Can Lambda Calculus Capture Construction Grammar

Generative syntax holds as its central feature the fact that syntax is the projection of lexical requirements. Lexical requirements, of which theta roles are an example, both guide and constrain the generative process in the number of elements that they require and the types of elements that can fill their slots. Syntax is therefore highly verb-centric, in that the theta roles encoded in the verbs lexical entry determines the kind and number of elements that follow in the construction. The problem of unbounded generativity is thus simplified to knowing the main verb and the number of arguments it can take. As Golderg points out, however, this places a large burden of information on the lexicon. Most verbs are surprisingly highly versatile and can take multiple numbers of different arguments in different contexts. Many verbs, such as kick, can appear in nearly all contexts. Kick appears in binary, transitive, ternary, and intransitive structures. Since language is dynamic new contexts can appear such that a verb can suddenly function inside of an intransitive structure, when it previously couldn't. Additionally, if we are to approach the lexicon in this way, we must also store semantic structures that occur only peripherally. Take the example of "he sneezed the napkin off the table". Under this approach, since we must store all instances of verbs taking a different number of arguments, inside of sneeze we must store "X causes Y to move Z". The meaning central to sneezing is not causative motion. However, it happens to be able to function in a context where it can be used to initiate causative motion. But obviously sneeze can't function in all the familiar contexts we expect normal causative motions verbs to function in. Since it appears sneeze only functions as a causative verb in relation to this specific

example, storing “X causes Y to move Z " inside its lexicon doesn’t actually work because the statement isn’t general: X,Y, and Z can only be specific words. Moreover, we can’t reasonably think that adding more specificity to its lexical entry -“X causes Y to move Z *iff* Z is a napkin (or Z is a light object)”- would save us from this conundrum. The general takeaway from examples like these is that semantic primitives can’t solely be bound to individual words. It must be the case that unique semantic primitives appear at a higher level, namely the phrasal level. This is where construction grammar comes into play. As its central feature is the fact that it challenges the notion that the semantic locus lies solely at the lexical level. Below is a definition of construction as defined by Adele Golgberg

C is a construction iff C is a form-meaning pair <F,S> such that some aspect of F, or some aspect of S, is not strictly predictable from C’s component parts or from other previously established constructions.

The definition has problems associated with it in regards to learnability. How does one know when to analyze a given utterance using standard grammatical principles and when not to. In a sense such a definition requires one to know all component parts of a language, i.e. their standard uses, before one could accurately say whether the utterance was a construction or not. They would need to know each word's standard meaning to be able to conclude that the composition doesn't fit the desired meaning. Additionally, learning a word for the first time inside of a construction may result in the listener to

presume that other standard utterances containing the same word are constructions when they aren't. Moreover, while it is correct to say that F or S is not strictly predictable from C's component parts, we still must generate an account for how the listener comes to learn the meaning of a construction: simply saying one analogizes from previous constructions doesn't explain how the first construction is learned.

What if we slightly alter the above example, in an effort to determine where the semantic locus of causative motion lies. Take "he sneezed the napkin from the table inside of SouthQuad". Here we do not get the meaning of causative motion, and thus we see that the preposition "off" is clearly central to creating this. If we don't assume a static lexicon, and instead something more combinatorial, we can conceptualize sneezing in "he sneezed the napkin off the table" as "sneezeoff". Under such a framework, we could predict that with significant language change we could see a structure such as "He sneezedoff from the table the napkin inside of SouthQuad".

Lexicon:

<sneezedoff> / <kicked> : <Agent, Theme> or possibly <Agent, Theme, Location>

The reason we likely won't see such a structure is not because sneezed and off shouldn't be treated as one semantic unit, but because the frequency of such a structure is low and therefore the cost of such a change would outweigh the benefits. Nevertheless, your mental representation of the sentence is somehow merging the two elements either way even as they are separate. But again, examples like these illustrate the fact that we need a framework that allows semantic primitives to emerge at levels higher than just the lexicon. The question we must ask, however, is whether the construction is central to this

semantic alteration or the verb itself: or whether even the construction vs. verb distinction is ill conceived. Construction grammar answers this question by taking a position that is essentially the flip side of the lexical based answers inside UG: the length of any entry in the lexicon is bounded by the total number of distinct constructions it can appear inside of.

Construction grammar essentially shifts the burden of argument necessitation onto the construction itself, instead of the verb. It is not that sneeze has an entry of causative motion that necessitates {Y,Z} but it is the construction that allows sneeze to appear inside of it.

Take “He <verb> book off the table”.

(a.) He tipped the book off the table

(b.) *He broke the book off the table

(c.) *He imploded the book off the table

(d.) The broke the branch off tree

There seems to be a much more complex semantic interaction going on between certain elements whose merger or lack thereof can dictate the grammaticality judgements of the sentence. Tipped, broken and imploded all share the meaning of a physical object undergoing damage and yet tipped can only appear with off, as in tippedoff, since tipped implies a direction compatible with the preposition “off” that broke and imploded do not share. This is complicated by the fact that you can say, “he broke off a piece of chocolate to give to his friend”. Thus the semantic incompatibility of broke-off in b.) cannot merely be a function of two lexical parameters *broken and off* but of additional elements

in the construction. Here what complicates the issue is that *off's* semantic content is being modified by the surrounding elements and therefore the *off* that is incompatible with *broke* in b.) is a different *off* from the chocolate example. *Off* in "broke off a piece of chocolate" functions almost reflexively with chocolate while in b.) off means removal from the table. Thus you are not merely assessing the semantic compatibility of multiple static elements but the semantic compatibility of multiple dynamic elements.

Below I will be examining Caused Motion Constructions, showing why these words cannot function in the current generative semantic framework and discuss Adele's solution to the problem. We will look at "He sneezed the napkin off the table" by first looking at a structurally parallel sentence *"He kicked the ball off the table"*.

[[He]] : e

[[kicked]] : <e, <<e,t>,<e,t>>>

λx.[λf.[λz[z kicked x s.t f(x) = 1]

[[the ball]] : e

[[off]] : <<e,<e,t>> = **λx.[λy.[x is off y]**

[[the table]] : e

Mirroring the above derivation for *"He sneezed the napkin off the table",* we see that a denotation of type <e, <<e,t>,<e,t>>> doesn't work for sneeze. This is for two reasons. Since you cannot say "he sneezed the napkin" like you can say "he kicked the ball", kick has a transitive specification that is <e,<e,t>>. Sneeze, however, cannot be transitive without the added path terms "off the table".

Therefore **λx.[λf.[λz[z sneezed x s.t f(x) = 1]** does not work since

“z sneezed x” is nonsensical. “He kicked the ball s.t that ball is in the goal” works while “he sneezed the napkin s.t the napkin is in the goal” does not work because sneeze does not have a transitive entry. It jumps from intransitive to ditransitive while the verb kick does not have such an incontinuity. This is displayed in the examples below

- He kicked <e,t> ->He kicked <<e,<e,t>> the ball -> He kicked <e,<<e,t>,<e,t>>> the ball off the table
- He sneezed <e,t> -> *He sneezed <<e,<e,t>> the napkin -> He sneezed <e,<<e,t>,<e,t>>> the napkin off the table

Moreover, the f = *sneeze* $\in D_{<e,t>}$ in “sneezed the napkin off the table” is highly restricted while it is not in “kicked the ball off the table”. Sneeze can really only take off at its path preposition. Thus the rule of functional application (FA) cannot be used since the argument sneeze would need to take would not be general but specific.

Example:

f $\in D_{<e,t>}$ = *into the goal* -> kicked the ball *into the goal* / *sneezed the napkin into the ground.

f $\in D_{<e,t>}$ = *over the post* -> kicked the ball *over the post* / *sneezed the napkin over the book.

While the preposition off can be intersective alongside kick it cannot be intersective alongside sneeze. An example of a preposition that functions as an intersective is the sentence “Tina is a pianist **from** Rome” -> Tina is a pianist and Tina is from Rome.

The ball was kicked and the ball was off the table works while **The napkin was sneezed and the napkin was off the table* does not work since the "napkin was sneezed" cannot stand alone as a valid predicate. The rule of predicate modification (PM) is used lambda calculus to merge intersective items.

Predicate Modification (PM)
If α is a branching node, $\{\beta, \gamma\}$ is the set of α's daughters, and $[\![\beta]\!]$ and $[\![\gamma]\!]$ are both in $D_{\langle\sigma,t\rangle}$, then
$[\![\alpha]\!] = \lambda x_\sigma[[\![\beta]\!](x) = [\![\gamma]\!](x) = 1]$ (Or equivalently: $[\![\alpha]\!] = \lambda x_\sigma[[\![\beta]\!](x) \wedge [\![\gamma]\!](x)]$)

You could use P to derive "he kicked the ball in the goal" but this is not possible with the verb *sneeze*. Interestingly, the mathematical structure of predicate modification relies on the fact that predicates are semantic unitaries that can be separated and partitioned from the phrasal level: the set {β, γ} is such such a partition. So if "off" cannot function as an intersective, it must be able to function as a subsective. An example of a subsective preposition is "John ran ***with*** Jill" -> John ran. The reason "off" cannot be subsective is again because "he sneezed" cannot stand alone as a valid predicate. Since PM and FA will both not work in deriving this sentence, we need to either create a new compositional rule, or modify the designation of "sneezed". Goldberg chose to go ahead with the second option. She posited a construction called the Caused Motion Construction which states that sneezed has the following theta roles = <cause, goal, theme>. The lambda derivation for the following sentence would be: <sneezed, off the table, napkin>

λx.[λf.[λz[f(x) -> Z s.t. X∩Z]. All the above derivational problems could be solved if allowed sneezed to be underlying intransitive, but this is precisely what we wanted to avoid from the beginning. We don't want to posit additional idiosyncratic

senses to words if we don't need to. What if there was a way to reach an intransitive function by combining two words to make a new word. This would come at the benefits of retaining the current rules of lambda calculus at the cost of postulating covert lexical items. Taking [[off]] = **[λx.[λy[x is off y]]]** and [[sneezed]] = **[λz.[z sneezed]]** there covert combination at the lexical level would create the formula with least alteration to both respective lambda functions.

[[sneezed-off]] = **[λz.[λx.[λy[z sneezed x off y]]].** Now, we can derive "he sneezed the napkin off the table" buy using three instances of functional application. First, the table in plugged in for y. Then the napkin is plugged in for x.Finally, he is plugged in for z.

I do not disagree with the fact that all the examples below can be unified by the Caused Motion Construction. That is to say that all these verbs do not normally contain motion senses and only do so in regards to the larger construction.

Examples:

(1) They laughed him off stage.

(2) Frank sneezed the tissue off the table

(3) The kids swam the logs upstream

(4) Frank squeezed the ball through the crack

Caused Motion Construction: <Cause, Goal, Theme>

While it could be true to say that the above examples all fit into this template, it may be incorrect to say that this template lies at the heart of the generative process in producing the sentences to begin with. It is the case that not all verbs be used in this template, and the Caused Motion Construction gives no guidance as to how this discrimination should be made. Which verbs have the possibility of adding a motion sense to them? The problem of constraining infectious sentences generated by syntax is only partially solved with theta roles and so the Caused Motion Construction template, essentially a theta role designation, carries with it the same problem present in the rest of syntax. Constructions bring this problem to the forefront because of how highly limited and constrained some of them can be. Taking the example of "Frank sneezed the tissue off the table" and starting with the template <Cause: Sneezed, *Goal*, Theme: napkin>, we immediately run into a problem. There is no criteria that the construction wears on its sleeve that could help us decide which prepositions to use that would result in a felicitous construction when filling in the goal slot.

(i) *Frank sneezed the napkin *into* the ground

(ii) ?Frank sneezed the napkin onto the floor

(iii) *Frank sneezed the napkin *around* the room

(iv) *Frank sneezed the napkin *over* the desk

It is the interaction between the semantics of two words, in the above examples, that determines whether the construction is felicitous. Yet again the given template doesn't give any information about the nature of this interaction between, as with any template, it

is blind to the elements that fill its slots. My argument is that inside of constructions, when the total meaning cannot be derived from the sum of the component parts, is due to the fact that the underlying mental representation is combining elements to create a new, smaller set of component parts. Once these component parts have been identified, normal semantic derivations can carry on as before. What determines whether words can combine inside of a construction, or whether the construction is even valid, depends entirely on the semantic interaction of the relevant elements: the elements whose commonplace semantics can't be used to parse the sentence. If combination is necessary at a junction where this is not possible, then the construction, as in examples (ii-iv) becomes invalid: *"Sneezed-Around"* and *"Sneezed-Over"* cannot form valid combinations because the motion semantics caused by a sneeze cannot encompass around and over prepositions: this sort of motion is simply too complex.