Compositional Semantics for GCG/CCG

Copyright, Ted Briscoe (`ejb@cl.cam.ac.uk`), GS18, Computer Lab

1 Generalized Categorial Grammars

1.1 Categorial Grammar

Classic (AB) categorial grammar consists of atomic categories of the form: N, NP, S, etc., and functor categories of the form S/N, (S\ NP)/NP, etc. constructed by combining atomic categories with slash and backslash with the functor leftmost and the 'outer' argument rightmost (see Wood, 1993 for a textbook introduction to CG and its generalisations).

Functors and arguments are combined by directional rules of (function-argument) application as in Figure 2 below. CGs of this form are weakly equivalent to CFGs but assign a fully binary branching structure. So ditransitive verb complements, whose categories will be ((S\ NP)/PP)/NP, will be assigned a different structure than in a standard CF PSG approach. Figure 1 shows the CG derivation for a simple example. One feature of CG

```
Kim                loves                  Sandy
NP                 (S\NP)/NP              NP
kim'               λ y,x [love′(x y)]     sandy′
                   ---------------------- FA
                   S\NP
                   λ x [love′(x sandy′)]
------------------------------ BA
S
love′(kim′ sandy′)
```

Figure 1: CG Derivation for *Kim loves Sandy*

is that syntax and semantics are more closely associated than in a standard 'rule-to-rule' framework as function application in the syntax directly corresponds to function application (beta reduction) in the lambda calculus (regardless of directionality). This framework is 'radically lexical' since now there are just two rules of syntactic combination (FA,BA) and one rule of semantic application. Everything else must be captured in terms of the lexical categories. For example, modifiers cannot be dealt with in terms of separate rules and instead must be characterised lexically as functor arguments which yield categories of the same type (X/X, X\X) e.g. N/N or (S\ NP)/(S\ NP) – can you see what classes of word these categories would be appropriate for?

1.2 Generalized Categorial Grammar

The main interest in exploring CGs is that various extensions of classic AB CG (with just function application) have been proposed in recent years. These deal well with phenomena like non-constituent coordination and mostly extend the generative capacity of the grammar to 'mild context-sensitivity' / indexed languages. The specific extension I will outline adds rules of composition, permutation and type raising to AB CG as in Figure 2. These license derivations involving non-standard constituency such as Figure 3.

Neither function composition nor permutation change the semantics associated with a given sentence, rather they introduce 'spurious' ambiguity in that they allow the same semantics to be recovered in different ways. This can be exploited to deal with non-constituent coordination (Figure 5), unbounded dependencies (Figure 4), and the relationship between intonation, focus and semantics. (See Wood, 1993 or Steedman, 2000 for fuller treatments of closely related approaches.) I use the term 'generalized' not to denote a specific theory, but as a loose cover term for extensions of CG, as does Wood.)

There are polynomial parsing algorithms (n^6) for some types of generalized CGs of this form (so long as rules such as type raising are constrained to apply finitely. Because of the 'spurious' ambiguity of GCGs some effort has been devoted to defining parsing algorithms which only find a single derivation in the equivalence class of derivations defining the same logical form. Steedman (2000) argues instead that the ambiguity is not spurious at all but rather correlates with different prosodies conveying different information structure (give-new, theme-rheme, focus – see Discourse Processing course).

Forward Application:	
$X/Y \ Y \Rightarrow X$	λ y [X(y)] (y) $\Rightarrow$ X(y)
Backward Application:	
$Y \ X\backslash Y \Rightarrow X$	λ y [X(y)] (y) $\Rightarrow$ X(y)
Forward Composition:	
$X/Y \ Y/Z \Rightarrow X/Z$	λ y [X(y)] λ z [Y(z)] $\Rightarrow$ λ z [X(Y(z))]
Backward Composition:	
$Y\backslash Z \ X\backslash Y \Rightarrow X\backslash Z$	λ z [Y(z)] λ y [X(y)] $\Rightarrow$ λ z [X(Y(z))]
(Generalized Weak) Permutation:	
$(X\|Y_1)\ldots\|Y_n \Rightarrow (X\|Y_n)\|Y_1\ldots$	λ $y_n\ldots,y_1$ [X($y_1\ldots,y_n$)] $\Rightarrow$ λ $y_1,y_n\ldots$ [X($y_1\ldots,y_n$)]
Type Raising:	
$(X \Rightarrow T/(T\backslash X)$	a $\Rightarrow$ λ T [T a]
$(X \Rightarrow T\backslash(T/X)$	a $\Rightarrow$ λ T [T a]

Figure 2: GCG Rule Schemata

$$\begin{array}{lll}
\text{Kim} & \text{loves} & \text{Sandy} \\
\text{NP} & \text{(S}\backslash\text{NP)/NP} & \text{NP} \\
\text{kim'} & \lambda\ \text{y,x [love}'\text{(x y)]} & \text{sandy}' \\
 & \text{- - - - - - - - - - P} & \\
 & \text{(S/NP)}\backslash\text{NP} & \\
 & \lambda\ \text{x,y [love}'\text{(x y)]} & \\
\multicolumn{2}{l}{\text{- BA}} & \\
\text{S/NP} & & \\
\multicolumn{2}{l}{\lambda\ \text{y [love}'\text{(kim}'\text{ y)]}} & \\
\multicolumn{3}{l}{\text{- FA}} \\
\text{S} & & \\
\multicolumn{2}{l}{\text{love}'\text{(kim}'\text{ sandy}'\text{)}} &
\end{array}$$

Figure 3: GCG Derivation for *Kim loves Sandy*

```
who         Kim    thinks                    Sandy               loves
S/(S/NP)    NP     (S\NP)/S                  NP                  (S\NP)/NP
who′        kim′   λ P,x [think′(x,P)]                           λ y,x [love′(x y)]
                   -------------- P
                   (S/S)\NP
                   λ x,P [think′(x,P)]
            -------------------- BA
            S/S
            λ P [think′(kim′,P)]
                                                                 ----------- P
                                                                 (S/NP)\NP
                                                                 λ x,y [love′(x y)]
                                         -------------------------------- BA
                                         S/NP
                                         λ y [love′(kim′, y)]
            ------------------------------------- FC
            S/NP
            λ y [think′(kim′,love′(sandy′, y))]
---------------------------------- FA
S
think′(kim′,love′(sandy′, who′))]
```

Figure 4: GCG Derivation for *who Kim thinks Sandy loves*

```
Kim gave            Sandy               a book                  and                 Lesley a pen
(S/NP)/NP           NP                  NP                      (X\X)/X
λy,z [g′(k′,y,z)]   sandy′              a-bk′                   λx,y [&′(x,y)]      lesley′ a-pen′
                    ---- T              ---- T                                      ---- T ---- T
                    T\(T/NP)            T\(T/NP)                                    (T\T)/NP (T\T)/
                    λ P [P sandy′]                              . . .
                    ---------------------- BC                   . . .
                    T\((T/NP)/NP)                               . . .
                    λP [P(sandy′, a-bk′)]                       . . .
                    ---------------------------------------------------- conj
                    T\((T/NP)/NP)
                    λ P [and′(P(sandy′,a-bk′), P(lesley′,a-pen′))]
------------------------------------------------ BC
S
and′(give′(sandy′,a-bk′), give′(lesley′,a-pen′))
```

Figure 5: GCG Derivation for *Kim gave Sandy a book and Lesley a pen*

```
Kim is         a conservative    and                      proud of it
S/(NP ∨ AP)    NP                (X\X)/X                  AP
               ----------OI                               -------OI
               (NP ∨ AP)                                  (NP ∨ AP)
                                 ----------------------------- FA
                                 (NP ∨ AP)\(NP ∨ AP)
               -------------------------------- BA
               (NP ∨ AP)
--------------------- FA
S
```

Figure 6: GCG Derivation for *Kim is a conservative and proud of it*

Bayer (1996) draws on another tradition in GCG research which emphasises the connection between CG and substructural or resource logics (see e.g. Carpenter, 1997; Morrill, 1994). This tradition has concentrated on demonstrating that the rules of GCGs can be shown to implement sound deductive systems, rather than on implementation via unification operations. Thus the slash operator is a form of (linear) implication: from X/Y, infer an X given a Y.

From this perspective, it makes sense to introduce rules (of inference) like $\wedge$-elimination (AE): given X $\wedge$ Y, infer Y, and $\vee$-introduction (OI): given Y, infer X $\vee$ Y. Bayer defines his GCG category set as the closure of the atomic category under the operators: / \, $\wedge$, and $\vee$. He assigns *and* the usual polymorphic category (X\X)/X and *be* the category (S\NP)/(NP $\vee$ AP). This along with the rule of OI is enough to licence coordinations of unlike categories when the verb allows different complement types, as in Fig 6 This approach can be generalised to featural mismatches 'within' the same cateogory simply by allowing disjunctive feature values and aplying OI and AE to these values (e.g. CASE: **acc** $\vee$ **dat**) (feature neutralisation in German).

Although the use of unrestricted disjunction operators with complex unification-based feature systems is known to lead to computational inefficiency, if not intractability, it is not clear that this move would be so problematic in the context of the 'logicist' approach to GCG, as the features would be restricted to finite-valued morphosyntactic ones.

2 CCG / GCG References

Bayer, S. ‘The coordination of unlike categories’, *Language* 72.3, 579–616, 1996.
Carpenter, R. *Type-Logical Semantics*, MIT Press, 1997.
Morrill, G. *Type-logical Grammar*, Kluwer, 1994.
Steedman, M. *Sutface Structure and Interpretation*, MIT Press, 1996.
Steedman, M. *The Syntactic Process*, MIT Press, 2000.
Wood, M. *Categorial Grammar*, Routledge, 1993

3 (Neo-)Davidsonian Semantics

We’ve seen that some quite simple constructions (e.g. adverbs) create problems for FOL representations of compositional semantics. The obvious interpretation of an adverb is that it modifies a verbal predicate or proposition, but this isn’t possible in FOL. We’ve extended FOL with LC but so far only used LC as a means to compositionally construct FOL semantics for sentences in a syntax-guided fashion. The

(1) a Kim kissed Sandy passionately
    b passionate1(kiss1(kim1, sandy1))
    c $\exists$ e kiss1(e, kim1, sandy1) $\wedge$ passionate1(e)

(2) a Possibly Kim kissed Sandy
    b possible1(kiss1(kim1, sandy1))
    c $\exists$ e kiss1(e, kim1, sandy1)) $\wedge$ passionate1(e)

Davidson was the first to suggest that we could replace the b) semantics with the c) semantics by reifying events, i.e. including event individuals/entities in the corresponding model. We will write them, e, $e1$, etc to indicate that events are of a different sort to other entities. States like *Kim weighs too much* are also reified so some prefer to talk about ‘eventualities’ rather than events. Actually, this move doesn’t quite work for *possibly* because there is a difference in meaning between b) and c) below – can you see it?

(3) a Possibly every unicorn is white
 b possible1($\forall$ x unicorn1(x) $\rightarrow$ white1(x))
 c ($\forall$ x unicorn1(x) $\rightarrow$ possible1(white1(x)))

(The problem is very similar to that discussed for propositional attitude verbs in the semantics handout for L100.)

Parsons took this a stage further by proposing that arguments to predicates become binary relations between event variables and entities:

(4) a Kim kissed Sandy passionately
 b $\exists$ e kiss1(e) $\wedge$ agent(e,kim1) $\wedge$ patient(e,sandy1) $\wedge$ passionate1(e)
 c $\exists$ e kiss1(e) $\wedge$ arg1(e,kim1) $\wedge$ arg2(e,sandy1) $\wedge$ passionate1(e)

The problem with relations like 'agent' and 'patient' is determining exactly what they entail which is constant across all verbs, so we generally prefer to use more semantically-neutral realtions, as in c). The advantage of this neo-Davidsonian, Parsons-style representation is that it makes it easy to handle argument optionality. For example, the nominalisation of (4i)s:

(5) a Kim's / The kissing of Sandy (was passionate).
 b $\exists$ e kiss1(e) $\wedge$ arg1(e,kim1) $\wedge$ arg2(e,sandy1) $\wedge$ passionate1(e)
 c $\exists$ e kiss1(e) $\wedge$ arg2(e,sandy1) $\wedge$ passionate1(e)

and we don't have to specify the agent, so c) is a reasonable semantics for this case. Some otehr advantages of this representation are that we can handle tense naturally, and PP adjectival and adverbial modifiers looks more similar:

(6) a $\exists$ e kiss1(e) $\wedge$ arg1(e,kim1) $\wedge$ arg2(e,sandy1) $\wedge$ passionate1(e) $\wedge$ past(e)
 b $\exists$ e,x kiss1(e) $\wedge$ arg2(e,sandy1) $\wedge$ passionate1(e) $\wedge$ past(e) $\wedge$ in1(e,x) $\wedge$ bar(x)
 c $\exists$ e,x kiss1(e) $\wedge$ arg1(e,kim1) $\wedge$ arg2(e,y) $\wedge$ passionate1(e) $\wedge$ past(e) $\wedge$ in1(y,x) $\wedge$ bar(x) $\wedge$ person1(x)

Exercises: Can you provide English sentences that match the semantics of these examples? Can you work out how to build these represenations compositionally for sentences using LC?

4 Further References

Sections 1-2.3 from Ann Copestake ‘Robust Minimal Recursion Semantics’ and if you like some of the references therein:
http://www.cl.cam.ac.uk/users/aac10/papers/rmrsdraft.pdf