Expletive form and function in a constraint-based lexical-realizational framework

Ash Asudeh
Department of Linguistics
University of Rochester

Nantes Expletives Workshop
May 23, 2024

lrfg.online

- This paper considers the following questions:
  1. How is the form of expletives determined in a lexical-realizational (Stump 2001) approach to morphology, such as Distributed Morphology (Halle and Marantz 1993)?
  2. What is the function of expletives in grammar?
  3. Can this function be captured in compositional semantics? What if the semantics is *resource-sensitive* (Asudeh 2012, 2022)?
- I will also use this opportunity to introduce a new framework, L_RFG which is a combination of Distributed Morphology (DM) and Lexical-Functional Grammar (LFG; Kaplan and Bresnan 1982, Bresnan et al. 2016, Dalrymple et al. 2019).
  - L_RFG is both a kind of DM and a kind of LFG, which perhaps sounds initially counter-intuitive but seems to deliver nice results.
- Indeed, once L_RFG has been introduced at an adequate level of detail, the answers to the questions above fall out relatively naturally.

1 The framework

1.1 Motivation

- L_RFG is the offspring of an unlikely marriage between Distributed Morphology as a theory of morphological realization and Lexical-Functional Grammar as a constraint-based theory of syntax and grammatical architecture.
- L_RFG combines the strengths of the two frameworks:
  1. Like LFG, it is a declarative, representational and constraint-based theory that is ideally suited to modelling nonconfigurationality.
  2. Like DM, it provides a realizational, morpheme-based view of word-formation and is good at modelling complex morphological structures including those found in polysynthetic languages, such as many North American Indigenous languages.

1.2 Grammatical architecture

- L_RFG is based on a version of LFG's *Correspondence Architecture* (Kaplan 1987, 1995, Asudeh 2006, 2012). The L_RFG correspondence architecture is shown in Figure 1.
  - C(onstituent)-structure is mapped to f(unctional)-structure by the standard ϕ correspondence function from LFG.
  - Similarly, f-structure is mapped to s(emantic)-structure by the standard σ correspondence function familiar from work in LFG+Glue (among others, Dalrymple et al. 1993, 2019, Dalrymple 1999, Asudeh 2012) and s-structure is mapped by the ι correspondence function to i(nformation)-structure (Dalrymple and Nikolaeva 2011).
  - This lower path through the architecture is called the *Structure-Function-Meaning Path* (*SFM path* for short).
- C-structure is also mapped to v(ocabulary)-structure via a new correspondence function, ν (meant to be reminiscent of a lower case *v*).
  - V-structure is the site of morphosyntactic *exponence*; the ν function maps from bundles of information (categories and features) in c-structure terminal nodes to representations of the information's morphophonological exponent.
  - V-structure is in turn mapped by the ρ correspondence function to p(rosodic)-structure (Bögel 2015) and p-structure is mapped by the o correspondence function to the p(honological)-string (o is meant to be reminiscent of *output*). P-structure and the p-string are represented in p-diagrams (Bögel 2015).
  - This upper path through the architecture is called the *Morphology-Prosody-Phonology Path* (*MPP path* for short).

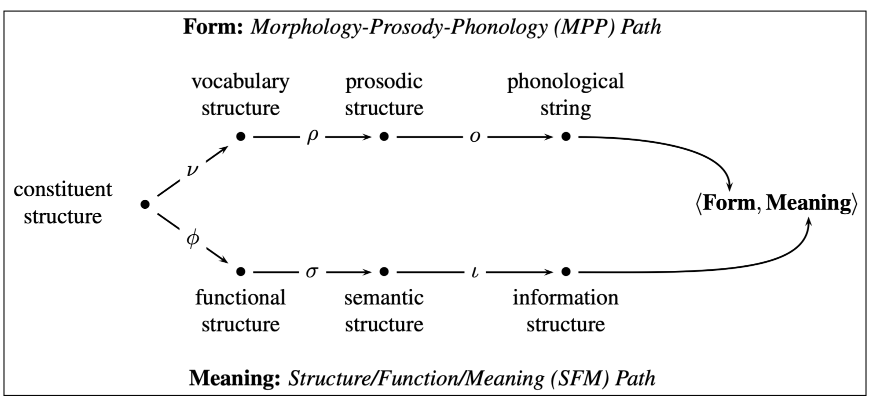


Figure 1: The L_RFG correspondence architecture (Asudeh et al. 2023)

- An L_RFG analysis has three main components (Melchin et al. 2020, Asudeh and Siddiqi 2023, Asudeh et al. 2023):

1. A set of annotated c-structure rules, as in standard LFG
2. A set of *Vocabulary Items* (VIs), each of which is a mapping from an exponendum to an exponent. The exponent is mapped to the realization. Thus, L_RFG separates exponence and realization as follows (Asudeh et al. 2023):

(1) *exponendum* $\xrightarrow{\nu}$ *exponent* $\xrightarrow{o \circ \rho}$ *realization*

The set of vocabulary items is called the *Vocabulary*.

3. A set of constraints for choosing the best available VI.
   (a) A family of **MostInformative** constraints which respectively prefer a VI based on its c-structural information (**MostInformative**$_c$), its f-structural information (**MostInformative**$_f$), and its semantic information (**MostInformative**$_s$)
   (b) A constraint, **MostSpecific**, which prefers a VI based on information in its exponent (i.e., its v-structure)

- The main distinction between L$_R$FG and standard LFG is that, in order to adopt the morphemic and realizational assumptions of DM, L$_R$FG gives up LFG's Strong Lexicalism (Chomsky 1970, Lapointe 1980, Bresnan et al. 2016). In particular:

1. The terminal nodes of L$_R$FG c-structure are labelled with a category (as in standard LFG) and are paired with a f(unctional)-description (as in standard LFG), but contain no information about *form*. This is instead captured in L$_R$FG by *exponence* (in v-structure) and *realization* (the rest of the MPP path). In other words, the terminal nodes contain information about the exponendum which must be matched against exponenda in the left-hand side of VIs in the Vocabulary for exponence.
2. The terminal nodes of L$_R$FG c-structures map to *morphemes* (more precisely, *allomorphs*), not *words*. L$_R$FG assumes an operation known in the DM literature as *spanning* (Ramchand 2008, Merchant 2015, Svenonius 2016, Haugen and Siddiqi 2016), such that a single morpheme may simultaneously be the exponent of multiple nodes; in other words, the mapping from c-structure to exponence is many-to-one.

- Despite this deviation, much of L$_R$FG's syntax is similar to LFG's. In particular, the division of syntactic information into a structural component (c-structure) and a functional component (f-structure) is important in L$_R$FG.

- Let's look at a simple example from English:

(2) Every dog likes most snacks.

- The L$_R$FG c-structure and f-structure for (2) are show in Figure 2.
  - The exponent v-structures are abbreviated with the English string in a morphemic breakdown. Which c-structure nodes map to which v-structures are shown by arrows marked ν.

- There is no equivalent of LFG's lexicon in DM (Marantz 1997), and this is also true of L$_R$FG, as a variety of DM.

- If there is no lexicon, the question arises: What populates the c-structure terminals? Here we adapt the standard LFG answer: annotated phrase-structure rules license the c-structure terminals.

- Here are the rules for the c-structure (and its correspondence to f-structure) in Figure 2:

(3)

a. $$\text{IP} \rightarrow \begin{array}{c}\text{DP}\\ (\uparrow \textsc{subj}) = \downarrow\end{array} \quad \begin{array}{c}\text{I}'\\ \uparrow = \downarrow\end{array}$$

b. $$\text{VP} \rightarrow \begin{array}{c}\text{V}\\ \uparrow = \downarrow\end{array} \quad \begin{array}{c}\text{DP}\\ (\uparrow \textsc{obj}) = \downarrow\end{array}$$

c. $$\text{vP} \rightarrow \begin{array}{c}\sqrt{}\\ \uparrow = \downarrow\\ @\textsc{root}(\_\_)\end{array} \quad \begin{array}{c}\text{v}\\ \uparrow = \downarrow\end{array}$$

d. $$\text{AgrP} \rightarrow \begin{array}{c}\text{vP}\\ \uparrow = \downarrow\end{array} \quad \begin{array}{c}\text{Agr}\\ \uparrow = \downarrow\\ @\textsc{agr}!\end{array}$$

e. $$\text{DP} \rightarrow \begin{array}{c}\text{D}\\ \uparrow = \downarrow\\ @\textsc{det}!\end{array} \quad \begin{array}{c}\text{NP}\\ \uparrow = \downarrow\end{array}$$

f. $$\#\text{P} \rightarrow \begin{array}{c}\text{nP}\\ \uparrow = \downarrow\end{array} \quad \begin{array}{c}\#\\ \uparrow = \downarrow\\ @\textsc{num}!\end{array}$$

g. $$\text{nP} \rightarrow \begin{array}{c}\sqrt{}\\ \uparrow = \downarrow\\ @\textsc{root}(\_\_)\end{array} \quad \begin{array}{c}\text{n}\\ \uparrow = \downarrow\end{array}$$

- Note that, as it happens, these rules all license binary branching trees, but L_RFG is not committed to binary branching.
- The standard adherence to binary branching in DM is not a feature of that framework per se either, but rather derives from DM's usual assumption of a Minimalist syntax, which in turns assumes a binary Merge operation (Chomsky 1995).
- Thus, L_RFG c-structures can in principle be n-ary branching (Everdell et al. 2021), just like in LFG. Also as in LFG, L_RFG allows c-structure nodes to be optional if they are not necessary for other aspects of well-formedness in the architecture.
- L_RFG rules make heavy use of LFG's templates (Dalrymple et al. 2004, Asudeh et al. 2013), which we call *macros* instead (*template* has a different usage in the DM literature, which could be potentially confusing).
- There are three major kinds of macros in the rules in (3).

1. The ROOT macro individuates roots by their PRED values, which are uniquely instantiated *semantic forms*, as in LFG.
   - Root individuation is a big topic in DM (see Harley 2014 and replies), because roots are generally assumed to be category-less, such that the root itself does not determine its category (Marantz 1997; also see Borer 2005a,b,c for proposals in *Exo-Skeletal Syntax* which have also been influential in DM).
   - DM assumes that the categorial status of a root is determined by *categorizers*, such as v ('little v') and n ('little n') in Figure 2.
   - L_RFG's solution to root individuation is that *roots are individuated by their* PRED*s*.
   - Note that the argument of the ROOT macro is underspecified, which means that any of the language's possible PRED values can satisfy it.
   - A rule containing this kind of underspecification is in effect a *metarule* (Gazdar et al. 1985) over the possible instantiated versions of the rule.
   - The possible instantiations of PRED values are listed using the LIST macro, which is also how the theory handles arbitrary morphological facts, such as declension and conjugation classes in languages that feature these. I do not show the LIST macro here, but further details are forthcoming in Asudeh et al. (2024).

2. The second kind of macros featured above are L$_R$FG's *bang macros*, which enumerate choices.
   - For example, the DET! (pronounced *det-bang*) macro in (3b) lists the possible basic quantificational determiners in English.[1]
   - The other bang macros are not particularly interesting, because of English's relative morphological impoverishment and its consequent lack of many options for number (NUM!) and agreement (AGR!), but they do present an opportunity for talking about a couple of other aspects of the L$_R$FG framework.
   - The theory makes use of (Paninian) *blocking* and uses underspecification to capture the Elsewhere Condition.
   - Thus, specification of plural by NUM! as the only option is legitimate because singular number is assumed to be the underspecified, elsewhere case (at least for English). Singular is the *lack* of plural marking and is unmarked.
   - Notice that L$_R$FG uses the terms *marked* and *unmarked* in the traditional morphological way, meaning to be or to not be marked (i.e., have morphophonological content). We do not use 'marked' to mean 'odd,' 'typologically rare,' 'infrequent' or in any of the other ways that the term has been stretched to cover.
   - Similarly, the English AGR macro can be defined to be about just one case, third singular present tense subject agreement, since it's the only case of (regular) agreement marking in the language.
   - The agreement macro in turns calls the 3SG macro.
3. This last macro is the third kind of macro. It does not represent choice (unlike bang macros) or individuate roots, but is rather just a simple packaging of recurrent information. This is the most basic kind of macro. It also provides an opportunity to observe that macros can call other macros.

- The macros used in (3) are these:

(4)
a. ROOT(X) := $(\uparrow$ PRED$)$ = X
b. DET! := $(\uparrow$ SPEC PRED$)$ = { 'every' | 'some' | 'most' | ... }
c. NUM! := $(\uparrow$ NUM$)$ = PL
d. AGR! := @3SG
e. 3SG := $(\uparrow$ SUBJ PERS$)$ = 3
$(\uparrow$ SUBJ NUM$)$ = SG

- The semantics of macro invocation is very simple: each @ call of a macro simply substitutes the macro's definition (on the right-hand side of the := symbols above) at the invocation point.

[1]The complex determiners would of course be built up around these compositionally. Note that I have specified the @DET! macro here as a disjunctive list of choices of PRED values for a SPEC. As mentioned above, this would now be taken care of by a call to the LIST macro.

- The last element that we need to complete this illustrative analysis is the relevant subset of English's Vocabulary:

(5) a. $\langle$ [D], $(\uparrow \textsc{pred}) = \text{`every'}$ $\rangle \xrightarrow{\nu}$ [*every*]
$(\textsc{spec}\ \uparrow)$
$\lambda P \lambda Q.\textbf{every}(P)(Q)$:
$[((\textsc{spec}\ \uparrow)_\sigma\ \textsc{var}) \multimap ((\textsc{spec}\ \uparrow)_\sigma\ \textsc{restr})] \multimap$
$\forall X.[(\textsc{spec}\ \uparrow)_\sigma \multimap X] \multimap X$

b. $\langle$ [$\sqrt{\ }$], $(\uparrow \textsc{pred}) = \text{`dog'}$ $\rangle \xrightarrow{\nu}$ [*dog*]
$\textbf{dog} : (\uparrow_\sigma\ \textsc{var}) \multimap (\uparrow_\sigma\ \textsc{restr})$

c. $\langle$ [$\sqrt{\ }$], $(\uparrow \textsc{pred}) = \text{`like'}$ $\rangle \xrightarrow{\nu}$ [*like*]
$\textbf{like} : (\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$

d. $\langle$ [Agr], $(\uparrow \textsc{subj pers}) = 3$ $\rangle \xrightarrow{\nu}$ [-*s*]
$(\uparrow \textsc{subj num}) = \textsc{sg}$

e. $\langle$ [D], $(\uparrow \textsc{pred}) = \text{`most'}$ $\rangle \xrightarrow{\nu}$ [*most*]
$(\textsc{spec}\ \uparrow)$
$\lambda P \lambda Q.\textbf{most}(P)(Q)$:
$[((\textsc{spec}\ \uparrow)_\sigma\ \textsc{var}) \multimap ((\textsc{spec}\ \uparrow)_\sigma\ \textsc{restr})] \multimap$
$\forall X.[(\textsc{spec}\ \uparrow)_\sigma \multimap X] \multimap X$

f. $\langle$ [$\sqrt{\ }$], $(\uparrow \textsc{pred}) = \text{`snack'}$ $\rangle \xrightarrow{\nu}$ [*snack*]
$\textbf{snack} : (\uparrow_\sigma\ \textsc{var}) \multimap (\uparrow_\sigma\ \textsc{restr})$

g. $\langle$ [#], $(\uparrow \textsc{num}) = \textsc{pl}$ $\rangle \xrightarrow{\nu}$ [-*s*]
$\textbf{Pl} : [(\uparrow_\sigma\ \textsc{var}) \multimap (\uparrow_\sigma\ \textsc{restr})] \multimap [(\uparrow_\sigma\ \textsc{var}) \multimap (\uparrow_\sigma\ \textsc{restr})]$

- The general form of an L_RFG Vocabulary Item is as follows (Asudeh et al. 2023):

(6) $\langle$ $[C_1,\ldots,C_n]$, $F \cup G \cup I$ $\rangle \xrightarrow{\nu} [\quad]_{\textit{v-structure}}$
distribution *function/meaning*

- A vocabulary item is a mapping from an exponendum, the left-hand side, to an exponent, the right-hand side (again, these are abbreviated here).

- The exponendum consists of two parts:

1. A *list* of categories, minimally (and typically) of length one. However, an item may be listed in the Vocabulary as *spanning* more than one category, so the list is allowed to be longer in principle.
2. A set F of f-structural information (an f-description), a set G of meaning information, modelled as a set of Glue meaning constructors, and a set I of i-structural information. These are presented in a union of sets which we call a *fugui* (pronounced "foo-gooey"), based on the resemblance of the set-theoretic union term to this nonce word of English.

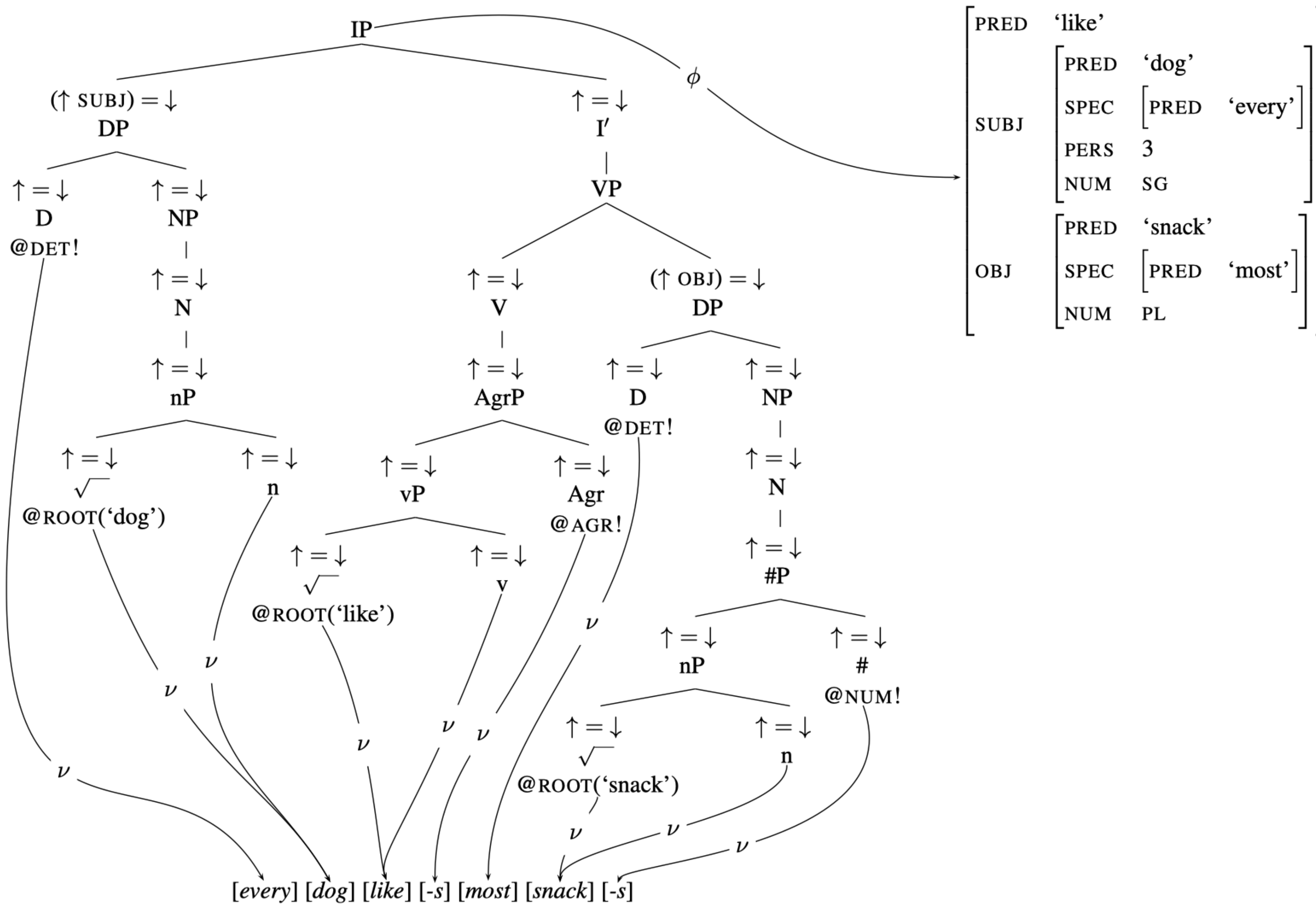
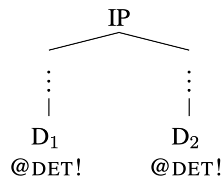


Figure 2: C-structure and f-structure for English *Every dog likes most snacks*

1.3 Exponenda and Vocabulary lookup

- Let us look at the nodes labelled D in Figure 2. The content of these nodes is repeated here:

(7)



I have arbitrarily identified these as nodes 1 and 2 in the c-structure.

- As part of the contentful grammatical vocabulary of the language, determiners have PREDs, but they are not roots, since there is no evidence that determiners are labile in the way roots are.
  - Thus, they are not associated with the ROOT macro by the rule (3e) but rather with the DET! macro.
  - This macro was defined as follows above, repeated here:

  (4b) DET! := $(\uparrow$ SPEC PRED$)$ = { 'every' | 'some' | 'most' | … }

  - The vocabulary items for *every* and *most* are repeated here:

  (5) a. $\langle$ [D], $(\uparrow$ PRED$)$ = 'every' $\rangle \xrightarrow{\nu}$ [*every*]
  (SPEC $\uparrow$)
  $\lambda P \lambda Q.\textbf{every}(P)(Q)$:
  $[((\text{SPEC} \uparrow)_\sigma \text{ VAR}) \multimap ((\text{SPEC} \uparrow)_\sigma \text{ RESTR})] \multimap$
  $\forall X.[(\text{SPEC} \uparrow)_\sigma \multimap X] \multimap X$

  e. $\langle$ [D], $(\uparrow$ PRED$)$ = 'most' $\rangle \xrightarrow{\nu}$ [*most*]
  (SPEC $\uparrow$)
  $\lambda P \lambda Q.\textbf{most}(P)(Q)$:
  $[((\text{SPEC} \uparrow)_\sigma \text{ VAR}) \multimap ((\text{SPEC} \uparrow)_\sigma \text{ RESTR})] \multimap$
  $\forall X.[(\text{SPEC} \uparrow)_\sigma \multimap X] \multimap X$

  - The question is, how do we map the information in the nodes to the correct instance of a determiner VI in each case?
  - In earlier versions of L_RFG(Melchin et al. 2020, Asudeh and Siddiqi 2022, 2023: e.g.,), we postulated a *bridging function*, Φ ('big phi') between v-structure and f-structure so that v-structure could be sensitive to f-structural content.
    - However, in recent work we have dispensed with this usage (Asudeh et al. 2023). Here I propose to bring big phi back as part of Vocabulary lookup, i.e. how information in c-structure terminal nodes is matched against potential vocabulary items.
  - Let us propose a matching function, μ, which we can define as follows:

  (8) For all nodes $n \in T$, where T is the set of terminal nodes in a c-structure,
  $\mu(n) = \{\langle x, y\rangle \mid x = \lambda(n) \wedge \phi(n) = y\}$

    - In other words, μ returns a pair consisting of a category—the label of μ's argument, n, per LFG's usual labelling function, λ (Kaplan 1989)—and an f-structure constructed by applying LFG's standard c-structure/f-structure correspondence function, ϕ, to n.
    - This f-structure is further populated and constrained by the f-structural information associated with n.
    - In this case, this is a disjunction defined by DET!. However, this is the same disjunction in both instances of D in Figure 2, as emphasized in (7) above. Thus, in both cases, the same set is delivered by μ:

(9) $\mu(1) = \mu(2) =$

$$\left\{ \left\langle \text{D}, \begin{bmatrix} \text{SPEC} & \begin{bmatrix} \text{PRED} & \text{`every'} \end{bmatrix} \end{bmatrix} \right\rangle, \left\langle \text{D}, \begin{bmatrix} \text{SPEC} & \begin{bmatrix} \text{PRED} & \text{`some'} \end{bmatrix} \end{bmatrix} \right\rangle, \left\langle \text{D}, \begin{bmatrix} \text{SPEC} & \begin{bmatrix} \text{PRED} & \text{`most'} \end{bmatrix} \end{bmatrix} \right\rangle, \ldots \right\}$$

- Note that μ is in essence just defining how *lexical insertion* works (but where the Vocabulary is an inert 'listicon' that does not contain any operations of its own, such as lexical rules).
- There is no substantive difference here between using just the category for choosing suitable lexical items, as in standard DM or LFG.
- Standard LFG treats the terminal node as the lexical item for *every* or *most*, which includes a matching category for lexical insertion (given the c-structure rule), an f-description, a meaning constructor, and a form. L_RFG treats the terminal node as (the same) bundle of information *except form*, such that the information is mapped to an exponent that specifies further constraints on the realization of the form.

- In order to match the f-structure in the second coordinates of (9), we now use big phi, Φ. We define Φ as follows:

(10) Given some vocabulary item, vi, such that $vi = \langle\langle \ldots, \text{F} \cup \text{G} \cup \text{I} \rangle, \ldots\rangle$,
$\Phi(vi)$ is the minimal f-structure that satisfies F.

- Note that given the ν-mapping in (6), the Vocabulary is a set of vocabulary items such that each member of the set is a pair consisting of an exponendum, the left-hand side of a vocabulary item, and an exponent, the right-hand side of the vocabulary item; i.e., when we treat VIs set-theoretically, they are pairs consisting of an exponendum, which is itself a pair, and an exponent.
- In short, Φ does the same thing as ϕ, but whereas ϕ takes a c-structure node as an argument and returns the minimal f-structure satisfying its associated f-description, Φ takes a vocabulary item as an argument and returns the minimal f-structure satisfying the F in the vocabulary item's fugui.

- The output of ϕ applied to each of the nodes labelled D in (7), i.e. node 1 and node 2, is such that they can respectively correspond to the following f-structures, arbitrarily labelled as f and g:

(11) a. $f\begin{bmatrix} \text{SPEC} & \begin{bmatrix} \text{PRED} & \text{`every'} \end{bmatrix} \end{bmatrix}$

b. $g\begin{bmatrix} \text{SPEC} & \begin{bmatrix} \text{PRED} & \text{`most'} \end{bmatrix} \end{bmatrix}$

- We can now define the **MostInformative**$_f$ constraint (**MostInformative** in f-structural terms) can now be defined as follows:

(12) **MostInformative**$_f$
Given a c-structure node n that is the target of exponence and two Vocabulary Items, α and β,

$$\textbf{MostInformative}_f(\alpha, \beta) = \begin{cases} \alpha \textbf{ if } \exists t \exists f \forall g. f = \Phi(\alpha) \wedge g = \Phi(\beta) \wedge t \in \phi(n) \wedge f, g \sqsubseteq t \wedge g \sqsubset f \\ \beta \textbf{ if } \exists t \exists f \forall g. f = \Phi(\beta) \wedge g = \Phi(\alpha) \wedge t \in \phi(n) \wedge f, g \sqsubseteq t \wedge g \sqsubset f \\ \bot \textbf{ otherwise} \end{cases}$$

- The constraint returns one of three cases.
- In the two first cases, we check whether there are two VIs that each subsume the f-structure of the target node (f-structure t).

- This allows the exponenda in the VIs to be more general than the one they are trying to match (capturing DM's Subset Principle) but does not allow them to be inconsistent with the target's f-structure; this is modelled by subsumption on f-structures ($\sqsubseteq$; Bresnan et al. 2016).
- The last term in each of the first two cases checks which of the two VI's f-descriptions creates an f-structure that contains strictly more information than the other; this is modelled by proper subsumption on f-structures.

- In other words, **MostInformative**$_f$ checks whether there is a vocabulary item whose corresponding f-structure subsumes the target f-structure and which is most informative with respect to other candidate VIs for the target.
- It returns whichever VI is most informative with respect to f-structure.
- If there are no such candidates or if more than one candidate fits the bill, the constraint instead *borks* and returns $\bot$: there is no *most* informative VI.

- In the case of the f-structures in (11), neither subsumes (never mind *properly* subsumes) the other, because they have distinct PREDs.
  - Therefore, **MostInformative**$_f$ chooses neither over the other and borks. It is important to note that borking means the constraint is indecisive, not violated.
  - Thus, one available instantiation of the relevant c-structure is the one in Figure 2; thus the relevant sentence gets a successful parse and is deemed grammatical.
- In contrast to our previous approach, though, the Glue meaning constructors do not populate the terminal node, but rather just come along for the ride with the VIs that match the c-structural information (category) and f-structural information (the minimal f-structure satisfying the f-description) on the target c-structure node.
- In effect, we have stretched the function cascade in (1) as follows:

(1′) $$\underbrace{\textit{target of exponence}}_{\textit{c-structure}} \xrightarrow{\mu} \underbrace{\textit{exponendum} \xrightarrow{\nu} \textit{exponent}}_{\textit{Vocabulary}} \xrightarrow{o\,\circ\,\rho} \underbrace{\textit{realization}}_{\textit{form}}$$

  - This in turn means that the ν-annotated arrows in L$_{\mathrm{R}}$FG diagrams like Figure 2 should really be labelled $\nu \circ \mu$. However, to avoid clutter, we can take the μ mapping as read and continue to label the arrows with just ν.
- As I mentioned above, the Glue meaning constructors in this approach do not populate terminal nodes in c-structure, the targets of exponence, but are rather associated directly with exponenda in the Vocabulary.
  - The meaning constructors, once instantiated to the arbitrary mnemonic labels added to the f-structure in Figure 2, as in (13), can construct the usual two possible Glue proofs (one corresponding to surface scope, *every* > *most*, and the other to inverse scope, *most* > *every*); the two possibilities are shown in Figures 3 and 4.

(13) $$l\begin{bmatrix} \text{PRED} & \text{‘like’} \\ \text{SUBJ} & d\begin{bmatrix} \text{PRED} & \text{‘dog’} \\ \text{SPEC} & \begin{bmatrix} \text{PRED} & \text{‘every’} \end{bmatrix} \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \end{bmatrix} \\ \text{OBJ} & s\begin{bmatrix} \text{PRED} & \text{‘snack’} \\ \text{SPEC} & \begin{bmatrix} \text{PRED} & \text{‘most’} \end{bmatrix} \\ \text{NUM} & \text{PL} \end{bmatrix} \end{bmatrix}$$

$$
\dfrac{\lambda P\lambda Q.\mathbf{every}(P)(Q) : (v_1 \multimap r_1) \multimap \forall X.(d_\sigma \multimap X) \multimap X \qquad \mathbf{dog} : (v_1 \multimap r_1)}{\lambda Q.\mathbf{every}(\mathbf{dog})(Q) : \forall X.(d_\sigma \multimap X) \multimap X}
$$

$$
\dfrac{\mathbf{snack} : (v_2 \multimap r_2) \qquad \mathbf{Pl} : (v_2 \multimap r_2) \multimap (v_2 \multimap r_2)}{\mathbf{Pl}(\mathbf{snack}) : (v_2 \multimap r_2)}
$$

$$
\dfrac{\lambda P\lambda Q.\mathbf{most}(P)(Q) : (v_2 \multimap r_2) \multimap \forall X.(s_\sigma \multimap X) \multimap X \qquad \mathbf{Pl}(\mathbf{snack}) : (v_2 \multimap r_2)}{\lambda Q.\mathbf{most}(\mathbf{Pl}(\mathbf{snack}))(Q) : \forall X.(s_\sigma \multimap X) \multimap X}
$$

$$
\dfrac{\mathbf{like} : s_\sigma \multimap d_\sigma \multimap l_\sigma}{\lambda y_1 \lambda x_1.\mathbf{like}(y_1)(x_1) : s_\sigma \multimap d_\sigma \multimap l_\sigma}\Rightarrow_\eta
$$

$$
\dfrac{\lambda y_1 \lambda x_1.\mathbf{like}(y_1)(x_1) : s_\sigma \multimap d_\sigma \multimap l_\sigma \qquad [y : s_\sigma]^1}{\lambda x_1.\mathbf{like}(y)(x_1) : d_\sigma \multimap l_\sigma}\multimap_{\mathcal{E}}, \Rightarrow_\beta
$$

$$
\dfrac{\lambda x_1.\mathbf{like}(y)(x_1) : d_\sigma \multimap l_\sigma \qquad [x : d_\sigma]^2}{\mathbf{like}(y)(x) : l_\sigma}\multimap_{\mathcal{E}}, \Rightarrow_\beta
$$

$$
\dfrac{\mathbf{like}(y)(x) : l_\sigma}{\lambda y.\mathbf{like}(y)(x) : s_\sigma \multimap l_\sigma}\multimap_{\mathcal{I},1}
$$

$$
\dfrac{\lambda Q.\mathbf{most}(\mathbf{Pl}(\mathbf{snack}))(Q) : \forall X.(s_\sigma \multimap X) \multimap X \qquad \lambda y.\mathbf{like}(y)(x) : s_\sigma \multimap l_\sigma}{\mathbf{most}(\mathbf{Pl}(\mathbf{snack}))(\lambda y.\mathbf{like}(y)(x)) : l_\sigma}\multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[l_\sigma/X], \Rightarrow_\beta
$$

$$
\dfrac{\mathbf{most}(\mathbf{Pl}(\mathbf{snack}))(\lambda y.\mathbf{like}(y)(x)) : l_\sigma}{\lambda x.\mathbf{most}(\mathbf{Pl}(\mathbf{snack}))(\lambda y.\mathbf{like}(y)(x)) : d_\sigma \multimap l_\sigma}\multimap_{\mathcal{I},2}
$$

$$
\dfrac{\lambda Q.\mathbf{every}(\mathbf{dog})(Q) : \forall X.(d_\sigma \multimap X) \multimap X \qquad \lambda x.\mathbf{most}(\mathbf{Pl}(\mathbf{snack}))(\lambda y.\mathbf{like}(y)(x)) : d_\sigma \multimap l_\sigma}{\mathbf{every}(\mathbf{dog})(\lambda x.\mathbf{most}(\mathbf{Pl}(\mathbf{snack}))(\lambda y.\mathbf{like}(y)(x))) : l_\sigma}\multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[l_\sigma/X], \Rightarrow_\alpha
$$

Figure 3: Surface scope (*every* > *most*)

$$
\dfrac{\mathbf{snack} : (v_2 \multimap r_2) \qquad \mathbf{Pl} : (v_2 \multimap r_2) \multimap (v_2 \multimap r_2)}{\mathbf{Pl}(\mathbf{snack}) : (v_2 \multimap r_2)}
$$

$$
\dfrac{\lambda P\lambda Q.\mathbf{most}(P)(Q) : (v_2 \multimap r_2) \multimap \forall X.(s_\sigma \multimap X) \multimap X \qquad \mathbf{Pl}(\mathbf{snack}) : (v_2 \multimap r_2)}{\lambda Q.\mathbf{most}(\mathbf{Pl}(\mathbf{snack}))(Q) : \forall X.(s_\sigma \multimap X) \multimap X}
$$

$$
\dfrac{\lambda P\lambda Q.\mathbf{every}(P)(Q) : (v_1 \multimap r_1) \multimap \forall X.(d_\sigma \multimap X) \multimap X \qquad \mathbf{dog} : (v_1 \multimap r_1)}{\lambda Q.\mathbf{every}(\mathbf{dog})(Q) : \forall X.(d_\sigma \multimap X) \multimap X}
$$

$$
\dfrac{\mathbf{like} : s_\sigma \multimap d_\sigma \multimap l_\sigma}{\lambda y_1 \lambda x.\mathbf{like}(y_1)(x) : s_\sigma \multimap d_\sigma \multimap l_\sigma}\Rightarrow_\eta
$$

$$
\dfrac{\lambda y_1 \lambda x.\mathbf{like}(y_1)(x) : s_\sigma \multimap d_\sigma \multimap l_\sigma \qquad [y : s_\sigma]^1}{\lambda x.\mathbf{like}(y)(x) : d_\sigma \multimap l_\sigma}\multimap_{\mathcal{E}}, \Rightarrow_\beta
$$

$$
\dfrac{\lambda Q.\mathbf{every}(\mathbf{dog})(Q) : \forall X.(d_\sigma \multimap X) \multimap X \qquad \lambda x.\mathbf{like}(y)(x) : d_\sigma \multimap l_\sigma}{\mathbf{every}(\mathbf{dog})(\lambda x.\mathbf{like}(y)(x)) : l_\sigma}\multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[l_\sigma/X], \Rightarrow_\beta
$$

$$
\dfrac{\mathbf{every}(\mathbf{dog})(\lambda x.\mathbf{like}(y)(x)) : l_\sigma}{\lambda y.\mathbf{every}(\mathbf{dog})(\lambda x.\mathbf{like}(y)(x)) : s_\sigma \multimap l_\sigma}\multimap_{\mathcal{I},1}
$$

$$
\dfrac{\lambda Q.\mathbf{most}(\mathbf{Pl}(\mathbf{snack}))(Q) : \forall X.(s_\sigma \multimap X) \multimap X \qquad \lambda y.\mathbf{every}(\mathbf{dog})(\lambda x.\mathbf{like}(y)(x)) : s_\sigma \multimap l_\sigma}{\mathbf{most}(\mathbf{Pl}(\mathbf{snack}))(\lambda y.\mathbf{every}(\mathbf{dog})(\lambda x.\mathbf{like}(y)(x))) : l_\sigma}\multimap_{\mathcal{E}}, \forall_{\mathcal{E}}[l_\sigma/X], \Rightarrow_\alpha
$$

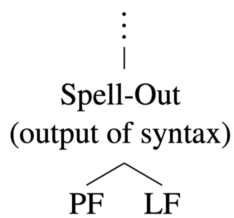
Figure 4: Inverse scope (*most* > *every*)

- It is important to note, though, that this also continues to allow $\mathrm{L_R}$FG to have a sensible notion of *morphosemantics*, i.e. a direct morphology-semantics interface, because exponence in the Vocabulary is a function on a bundle of information that includes semantics.

- Thus, meaning can condition exponence.

- This is not really possible in the classic 'Y model' that DM inherits from minimalism, because Phonological Form (PF) and Logical Form (LF) merely interpret the output of syntax and PF is never conditioned on LF directly:

(14)



- I next turn to a reconsideration of morphosemantics in $\mathrm{L_R}$FG in light of the moves made above.

1.4 A refinement of morphosemantics and semantic blocking in $\mathrm{L_R}$FG

- Let us define **MostInformative**$_s$, the semantic version of **MostInformative**, preliminary as follows:

(15) **MostInformative**$_s$ (preliminary)
Given two Vocabulary Items, α and β, such that $\alpha = \langle\langle \ldots, \mathrm{F}_1 \cup \mathrm{G}_1 \cup \mathrm{I}_1\rangle, \ldots\rangle$ and $\beta = \langle\langle \ldots, \mathrm{F}_1 \cup \mathrm{G}_2 \cup \mathrm{I}_1\rangle, \ldots\rangle$,

$$\textbf{MostInformative}_s(\alpha,\beta) = \begin{cases} \alpha \textbf{ if } [\![\mathrm{G}_1]\!] \subset [\![\mathrm{G}_2]\!] \\ \beta \textbf{ if } [\![\mathrm{G}_2]\!] \subset [\![\mathrm{G}_1]\!] \\ \perp \textbf{ otherwise} \end{cases}$$

  - Semantic informativeness is captured as entailment in the model.
  - If VI α's meaning (proof conclusion) entails β's meaning, then choose α. If the reverse holds, choose β. Otherwise, **MostInformative**$_s$ borks and the constraint is indecisive.

- Even in this preliminary version, **MostInformative**$_s$ solves a puzzle for DM.
  - Namely, if two VIs are such that one is semantically contentless (its G set is empty) and the other is not, but they are in competition as the exponenda of a single c-structural locus, **MostInformative**$_s$ will always prefer the contentful option.
  - A relevant case occurs in English, where third singular agreement loses out to tense marking.
  - In other words, assuming that the past tense marker [-*d*] and the third singular marker [-*s*] are vying to be exponents of a single c-structure node, it is potentially a puzzle for standard DM as to why we get (16) as the expression of past tense in English rather than (17).

(16) Max yawned. $=$ past

(17) Max yawns. $\neq$ past

- This is a puzzle in standard DM because exponence is not conditioned by meaning, as discussed above.
- But the past tense morpheme [-*d*] in English states information about tense, which is contentful, but third singular agreement does not add any content of its own.
- Therefore, **MostInformative**$_s$ will always pick [-*d*] to mark the past tense rather than [-*s*]. Another example is cases where an exponendum for number is in competition with an exponendum for gender.
- Grammatical gender is not contentful, but grammatical number (at least for plurals). Therefore, the VI expressing number will be preferred over the VI expressing gender.

- A deeper consideration of **MostInformative**$_s$ concerns competition between two contentful VIs.
  - That is, what if we compare two VIs that share a category and make no f-structural or i-structural distinctions from each other, but where the meaning computed for one VI entails the other? Such a case is provided by the inchoative/causative alternation in English (see also, Asudeh 2021), as in:

(18) On special occasions, the vase <u>sits</u> on the shelf. inchoative

(19) On special occasions, Max <u>sets</u> the vase on the shelf. causative

  - I have used the verbs *sit*/*set* here purposefully, because some DM analyses treat this as a case of *suppletion* with an external causativizing functional head. However, Siddiqi (2024) has argued convincingly against this sort of analysis.

- Reasonable VIs for *sit* and *set* (recall that the present tense inflection is handled by Agr) are shown in (20) and (21). In this case, some event structure is useful in the meaning language, so I have adopted a version of event semantics:

(20) $$\left\langle\ [\sqrt{\ }],\ \begin{array}{l}(\uparrow \textsc{pred}) = \text{‘sit’} \\ \lambda x \lambda e.\mathbf{sit}(e) \wedge \mathbf{patient}(e) = x : \\ (\uparrow \textsc{subj})_\sigma \multimap (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma \end{array} \right\rangle \xrightarrow{\nu} [\textit{sit}]$$

(21) $$\left\langle\ [\sqrt{\ }],\ \begin{array}{l}(\uparrow \textsc{pred}) = \text{‘sit’} \\ \lambda y \lambda x \lambda e.\mathbf{sit}(e) \wedge \mathbf{patient}(e) = y \wedge \mathbf{agent}(e) = x : \\ (\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma \end{array} \right\rangle \xrightarrow{\nu} [\textit{set}]$$

- Observe that the following entailment holds (by the standard logical rule of conjunction simplification):

(22) $$\frac{\mathbf{sit}(e) \wedge \mathbf{patient}(e) = y \wedge \mathbf{agent}(e) = x}{\mathbf{sit}(e) \wedge \mathbf{patient}(e) = x}\ \textsc{cs}$$

- The linear logic terms are irrelevant to this entailment pattern.
  - Nevertheless, there is also another factor at play here. All else being equal, **MostInformative**$_s$ should *always* prefer $[\![\textit{set}]\!]$ over $[\![\textit{sit}]\!]$.
  - As far as the grammar is concerned, this is true, but grammar is not the only consideration in forming a linguistic expression. The speaker's intentions also matter.
  - That is, presumably the speaker has some conceptual *message* in mind (Levelt 1989) that they wish to encode linguistically.
  - The message is not part of the grammar per se, but the speaker must pick the right VIs to encode the intended message grammatically.

- Thus, if the speaker has a message about someone causing something to happen, they would presumably pick the causative VI.
- If they wish to leave the causer out, they will presumably pick the inchoative VI.
- In sum, the relationship between the message and the grammar is not a purely linguistic one: it involves the interface between thought and grammar.
- The hearer's job is much easier: they actually hear the speaker's expression and the job of the grammar in this direction is to pull up the right VIs to parse the expression.
- If the arguments present in the expression are such that only the inchoative makes sense (i.e., there is no causer mentioned in the expression and the subject is a patient), then the hearer must simply pick the right VI to accomplish the task.
- But this in turn makes it seem that **MostInformative**$_s$ is simply switched off in parsing. This is undesirable, because we do not wish to postulate different grammars for production and parsing. In othwr words, we want grammar, as a knowledge system, to be *process neutral* (Pollard and Sag 1994).

- Despite their irrelevance to the entailment pattern in the meaning language, we can in fact leverage the resource sensitivity of the Glue terms (Asudeh 2012) to our advantage to make precise how **MostInformative**$_s$ 1) can be sensitive to an intended message on the speaker's part (without making messages part of grammar) and 2) not have to be switched off on the hearer's part. Let us revise **MostInformative**$_s$ as follows:

(23) **MostInformative**$_s$ (revised)
Given an expression Σ such that its set of Vocabulary Items, V_Σ, contains two Vocabulary Items, α and β, where $\alpha = \langle\langle \ldots, \mathrm{F}_1 \cup \mathrm{G}_1 \cup \mathrm{I}_1 \rangle, \ldots\rangle$ and $\beta = \langle\langle \ldots, \mathrm{F}_1 \cup \mathrm{G}_2 \cup \mathrm{I}_1 \rangle, \ldots\rangle$,

$$\top(\textbf{Proof}(\mathrm{V}_\Sigma)) \rightarrow \textbf{MostInformative}_s(\alpha, \beta) = \begin{cases} \alpha \textbf{ if } [\![\mathrm{G}_1]\!] \subset [\![\mathrm{G}_2]\!] \\ \beta \textbf{ if } [\![\mathrm{G}_2]\!] \subset [\![\mathrm{G}_1]\!] \\ \bot \textbf{ otherwise} \end{cases}$$

- The symbol $\top$ is here used as a function that returns true if the set of proofs it takes as an argument (possibly a singleton) is such that the proofs all terminate successfully (i.e., every proof in the set reaches its goal condition; Asudeh 2012).
- The function **Proof** calculates the set of valid (per the fragment of linear logic that is the Glue logic) proofs that can be calculated from the expression's vocabulary items, V_Σ. We state this as a precondition on **MostInformative**$_s$, such that if the condition is false, the constraint succeeds vacuously.
- In sum, if the multiset of Glue terms gathered from the expression's VIs are such that not linear conditionals can be successfully eliminated, then all bets are off with respect to **MostInformative**$_s$.

2 Expletives in L$_\mathrm{R}$FG

- I take an expletive to be an element that is definitionally something that lacks meaning.
- In this framework, this means that an expletive lacks a meaning constructor.
- I will here focus on English expletives. These present two main challenges:

1. There are two expletives, *it* and *there*:

   (24) It rained.

   (25) There is a student outside.

- If exponence is defined in terms of a function, how can two contentless things map to different exponents?

2. Copy raising (see, e.g., Asudeh and Toivonen 2012, Asudeh 2012 and references therein) shows connectivity between an upper expletive and a lower expletive.

(26) It seems like Max is tired.

(27) It seems like it rained.

(28) It seems like there was rain last night.

(29) %There seems like there was rain last night.

(30) *There seems like it rained last night.

- It must be ensured that the potential for this connectivity is preserved.

- I propose the following vocabulary items for English expletives:

(31) $\langle\, [\mathrm{D}], \varnothing \,\rangle \xrightarrow{\nu} [\mathit{it}]$

(32) $\langle\, [\mathrm{D}], (\uparrow_\sigma \textsc{location}) \,\rangle \xrightarrow{\nu} [\mathit{there}]$

  - The first problem, that of exponence is immediately solved: the ν-mapping maps an entirely empty fugui to [*it*], based on the assumption that *it* is the least (featurally) marked English pronoun. In contrast, the ν-mapping maps ($\uparrow_\sigma$ LOCATION) to [*there*].
  - The entry for [*there*] implicitly assumes a semantic structure as in Asudeh and Giorgolo (2012) and much subsequent Glue work since.
  - The statment ($\uparrow_\sigma$ LOCATION) requires there to be a LOCATION (with some value) in the semantic structure.

- What delivers the required information in semantic structure?
  - By hypothesis, it must be the VI for *be*.
  - Let us assume that *be* requires its subject to be a LOCATION (cf. Ramchand (2008) on PATH) with an equation like the following:

(33) $(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{location})$

  - Importantly, the subject does not need to have a meaning constructor for this equation to be satisfied: it just states that whatever is the σ-correspondent of the subject is the value of LOCATION at semantic structure.

- How is the connectivity problem solved?
  - The two expletives have slightly different f-structural content (required for predicates to be able to select the right one).
  - Therefore an *it* f-structure cannot be equated with a *there* f-structure because it would create inconsistency in the value for the feature EXPLETIVE, which is not permitted by the definition of f-structure in LFG.

3 Conclusion

1. How is the form of expletives determined in a lexical-realizational (Stump 2001) approach to morphology, such as Distributed Morphology (Halle and Marantz 1993)?

   - See § 2 above

2. What is the function of expletives in grammar?

   - Syntactically, they fulfill some kind of subject requirement.
   - Semantically, perhaps they are involved in converting an event to a situation (e.g., per Wurmbrand and Lohninger 2024)

3. Can this function be captured in compositional semantics? What if the semantics is *resource-sensitive* (Asudeh 2012, 2022)?

   - Yes in both cases.

References

Asudeh, Ash. 2006. Direct Compositionality and the Architecture of LFG. In Miriam Butt, Mary Dalrymple, and Tracy Holloway King, eds., *Intelligent Linguistic Architectures: Variations on Themes by Ronald M. Kaplan*, 363–387. Stanford, CA: CSLI Publications.

—. 2012. *The Logic of Pronominal Resumption*. Oxford: Oxford University Press.

—. 2021. The Unrealized and the Unheard. In I Wayan Arka, Ash Asudeh, and Tracy Holloway King, eds., *Modular Design of Grammar: Linguistics on the Edge*, 22–42. Oxford University Press.

—. 2022. Glue Semantics. *Annual Review of Linguistics* 8: 321–341.

Asudeh, Ash, Bronwyn Bjorkman, Frances Dowle, Neil Myler, Daniel Siddiqi, and Lisa Sullivan. 2024. Fusional morphology, metasyncretism, and secondary exponence: A morphemic, realizational approach to Latin declension. To be presented at the LFG24 Conference, Accra, Ghana.

Asudeh, Ash, Tina Bögel, and Daniel Siddiqi. 2023. Modelling Exponents. Konstanz: PubliKon. Forthcoming.

Asudeh, Ash, Mary Dalrymple, and Ida Toivonen. 2013. Constructions with Lexical Integrity. *Journal of Language Modelling* 1(1): 1–54.

Asudeh, Ash, and Gianluca Giorgolo. 2012. Flexible Composition for Optional and Derived Arguments. In Miriam Butt and Tracy Holloway King, eds., *Proceedings of the LFG12 Conference*, 64–84. Stanford, CA: CSLI Publications.

Asudeh, Ash, and Daniel Siddiqi. 2022. Realizational Morphosemantics in L_RFG. 20–40. Stanford, CA: CSLI Publications.

—. 2023. Morphology in LFG. In Mary Dalrymple, ed., *Handbook of Lexical Functional Grammar*, 855–901. Berlin: Language Science Press.

Asudeh, Ash, and Ida Toivonen. 2012. Copy Raising and Perception. *Natural Language and Linguistic Theory.* 30(2): 321–380.

Bögel, Tina. 2015. The Syntax-Prosody Interface in Lexical Functional Grammar. Ph.D. thesis, University of Konstanz.

Borer, Hagit. 2005a. *Structuring Sense, Vol. I: In Name Only*. Oxford: Oxford University Press.

—. 2005b. *Structuring Sense, Vol. II: The Normal Course of Events*. Oxford: Oxford University Press.

—. 2005c. *Structuring Sense, Vol. III: Taking Form*. Oxford: Oxford University Press.

Bresnan, Joan, Ash Asudeh, Ida Toivonen, and Stephen Wechsler. 2016. *Lexical-Functional Syntax*. Chichester, UK: Wiley-Blackwell, 2nd edn.

Chomsky, Noam. 1970. Remarks on Nominalization. In Roderick A. Jacobs and Peter S. Rosenbaum, eds., *Readings in English Transformational Grammar*, 184–221. Waltham, MA: Ginn and Company.

—. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

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

Dalrymple, Mary, Ronald M. Kaplan, and Tracy Holloway King. 2004. Linguistic Generalizations over Descriptions. In Miriam Butt and Tracy Holloway King, eds., *Proceedings of the LFG04 Conference*, 199–208. Stanford, CA: CSLI Publications.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell III, and Annie Zaenen, eds. 1995. *Formal Issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications.

Dalrymple, Mary, John Lamping, and Vijay Saraswat. 1993. LFG Semantics via Constraints. In Steven Krauwer, Michael Moortgat, and Louis des Tombe, eds., *Proceedings of the Sixth Conference of the European ACL*, 97–105. Utrecht, The Netherlands: Association for Computational Linguistics.

Dalrymple, Mary, John J. Lowe, and Louise Mycock. 2019. *The Oxford Reference Guide to Lexical Functional Grammar*. Oxford: Oxford University Press.

Dalrymple, Mary, and Irina Nikolaeva. 2011. *Objects and Information Structure*. Cambridge: Cambridge University Press.

Everdell, Michael, Paul B. Melchin, Ash Asudeh, and Daniel Siddiqi. 2021. Beyond c-structure and f-structure: On the argument-adjunct distinction in Odam. In Miriam Butt, Jamie Y. Findlay, and Ida Toivonen, eds., *Proceedings of the LFG21 Conference*, 125–145. Stanford, CA: CSLI Publications.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

Halle, Morris, and Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In Samuel Jay Keyser and Ken Hale, eds., *The View from Building 20*, 111–176. Cambridge, MA: MIT Press.

Harley, Heidi. 2014. On the identity of roots. *Theoretical Linguistics* 40(3–4): 225–276.

Haugen, Jason D., and Daniel Siddiqi. 2016. Towards a restricted realizational theory: Multimorphemic monolistemicity, portmanteaux, and post-linearization spanning. In Siddiqi and Harley 2016, 343–386.

Kaplan, Ronald M. 1987. Three Seductions of Computational Psycholinguistics. In Peter Whitelock, Mary McGee Wood, Harold L. Somers, Rod Johnson, and Paul Bennett, eds., *Linguistic Theory and Computer Applications*, 149–181. London: Academic Press. Reprinted in Dalrymple et al. (1995: 339–367).

—. 1989. The Formal Architecture of Lexical-Functional Grammar. *Journal of Information Science and Engineering* 5: 305–322. Revised version published in Dalrymple et al. (1995: 7–27).

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

Kaplan, Ronald M., and 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: MIT Press. Reprinted in Dalrymple et al. (1995: 29–135).

Lapointe, Steven. 1980. A Theory of Grammatical Agreement. Ph.D. thesis, University of Massachusetts, Amherst.

Levelt, Willem J. M. 1989. *Speaking: From Intention to Articulation*. Cambridge, MA: MIT Press.

Marantz, Alec. 1997. No escape from syntax: Don't try morphological analysis in the privacy of your own lexicon. *University of Pennsylvania Working Papers in Linguistics* 4(2): 201–225.

Melchin, Paul B., Ash Asudeh, and Dan Siddiqi. 2020. Ojibwe Agreement in Lexical-Realizational Functional Grammar. In Miriam Butt and Ida Toivonen, eds., *Proceedings of the LFG20 Conference*, 268–288. Stanford, CA: CSLI Publications.

Merchant, Jason. 2015. How much context is enough? Two cases of span-conditioned stem allomorphy. *Linguistic Inquiry* 46(2): 273–303.

Pollard, Carl, and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago, IL and Stanford, CA: The University of Chicago Press and CSLI Publications.

Ramchand, Gillian Catriona. 2008. *Verb Meaning and the Lexicon: A First Phase Syntax*. Cambridge: Cambridge University Press.

Siddiqi, Daniel. 2024. On the Taxonomy of Root Suppletion. In *Proceedings of the 39th West Coast Conference on Formal Linguistics*. Cascadilla Press. Forthcoming.

Siddiqi, Daniel, and Heidi Harley, eds. 2016. *Morphological Metatheory*. Amsterdam: John Benjamins.

Stump, Gregory T. 2001. *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge: Cambridge University Press.

Svenonius, Peter. 2016. Spans and words. In Siddiqi and Harley 2016, 201–222.

Wurmbrand, Susanne, and Magdalena Lohninger. 2024. An implicational universal in complementation: Theoretical insights and empirical progress. In Jutta M. Hartmann and Angelika Wöllstein, eds., *Propositional Arguments in Cross-Linguistic Research: Theoretical and Empirical Issues*. Tübingen: Gunter Narr Verlag. To appear.