diamond n\end{array}}{\lambda p \exists x(\mathbf{kiss}(x,p)) : j \multimap k} \multimap E}{\exists x(\mathbf{kiss}(x,\mathbf{john})) : e} \multimap E$$

6 Comparing this approach with the non-monad approach

The two approaches we have presented in these two papers are not distinguishable at the level of data coverage. They predict exactly the same set of well formed expressions. From a grammatical perspective there is no reason to prefer one of the two approaches over the other.

However, the two approaches make *different predictions* in terms of *language processing*:

- Asudeh and Giorgolo (2012) predict in fact that the interpretation of an expression that contains a verb like “eat” will prompt the generation of two parallel semantic derivations

    1. The non-optional semantic resource contributed by the verb is always introduced.
    2. However the additional, optional resource can either enter the semantic derivation or not. We effectively split the process of interpreting an utterance in two, as a listener may decide to either include the optional resource or not.[9]

[9] It woud be possible to avoid the need for two semantic derivations by granting a special status to the optional resource in the semantic derivation. We could allow the optional resource to be always present but freely discarded in case it is not needed. This would however require us to radically modify the core assumption of resource sensitivity of Glue Semantics.

- The analysis proposed here predicts that the interpretation of the same expression requires only one semantic derivation: the shape of the derivation is determined by the process that goes from the phonological string to the meaning of the utterance

Assuming that cognitive operations come with a computational cost (in terms of cognitive resources, as typically measured by processing time) we should therefore be able to distinguish the two approaches with an experiment that tests the interpretation of similar expressions.

According to our other analysis (Asudeh and Giorgolo 2012), we expect an increase of computational costs in cases that involve an implicit argument (assuming that listeners attempt to first derive the meaning of the utterance without resorting to the optional semantic resource).

On the other hand, the present approach predicts that a sentence like “John ate” should have the same processing costs as a regular intransitive sentence (with the possibility of an additional unitary cost due to the more complex denotation associated with the verb).

7 Conclusion

- We have presented a unified analysis of implicit arguments in LFG.
- Our solution is an adaptation of the analysis of Blom et al. (2011)
- The solution is in a sense hyperlexical, as the machinery needed to obtain the existentially bound reading associated with implicit arguments is fully expressed in the lexical entry.
- We have shown how this approach can be interpreted as another example of a monadic computation, an extension to the compositional process of Glue Semantics that we have already introduced on the basis of independent considerations in Giorgolo and Asudeh (2011).
- The main advantages of this approach are its generality, the fact that we avoid the need to claim that verbs like “eat” or passive constructions are semantically (or maybe just compositionally) ambiguous and the fact that we can reuse machinery already in Glue Semantics without having to postulate anything new.

References

Asudeh, Ash. 2012. *The Logic of Pronominal Resumption*. Oxford: Oxford University Press.

Asudeh, Ash, and Gianluca Giorgolo. 2012. Flexible Composition for Optional and Derived Arguments. LFG 2012.

Benton, Nick, G. M. Bierman, and Valeria de Paiva. 1998. Computational Type from a Logical Perspective. *Journal of Functional Programming* 8(2): 177–193.

Blom, Chris. 2012. Optional Arguments in Abstract Categorial Grammar. Master's thesis, Utrecht University.

Blom, Chris, Philippe de Groote, Yoad Winter, and Joost Zwarts. 2011. Implicit Arguments: Event Modification or Option Type Categories? In *Pre-proceedings of the 18th Amsterdam Colloquium*.

Bresnan, Joan. 1978. A Realistic Transformational Grammar. In Morris Halle, Jan Bresnan, and George A. Miller, eds., *Linguistic Theory and Psychological Reality*. MIT Press.

Carlson, Greg N. 1984. Thematic Roles and Their Role in Semantic Interpretation. *Linguistics* 22(3): 259–280.

Dowty, David. 1982. Grammatical Relations and Montague Grammar. In Pauline I. Jacobson and Geoffrey K. Pullum, eds., *The Nature of Syntactic Representation*, 79–130. D. Reidel.

Fodor, Jerry A., and Janet Dean Fodor. 1980. Functional Structure, Quantifiers, and Meaning Postulates. *Linguistic Inquiry* 11(4): 759–770.

Giorgolo, Gianluca, and Ash Asudeh. 2011. Multidimensional Semantics with Unidimensional Glue Logic. In Miriam Butt and Tracy Halloway King, eds., *Proceedings of the LFG11 Conference*, 236–256. CSLI Publications.

Kuhn, Jonas. 2001. Resource Sensitivity in the Syntax-Semantics Interface and the German Split NP Construction. In W. Detmar Meurers and Tibor Kiss, eds., *Constraint-Based Approaches to Germanic Syntax*. Stanford, CA: CSLI Publications.

Lasersohn, Peter Nathan. 1993. Lexical Distributivity and Implicit Arguments. In *Proceedings from Semantics and Linguistic Theory III*, 145–161.

Needham, Stephanie, and Ida Toivonen. 2011. Derived Arguments. In Miriam Butt and Tracy Holloway King, eds., *Proceedings of the LFG11 Conference*.

Shan, Chung-chieh. 2001. Monads for Natural Language Semantics. In Kristina Striegnitz, ed., *Proceedings of the ESSLLI-2001 Student Session*, 285–298. 13th European Summer School in Logic, Language and Information.

Stanley, Jason. 2000. Context and Logical Form. *Linguistics and Philosophy* 23: 391–434.