Ling 5702: Lecture Notes 11
Compositional Semantics for Graphs

11.1 Generalized Categorial Grammar

The desiderata described in the previous lecture require referents for reified sets to have scope arguments in order to distinguish, for example, the set of all numbers, scoping at the top level, from the set of all numbers *for each line*, scoping inside a set of lines. An interesting consequence of these representational decisions is that each content lexical item is associated with not one but two reified set referents: one which serves as a restrictor, whose iterator serves as an argument to eventuality referents of modifiers of this lexical item; and one which serves as a scope body, whose iterator serves as an argument to eventuality referents of lexical items that take this lexical item as an argument. In this dependency representation, $\mathbf{s_3}$ is a restrictor, whose iterator is modified by *starting*, and $\mathbf{s'_3}$ is a scope body, whose iterator is an argument of *print*. Either of these reified sets can be co-referenced by pronouns in a following sentence, for example: *They might not have a number,* where *they* co-refers with the restrictor (lines in general), or: *They should have the number indented,* where *they* co-refers with the scope body (lines having a number).

We can use a Generalized Categorial Grammar (GCG) (Bach, 1981; Oehrle, 1994) to distinguish argument and modifier compositions (from which restrictor and scope body sets are derived in a tree-structured scope graph). A generalized categorial grammar is a tuple $\langle U, V, R, W, M\rangle$ of a set U of primitive category types, a set V of type-constructing operators, a set R of inference rules, a set W of vocabulary items, and a mapping M from vocabulary items to complex types. The set of primitive category types U specify various linguistic forms for descriptions of entities or eventualities, corresponding to different clause types, e.g. for English:

V: finite verbal (*she knew it*)
I: infinitive verbal (*her to know it*)
B: base-form verbal (*her know it*)
L: participial verbal (*her known it*)
A: adjectival/predicative (*him knowing it*)
R: adverbial (*him knowingly*)
G: gerund (*him knowing it*)
N: nominal (*his knowledge of it*)
D: determiner (*her knowledge of it's*)
O: genitive (*of her knowledge of it*)
C: complementized finite (*that he knew it*)
F: embedded infinitive (*for him to know it*)
E: embedded base-form (*that he know it*)
Q: subject-aux inverted (*did she know it*)
S: complete utterance (*know it*)

The set of type-constructing operators O specify various kinds of arguments, e.g.:

-a: initial argument (similar to CG '\')

-b: final argument (similar to CG '/')
-c: initial conjunct
-d: final conjunct
-g: gap argument (similar to HPSG 'SLASH')
-h: held argument for right node raising
-i: interrogative pronoun argument (HPSG 'QUE')
-r: relative pronoun argument (HPSG 'REL')

Using this set of primitive category types U and type-constructing operators O, a set of complex categories C can be defined such that:

1. every U is in C
2. every $C \times V \times C$ is in C
3. nothing else is in C

Like a Combinatory Categorial Grammar (Steedman, 2000), a GCG defines syntactic dependencies for compositions that are determined by the number and kind of unsatisfied dependencies of the composed category types. These are similar to dependencies for subject, direct object, preposition complement, etc., of Stanford dependencies (de Marneffe et al., 2006), but are reduced to numbers: '1' for subject, '2' for direct object or preposition complement or sentential complement (depending of the head category), '3' for indirect object, etc.

Semantic dependencies introduced in this paper are then associated with these syntactic dependencies, with the referent of a subject associated with the first argument of an eventuality, the referent of a direct object associated with the second argument, etc, for all verb forms other than passive verbs. In the case of passive verbs, the referent of a subject is associated with the second argument of an eventuality, the referent of a direct object associated with the third argument, and so on.

In order to model referents of verbs as eventualities which can be quantified over by adverbs like 'never,' 'once,' 'twice,' etc. (Parsons, 1990), and also in order to have a consistent treatment of argument and modifier attachment across all category types, eventualities associated with verbs are also quantified. Outgoing semantic dependencies to arguments of eventualities are then applied as constraints to the discourse referent variable of this quantifier's restrictor set. Incoming dependencies to modificands of modifiers are also applied as constraints to discourse referent variables of restrictor sets, but incoming dependencies to arguments of predicates are applied as constraints to discourse referent variables of nuclear scope sets. This assignment to restrictor or nuclear scope sets must be defined by parser operations, so associations between syntactic and semantic dependencies must be left partially undefined in lexical entries. Lexical entries are therefore defined here with separate syntactic and semantic dependencies, with syntactic dependencies from lexical items identified using even numbers, and semantic dependencies from lexical items identified using odd numbers. For example, a lexical mapping for the finite transitive verb *contains* might be:

$$\text{contains} \;\Rightarrow\; \textbf{V-aN-bN} : \lambda_i\,(\mathbf{f_0}\ i)=\text{contains} \wedge (\mathbf{f_0}\ (\mathbf{f_1}\ (\mathbf{f_1}\ (\mathbf{f_1}\ i))))=\textsc{Contain}$$
$$\wedge\ (\mathbf{f_1}\ (\mathbf{f_1}\ (\mathbf{f_1}\ (\mathbf{f_1}\ i))))=(\mathbf{f_1}\ (\mathbf{f_3}\ i)) \wedge (\mathbf{f_2}\ (\mathbf{f_1}\ (\mathbf{f_1}\ (\mathbf{f_1}\ i))))=(\mathbf{f_1}\ (\mathbf{f_5}\ i))$$

These dependency constraints are represented graphically in Figure 1a.

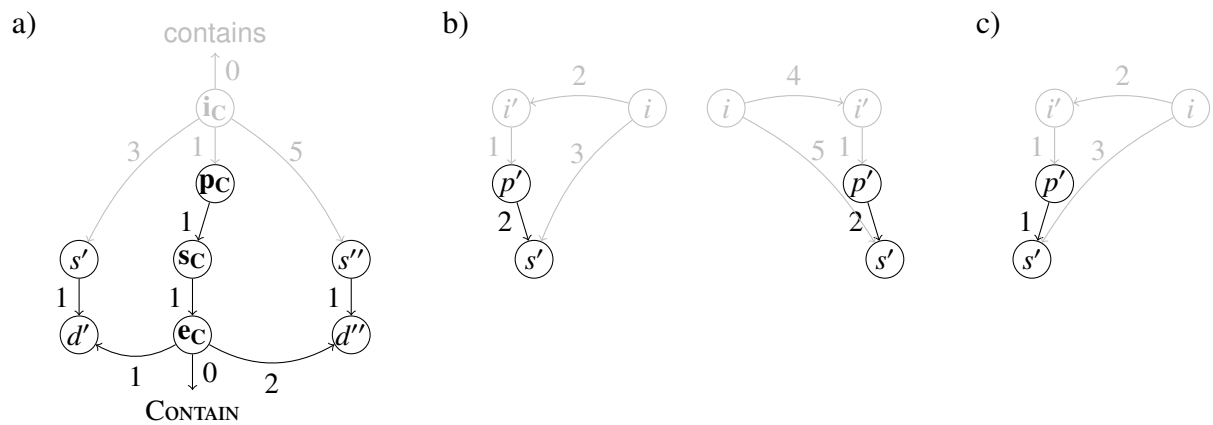

Figure 1: Example lexical semantic dependencies for the verb *contains* (a), and dependency equations for argument attachment (b) and modifier attachment (c) in GCG deduction rules.

11.2 Inference rules for argument attachment

As in other categorial grammars, inference rules for local argument attachment apply functors of category $c\text{-}\mathbf{a}d$ or $c\text{-}\mathbf{b}d$ to initial or final arguments of category d:

$$d : g \quad c\text{-}\mathbf{a}d : h \Rightarrow c : (\mathbf{f}_{c\text{-}\mathbf{a}d}\ g\ h) \tag{Aa}$$

$$c\text{-}\mathbf{b}d : g \quad d : h \Rightarrow c : (\mathbf{f}_{c\text{-}\mathbf{b}d}\ g\ h) \tag{Ab}$$

where $\mathbf{f}_{u\varphi_1\ldots\varphi_n}$ are composition functions for $u{\in}U$ and $\varphi{\in}\{\text{-}\mathbf{a}, \text{-}\mathbf{b}, \text{-}\mathbf{c}, \text{-}\mathbf{d}\}{\times}C$, which connect the lexical item $(\mathbf{f}_{2n}\ i)$ of an initial (left) child function g as the nth argument of lexical item i of a final (right) child function h, or vice versa:

$$\mathbf{f}_{u\varphi_{1..n-1}\text{-}\mathbf{a}d} \stackrel{\text{def}}{=} \lambda_{g\,h\,i}\,(g\,(\mathbf{f}_{2n}\ i)) \wedge (h\ i) \wedge (\mathbf{f}_{2n+1}\ i){=}(\mathbf{f_2}\,(\mathbf{f_1}, (\mathbf{f}_{2n}\ i))) \tag{1a}$$

$$\mathbf{f}_{u\varphi_{1..n-1}\text{-}\mathbf{b}d} \stackrel{\text{def}}{=} \lambda_{g\,h\,i}\,(g\ i) \wedge (h\,(\mathbf{f}_{2n}\ i)) \wedge (\mathbf{f}_{2n+1}\ i){=}(\mathbf{f_2}\,(\mathbf{f_1}\,(\mathbf{f}_{2n}\ i))) \tag{1b}$$

as shown in Figure 1b. This associates the lexical semantic argument of the predicate $(\mathbf{f}_{2n+1}\ i)$ with the scope body of the syntactic argument $(\mathbf{f_2}\,(\mathbf{f_1}\,(\mathbf{f}_{2n}\ i)))$. For example, the following inference attaches a subject to a verb:

$$\dfrac{\dfrac{\text{every line}}{\mathbf{N} : \lambda_i\,(\mathbf{f_0}\ i){=}\text{line} \ldots} \quad \dfrac{\text{contains two numbers}}{\mathbf{V\text{-}aN} : \lambda_i\,(\mathbf{f_0}\ i){=}\text{contains} \ldots}}{\mathbf{V} : \lambda_i\,(\mathbf{f_0}\,(\mathbf{f_2}\ i)){=}\text{line} \ldots \wedge (\mathbf{f_0}\ i){=}\text{contains} \ldots \wedge\ (\mathbf{f_3}\ i){=}(\mathbf{f_2}\,(\mathbf{f_1}\,(\mathbf{f_2}\ i)))}\ \text{Aa}$$

11.3 Inference rules for modifier attachment

Crucial to a graphical representation of scope, this grammar uses distinguished inference rules for modifier attachment, which also allows modifier categories to be consolidated with categories for modifiers in other contexts, and with certain predicative categories. Inference rules for modifier

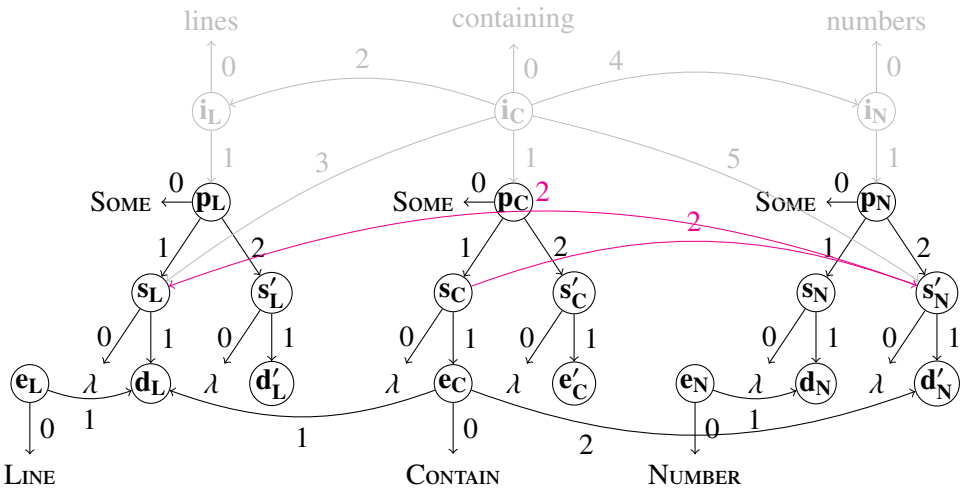

Figure 2: Compositional analysis of noun phrase *lines containing numbers* exemplifying both argument attachment (to *numbers*) and modifier attachment (to *lines*). Lexical dependencies are shown in gray, and continuation dependencies (which do not result from syntactic composition) are highlighted.

attachment apply initial or final modifiers of category $u\text{-}\mathbf{a}d$ to modificands of category c, for $u \in U$ and $c, d \in C$:

$$u\text{-}\mathbf{a}d : g \quad c : h \Rightarrow c : (\mathbf{f}_{\text{IM}}\, g\, h) \tag{Ma}$$

$$c : g \quad u\text{-}\mathbf{a}d : h \Rightarrow c : (\mathbf{f}_{\text{FM}}\, g\, h) \tag{Mb}$$

where $\mathbf{f}_{\text{IM}}$ and $\mathbf{f}_{\text{FM}}$ are category-independent composition functions for initial and final modifiers, which return the lexical item of the argument (j) rather than of the predicate (i):[1]

$$\mathbf{f}_{\text{IM}} \stackrel{\text{def}}{=} \lambda_{g\,h\,j}\, \exists_i\, (\mathbf{f_2}\ i){=}j \wedge (g\ i) \wedge (h\ j) \wedge (\mathbf{f_3}\ i){=}(\mathbf{f_1}\ (\mathbf{f_1}\ (\mathbf{f_2}\ i))) \tag{2a}$$

$$\mathbf{f}_{\text{FM}} \stackrel{\text{def}}{=} \lambda_{g\,h\,j}\, \exists_i\, (\mathbf{f_2}\ i){=}j \wedge (g\ j) \wedge (h\ i) \wedge (\mathbf{f_3}\ i){=}(\mathbf{f_1}\ (\mathbf{f_1}\ (\mathbf{f_2}\ i))) \tag{2b}$$

as shown in Figure 1c. This allows categories for predicates to be re-used as modifiers. Unlike argument attachment, modifier attachment associates the lexical semantic argument of the modifier $(\mathbf{f}_{2n+1}\ i)$ with the restrictor of the modificand $(\mathbf{f_1}\ (\mathbf{f_1}\ (\mathbf{f}_{2n}\ i)))$. For example, the following inference attaches an adjectival modifier to a noun phrase:

$$\dfrac{\dfrac{\text{every line}}{\mathbf{N} : \lambda_i\, (\mathbf{f_0}\ i){=}\text{line} \ldots} \quad \dfrac{\text{containing two numbers}}{\mathbf{A\text{-}aN} : \lambda_i\, (\mathbf{f_0}\ i){=}\text{containing} \ldots}}{\mathbf{N} : \lambda_i\, (\mathbf{f_0}\ i){=}\text{line} \ldots \wedge \exists_j\, (\mathbf{f_0}\ j){=}\text{containing} \ldots \wedge\ (\mathbf{f_2}\ j){=}i \wedge (\mathbf{f_3}\ j){=}(\mathbf{f_1}\ (\mathbf{f_1}\ (\mathbf{f_2}\ j)))}\ \text{Mb}$$

[1]This reversal of direction for modifier dependencies is similar to that described in dependency accounts of Tree Adjoining Grammars (Joshi, 1985; Candito and Kahane, 1998).

11.4 Inference rules for non-local dependencies

The categorial grammar adopted in this paper also uses distinguished inference rules to introduce, propagate, and bind missing non-local arguments k, similar to the gap or slash rules of Generalized Phrase Structure Grammar (Gazdar et al., 1985) and Head-driven Phrase Structure Grammar (Pollard and Sag, 1994). Inference rules for gap attachment hypothesize gaps as initial arguments, final arguments, or modifiers, for $c, d \in C$:[2]

$$c\text{-}\mathbf{a}d : g \Rightarrow c\text{-}\mathbf{g}d : \lambda_k(\mathbf{f}_{c\text{-}\mathbf{g}d}\ \{k\}\ g) \tag{Ga}$$

$$c\text{-}\mathbf{b}d : g \Rightarrow c\text{-}\mathbf{g}d : \lambda_k(\mathbf{f}_{c\text{-}\mathbf{g}d}\ \{k\}\ g) \tag{Gb}$$

$$c : g \Rightarrow c\text{-}\mathbf{g}d : \lambda_k(\mathbf{f}_{\mathrm{IM}}\ \{k\}\ g) \tag{Gc}$$

where $\mathbf{f}_{u\varphi_1\ldots\varphi_{n-1}\text{-}\mathbf{g}d}$ is a composition function for $u{\in}U$ and $\varphi{\in}\{\text{-}\mathbf{a}, \text{-}\mathbf{b}, \text{-}\mathbf{c}, \text{-}\mathbf{d}\}{\times}C$, which connects the lexical item $(\mathbf{f}_{2n}\ i)$ of an initial (left) child function g as the nth argument of lexical item i of a child function h:

$$\mathbf{f}_{u\varphi_{1..n-1}\text{-}\mathbf{g}d} \stackrel{\text{def}}{=} \lambda_{g\,h\,i}\,(g\,(\mathbf{f}_{2n}\ i)) \wedge (h\ i) \wedge (\mathbf{f}_{2n+1}\ i){=}(\mathbf{f_1}\,(\mathbf{f_1}, (\mathbf{f}_{2n}\ i))) \tag{4}$$

Like modifier attachment, non-local dependency creation associates the lexical semantic argument of the predicate $(\mathbf{f}_{2n+1}\ i)$ with the restrictor of the eventual filler $(\mathbf{f_1}\,(\mathbf{f_1}\,(\mathbf{f}_{2n}\ i)))$. Non-local arguments k, using non-local operator and argument category $\psi{\in}\{\text{-}\mathbf{g}, \text{-}\mathbf{h}, \text{-}\mathbf{i}, \text{-}\mathbf{r}\}{\times}C$, are then propagated to the consequent from all possible combinations of antecedents, skipping over the composition function. For each rule $d{:}g\ \ e{:}h \Rightarrow c{:}(f\ g\ h) \in \{\text{Aa–b,Ma–b}\}$:

$$d{:}g\ \ e\psi{:}h \Rightarrow c\psi{:}\lambda_k(f\ g\ (h\ k)) \tag{Ac–d,Mc–d}$$

$$d\psi{:}g\ \ e{:}h \Rightarrow c\psi{:}\lambda_k(f\ (g\ k)\ h) \tag{Ae–f,Me–f}$$

$$d\psi{:}g\ \ e\psi{:}h \Rightarrow c\psi{:}\lambda_k(f\ (g\ k)\ (h\ k)) \tag{Ag–h,Mg–h}$$

For example, the null relative clause in the sentence *Print every number the line contains,* can be derived using the following inferences:

$$\dfrac{\dfrac{\text{the line}}{\mathbf{N} : \lambda_i\,(\mathbf{f_0}\ i){=}\mathsf{line}} \quad \dfrac{\dfrac{\text{contains}}{\mathbf{V\text{-}aN\text{-}bN} : \lambda_i\,(\mathbf{f_0}\ i){=}\mathsf{contains}}}{\mathbf{V\text{-}aN\text{-}gN} : \lambda_{k\,i}\,(\mathbf{f_0}\ i){=}\mathsf{contains} \wedge (\mathbf{f_4}\ i){=}k \wedge (\mathbf{f_5}\ i){=}(\mathbf{f_1}\,(\mathbf{f_1}\,(\mathbf{f_4}\ i)))}\ \text{Gb}}{\begin{array}{c}\mathbf{V\text{-}gN} : \lambda_{k\,i}\,(\mathbf{f_0}\ i){=}\mathsf{contains} \wedge (\mathbf{f_0}\,(\mathbf{f_2}\ i)){=}\mathsf{line} \wedge (\mathbf{f_3}\ i){=}(\mathbf{f_2}\,(\mathbf{f_1}\,(\mathbf{f_2}\ i))) \\ \wedge (\mathbf{f_4}\ i){=}k \wedge (\mathbf{f_5}\ i){=}(\mathbf{f_1}\,(\mathbf{f_1}\,(\mathbf{f_4}\ i)))\end{array}}\ \text{Ac}$$

11.5 Inference rules for interrogatives and relative clauses

This grammar uses distinguished inference rules for relative and interrogative pronouns which introduce a clause with a gap dependency, for $c, d, e \in C$, $\psi{\in}\{\text{-}\mathbf{g}\}{\times}C$:

$$d\text{-}\mathbf{i}e{:}g\ c\text{-}\mathbf{g}d{:}h \Rightarrow c\text{-}\mathbf{i}e{:}\lambda_{k\,j}\,\exists_i\,(g\,k\,i) \wedge (h\,i\,j) \tag{Fa}$$

$$d\text{-}\mathbf{r}e{:}g\ c\text{-}\mathbf{g}d{:}h \Rightarrow c\text{-}\mathbf{r}e{:}\lambda_{k\,j}\,\exists_i\,(g\,k\,i) \wedge (h\,i\,j) \tag{Fb}$$

[2]Here, set notation is used in order to save space: $\{k\} = (\lambda_i\ i{=}k)$.

For example:

$$\dfrac{\dfrac{\text{every number}}{\mathbf{N}:\lambda_i(\mathbf{f_0}\ i)\text{=number}} \qquad \dfrac{\dfrac{\text{which}}{\mathbf{N\text{-}rN}:\lambda_{k\,i}(\mathbf{f_0}\ i)\text{=which}\ \wedge\ (\mathbf{f_2}\ i)\text{=}k} \qquad \dfrac{\text{the line contains}}{\mathbf{V\text{-}gN}:\lambda_{k\,i}(\mathbf{f_0}\ i)\text{=contains}\ \wedge\ (\mathbf{f_4}\ i)\text{=}k\ \wedge\ \ldots}}{\mathbf{V\text{-}rN}:\lambda_{k\,i}(\mathbf{f_0}\ i)\text{=contains}\ \wedge\ \exists_j(\mathbf{f_4}\ i)\text{=}j\ \wedge\ (\mathbf{f_0}\ j)\text{=which}\ \wedge\ (\mathbf{f_2}\ j)\text{=}k}\ \text{Fb}}{\mathbf{N}:\lambda_i(\mathbf{f_0}\ i)\text{=line}\ \wedge\ \exists_j(\mathbf{f_0}\ j)\text{=contains}\ \wedge\ \exists_k(\mathbf{f_4}\ j)\text{=}k\ \wedge\ (\mathbf{f_0}\ k)\text{=which}\ \wedge\ (\mathbf{f_2}\ k)\text{=}i}\ \text{R}$$

Also, inference rules for relative pronoun attachment apply pronominal relative clauses of category $c\text{-}\mathbf{r}d$ to modificands of category e:

$$e:g\ \ c\text{-}\mathbf{r}d:h \Rightarrow e:\lambda_i\,\exists_j\,(g\ i)\wedge(h\ i\ j) \tag{R}$$

11.6 Other inference rules

The grammar adopted in this paper also uses inference rules for category change:

$$c\psi{:}g \Rightarrow d\chi{:}g \tag{T}$$

inference rules for argument elision:

$$c\text{-}\mathbf{a}d:g \Rightarrow c:g \qquad c\text{-}\mathbf{b}d:g \Rightarrow c:g \tag{Ea–b}$$

$$c\text{-}\mathbf{a}d\psi:g \Rightarrow c\psi:g \qquad c\text{-}\mathbf{b}d\psi:g \Rightarrow c\psi:g \tag{Ec–d}$$

inference rules for right-node raising:

$$c\text{-}\mathbf{b}d:g \Rightarrow\ c\text{-}\mathbf{h}d:\lambda_k\,(\mathbf{f}_{c\text{-}\mathbf{b}d}\ g\ \{k\}) \tag{Ha}$$

$$c\text{-}\mathbf{h}d:g\ \ d:h \Rightarrow\ c:\lambda_i\,\exists_j\,(g\ j\ i)\wedge(h\ j) \tag{Hb}$$

and inference rules for coordinating conjunctions:

$$d{:}g\ \ c\text{-}\mathbf{c}d{:}h \Rightarrow c{:}(\mathbf{f}_{c\text{-}\mathbf{a}d}\ g\ h) \tag{Ca}$$

$$c\text{-}\mathbf{d}d{:}g\ \ d{:}h \Rightarrow c{:}(\mathbf{f}_{c\text{-}\mathbf{b}d}\ g\ h) \tag{Cb}$$

$$d{:}g\ \ c\text{-}\mathbf{c}d{:}h \Rightarrow c\text{-}\mathbf{c}d{:}(\mathbf{f}_{c\text{-}\mathbf{c}d}\ g\ h) \tag{Cc}$$

where $\mathbf{f}_{u\varphi_1\ldots\varphi_{n-1}\psi\text{-}\mathbf{c}c}$ for $u{\in}U$, $\varphi{\in}\{\text{-}\mathbf{a},\text{-}\mathbf{b}\}{\times}C$, $\psi{\in}\{\epsilon\}{\cup}(\{\text{-}\mathbf{g},\text{-}\mathbf{h},\text{-}\mathbf{i},\text{-}\mathbf{r}\}{\times}C)$, and $c\in C$ is a special composition function for elided conjunctions in lists of three or more conjuncts:

$$\mathbf{f}_{u\varphi_1\ldots\varphi_{n-1}\psi\text{-}\mathbf{c}c} \stackrel{\text{def}}{=} \lambda_{g\,h\,i}\,(\mathbf{f_0}\ i){=}(\mathbf{f_0}\ (\mathbf{f}_{2n+2}\ i))\wedge(g\ (\mathbf{f}_{2n}\ (\mathbf{f}_{2n+2}\ i)))\wedge(h\ (\mathbf{f}_{2n+2}\ i)) \tag{5}$$

The grammar also supports distinguished inference rules for coordinating gap, relative pronoun, and other non-local dependencies:

$$d\psi{:}g\ \ c\text{-}\mathbf{c}(d\psi){:}h \Rightarrow c{:}\lambda_k(\mathbf{f}_{c\text{-}\mathbf{a}d}\ (g\ k)\ (h\ k)) \tag{Cd}$$

$$c\text{-}\mathbf{d}(d\psi){:}g\ \ d\psi{:}h \Rightarrow c{:}\lambda_k(\mathbf{f}_{c\text{-}\mathbf{b}d}\ g\ (h\ k)) \tag{Ce}$$

$$d\psi{:}g\ \ c\text{-}\mathbf{c}(d\psi){:}h \Rightarrow c\text{-}\mathbf{c}(d\psi){:}\lambda_k(\mathbf{f}_{c\text{-}\mathbf{c}d}\ (g\ k)\ (h\ k)) \tag{Cf}$$

References

Bach, E. (1981). Discontinuous constituents in generalized categorial grammars. *Proceedings of the Annual Meeting of the Northeast Linguistic Society (NELS)*, 11:1–12.

Candito, M.-H. and Kahane, S. (1998). Can the TAG derivation tree represent a semantic graph? In *Proceedings of the TAG+4 Workshop*, University of Pennsylvania.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC 2006*.

Gazdar, G., Klein, E., Pullum, G., and Sag, I. (1985). *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, MA.

Joshi, A. K. (1985). How much context sensitivity is necessary for characterizing structural descriptions: Tree adjoining grammars. In D. Dowty, L. K. and Zwicky, A., editors, *Natural language parsing: Psychological, computational and theoretical perspectives*, pages 206–250. Cambridge University Press, Cambridge, U.K.

Nguyen, L., van Schijndel, M., and Schuler, W. (2012). Accurate unbounded dependency recovery using generalized categorial grammars. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING '12)*, pages 2125–2140, Mumbai, India.

Oehrle, R. T. (1994). Term-labeled categorial type systems. *Linguistics and Philosophy*, 17(6):633–678.

Parsons, T. (1990). *Events in the Semantics of English*. MIT Press.

Pollard, C. and Sag, I. (1994). *Head-driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Steedman, M. (2000). *The syntactic process*. MIT Press/Bradford Books, Cambridge, MA.