1 First steps to a fragment of English

Every expression in our language (whether lexical item or phrase) is modeled as a sign—a triple

\[(\text{phonological form, syntactic category, semantic interpretation}).\]

To make life easier, we’ll use orthography instead of phonological form. We’ll start off with a bunch of syncategorematic rules involving the basic connectives. Later we’ll improve the system by converting these into proper lexical items. Here’s conjunction (J:34)):

**Conjunction rule (temporary):** If \(\alpha\) is an expression of the form \(\langle [\alpha], S, [\alpha] \rangle\) and \(\beta\) is an expression of the form \(\langle [\beta], S, [\beta] \rangle\), then there is an expression \(\gamma\) of the form

\(\langle [\alpha\text{-and-}\beta], S, \text{for any } w : [\gamma](w) = 1 \text{ iff } [\alpha](w) = [\beta](w) = 1 \rangle.\)

Notice that we specified the phonological, syntactic, and semantic effects of this rule at the same time. This is characteristic of the directly compositional style. Notice also that there is lots of English in our definitions. This is fine, as long as we are careful and precise. Later, when it gets too hard to be sufficiently careful in English, we’ll introduce more formalism.

**Disjunction rule (temporary):** If \(\alpha\) is an expression of the form \(\langle [\alpha], S, [\alpha] \rangle\) and \(\beta\) is an expression of the form \(\langle [\beta], S, [\beta] \rangle\), then there is an expression \(\gamma\) of the form

\(\langle [\alpha\text{-or-}\beta], S, \text{for any } w : [\gamma](w) = 1 \text{ iff } [\alpha](w) = 1 \text{ or } [\beta](w) = 1 \rangle.\)

An awkward kind of negation (better version to come soon):

**Negation rule (temporary):** If \(\alpha\) is an expression of the form \(\langle [\alpha], S, [\alpha] \rangle\), then there is an expression \(\beta\) of the form

\(\langle [\text{it-is-not-the-case-that-}\alpha], S, \text{for any } w : [\beta](w) = 1 \text{ iff } [\alpha](w) = 0 \rangle.\)

With these three rules and at least one sentence to work with (i.e., expression with syntactic category \(S\)), we have an infinite language with which we can do propositional logic. Assume that our language contains the expressions
Then we can use these rules to build up

$$\langle \text{[it's-raining-and-it's-Tuesday]}, S, \{ \begin{align*} w_1 &\rightarrow 1 \\ w_2 &\rightarrow 0 \\ w_3 &\rightarrow 0 \\ w_4 &\rightarrow 0 \end{align*} \rangle \rangle$$

$$\langle \text{[it's-raining-or-it's-Tuesday]}, S, \{ \begin{align*} w_1 &\rightarrow 1 \\ w_2 &\rightarrow 1 \\ w_3 &\rightarrow 1 \\ w_4 &\rightarrow 0 \end{align*} \rangle \rangle$$

$$\langle \text{[it's-not-the-case-that-it's-raining]}, S, \{ \begin{align*} w_1 &\rightarrow 0 \\ w_2 &\rightarrow 0 \\ w_3 &\rightarrow 1 \\ w_4 &\rightarrow 1 \end{align*} \rangle \rangle$$

$$\langle \text{[it's-not-the-case-that-it's-not-the-case-that-it's-raining]}, S, \{ \begin{align*} w_1 &\rightarrow 0 \\ w_2 &\rightarrow 0 \\ w_3 &\rightarrow 1 \\ w_4 &\rightarrow 1 \end{align*} \rangle \rangle$$

where the last two are our first example of a syntactic ambiguity. Note, however, that in this system there is no explicit representation of the syntactic structure: the ambiguity plays out only in
• The potential for different semantic interpretations;
• The fact that we employed the rules of the language in different ways in order to derive these equivalent surface forms.

What are the different steps by which we proved the well-formedness of it’s-not-the-case-that-it’s-raining-or-it’s-Tuesday in each case? Which one is equivalent to a material conditional?

2 Noun phrases

Various things are of category $NP$, notably definition descriptions (the boy) and names. We’ll assume names are of category $NP$, and are rigid designators in Kripke’s (1980) sense. This means that they are constant functions from worlds to individuals:

\[
\begin{align*}
([Barack Obama], NP, \begin{cases} w_1 \to BO \\
w_2 \to BO \\
w_3 \to BO \\
w_4 \to BO\end{cases}) & \quad ([Hulk Hogan], NP, \begin{cases} w_1 \to HH \\
w_2 \to HH \\
w_3 \to HH \\
w_4 \to HH\end{cases})
\end{align*}
\]

This isn’t a deep formal commitment, but it is empirically motivated (Gödel & Schmidt, etc.). It is, of course, tricky to say precisely what it means for an individual to exist at multiple worlds: see e.g. Lewis 1968. For now, we’ll have to take for granted that this makes sense.

3 Verb phrases

For now let’s say verb phrases have category $VP$. This includes both intransitive verbs like laughed and transitive verbs like hugged the kitten that already have their objects. (J. gives a unary branching rule to get here from an intransitive verb, but let’s ignore this subtlety since it won’t be needed in CG.)

We’ll also assume that verb phrases denote the intensions of sets (or the characteristic functions thereof). That is, at any world (and time), laughed denotes the set of individuals who laughed at that world (and time).

Here’s a rule for dealing with NP-VP combinations, thinking of these and sentences as set-denoting:

If $\alpha = ([[\alpha]], NP, [[\alpha]])$ and $\beta = ([[\beta]], VP, [[\beta]])$ then there is a $\gamma$ s.t. $\gamma = ([[\alpha-\beta]], S, \{w | [[\alpha]](w) \in [[\beta]](w)\})$.

For instance, if $\alpha$ is Barack Obama and $\beta$ is laughed, then $\gamma$ has the following properties:

• “Phonological” form: Barack Obama laughed
• Syntactic category: $S$
We could just as well write this in the following equivalent form, using functions for sets:

If $\alpha = ([\alpha], NP, [\alpha])$ and $\beta = ([\beta], VP, [\beta])$ then there is a $\gamma$ s.t. $\gamma = ([\alpha-\beta], S, \text{for all } w, [\gamma](w) = 1 \iff [\beta](w)([\alpha](w)) = 1)$.

For instance, if $\alpha$ is Barack Obama and $\beta$ is laughed, then $\gamma$ has the following properties:

- “Phonological” form: Barack Obama laughed
- Syntactic category: $S$
- Semantics: the $f$ s.t., for all $w$, $f(w) = 1 \iff [\text{laughed}](w)([\text{Barack Obama}](w)) = 1$.

We’ll mostly use the function talk officially, but set-talk is common too and sometimes more intuitive. You should get in the habit of translating between them.

4 Semantic types

Basic types: $Types$ includes

- $e$: the type of individuals, like BO and HH.
- $s$: the type of worlds. $w_3$ and $w_7$ are of type $s$.
- $t$: the type of truth-values. 0 and 1 are of type $t$, and (on our assumptions) nothing else.
- $\tau$: the type of times.

Functional types:

For any types $\alpha$ and $\beta$, there is a type $\langle\alpha, \beta\rangle$. This is the type of functions whose domain is the set of objects of type $\alpha$ and whose range is the set of objects of type $\beta$.

Nothing else is in $Types$.

Examples:

- $\langle s, e \rangle$ is the type of functions from worlds to individuals (i.e., the type of the denotation of an $NP$).
- $\langle s, \langle e, t \rangle \rangle$ is the type of verb phrase meanings: functions from worlds to functions from individuals to truth-values (equiv.: functions from any $w$ to some set of individuals, e.g., the laughers at $w$).
5 Transitive verbs

Transitive verbs denote, in effect, binary relations: world-relative sets of pairs, such that the subject-denotation verbs the object-denotation at the world in question. Suppose at \( w_3 \) the love relation is as follows:

\[
\text{love}(w_3) = \{(a, b), (a, c), (c, c)\}
\]

So we want, for example, \( a \) loves \( b \) to come out true and \( b \) loves \( b \) to come out false. The problem is that direct compositionality requires us to assign a meaning to every phrase constructed during the derivation: we can’t wait until the tree has been constructed and then look for an interpretation. (Aside: the grammars we’re developing don’t make reference to trees; the tree is just a record of the procedure we used to build the sentence and derive its meaning, something that’s useful to us theorists in reasoning about but not available to the grammar.)

So what’s the meaning of \( b \) loves \( b \)? Obviously, it should be the set of individuals who are the first element of some pair in the love relation for which the second element is \( b \). So we need to be able to see inside the relation. Thus, currying (or, if you’re feeling grumpy, you could call it “Schönfinkelization”).

If \( R : A \times B \) is a binary relation, then \( \text{curry}(R) \) is a function \( f : B \to (g : A \to \{0, 1\}) \). For all \( y \in B \), \( \text{curry}(R)(y) \) is the function \( g \) such that, for all \( x \in A \),

\[
g(x) = 1 \iff (x, y) \in R.
\]

You should be able to convince yourself that, for any \( R \), \( (x, y) \in R \) iff \( \text{curry}(R)(y)(x) = 1 \). It’s important here that we target the second members of the pairs first.

Conjunction and disjunction, using the set-based conception:

\[
\begin{align*}
\text{If } \alpha &= ([\alpha], VP, [\alpha]) \text{ and } \beta = ([\beta], VP, [\beta]), \text{ then there is a} \\
\gamma &= ([\alpha \text{-and-} \beta], VP, f : \text{for all } w, f(w) = [\alpha](w) \cap [\beta](w)) \\
\text{If } \alpha &= ([\alpha], VP, [\alpha]) \text{ and } \beta = ([\beta], VP, [\beta]), \text{ then there is a} \\
\gamma &= ([\alpha \text{-or-} \beta], VP, f : \text{for all } w, f(w) = [\alpha](w) \cup [\beta](w))
\end{align*}
\]

So, for all \( w \), \( [\text{laughed and kissed Barack}](w) = \{[\text{laughed}](w) \cap [\text{kissed Barack}](w)\} \). Using the NP-VP rule above, we can now interpret things like Hulk laughed and kissed Barack: it’s the set of worlds \( \{w \mid [\text{Hulk}](w) \in [\text{laughed and kissed Barack}](w)\} \).
6 Categorial grammar (at long last)

CG is developed as a DC-friendly syntactic formalism which also helps relieves ourselves of some of the special-purpose syntactic rules (e.g., the NP-VP rule). Ideally, we want to have as few grammar rules and have them be as general as possible.

The set of syntactic categories $\text{Cat}$ is defined recursively as follows:

- The basic categories $S, NP, N, PP, CP$ are in $\text{Cat}$.
- For any $A, B \in \text{Cat}$, the derived categories $A_L/B$ and $A_R/B$ are in $\text{Cat}$.
- Nothing else is in $\text{Cat}$.

Note the non-accidental similarity to the way we defined the type system above. The choice of basic categories is somewhat arbitrary, but we have to assume something to get started.

Here are some combinatorial rules:

**Right application**: If $\alpha = ([\alpha], A_RB, [\alpha])$ and $\beta = ([\beta], B, [\beta])$, then there is a $\gamma = ([\alpha-\beta], A, f : f(w) = J(\alpha)(w)(J(\beta)(w)))$.

**Left application**: If $\alpha = ([\alpha], A, [\alpha])$ and $\beta = ([\beta], B_LA, [\beta])$, then there is a $\gamma = ([\alpha-\beta], B, f : f(w) = J(\beta)(w)(J(\alpha)(w)))$.

As J. points out (p.78), the only difference between these is in the phonology, so we could easily reduce them to a single parametrized rule.

By adjusting the syntactic categories of the lexical items we can reduce the number of special-purpose grammar rules:

- Intransitive verbs (laughed, walked, ... get category $S_LNP$.
- Transitive verbs (saw, ate, kissed, ... get category $(S_LNP)_RNP$.

We then get trees like

```
S
   /   \\
NP S_LNP
   |    |
Barack laughed
```

```
S
   /   \\
NP S_LNP
   |    |
Hulk (S_LNP)_RNP
   |    |
kissed
   |    |
Barack
```

with the expected denotations: every node’s meaning is the result of applying the denotation of the argument category to the denotation of the functor category, while passing up reference to the world. (These are just begging to be labeled with denotations, as in:
But don’t worry if you can’t read the λ-terms. They’ll be introduced soon.)

7 Reducing directionality specifications

English verb phrases usually look for an argument (a subject) to their left. Most heads look for arguments (complements) to their right. Can we reduce the redundancy in the system, using an undirected slash and general principles to fill these values in? Consider:

- after/within/in the space of/during/... three years
- three years ago
- because of/despite/in the midst of/... your argument
- your argument notwithstanding

It would make sense to have a general principle stating that PPs look for arguments to their right, except for a few items which look left.

- PP/LNP: ago/notwithstanding
- PP/NP: after/within/in the space of/during/because of/despite/in the midst of/...
- **Lexical Rule:** PP/NP $\rightarrow$ PP/RNP

Lots of places we could look to save ink this way. (Just kidding: this kind of generalization is probably important for efficient learning.) For the most part, we’ll just live with the redundancy, but it’s worth keeping in mind that item-by-item specification is not a deep commitment here. (Once again, there’s a non-accidental similarity between Jacobson’s CG and GPSG/HPSG.)

8 Coordination

To make sense of coordination we need metavariables over syntactic categories. We’ll use capital letters $X,Y,Z$ for this purpose.

- $(X/LX)/RX$: and, or
This means that, for any category \( X \in \text{Cat} \), and or may have category \( (X/LX)/RX \). This means that coordination structures have an asymmetric, binary-branching structure, as in:

\[
\begin{array}{c}
\text{NP} \\
\text{NP} \\
\text{Barack} \\
\text{(NP/LNP)/RNP} \\
\text{and} \\
\text{Hulk} \\
\text{laughed} \\
\text{(S/LNP)/L(S/LNP)}/R(S/LNP) \\
\text{ran}
\end{array}
\]

\[
\begin{array}{c}
\text{S/LNP} \\
\text{NP} \\
\text{NP/LNP} \\
\text{NP} \\
\text{S/LNP} \\
\text{NP} \\
\text{(NP/LNP)/RNP} \\
\text{and} \\
\text{Hulk} \\
\text{laughed} \\
\text{(S/LNP)/L(S/LNP)}/R(S/LNP) \\
\text{ran}
\end{array}
\]

9 Features

Our CG will capture facts about agreement, etc. again in an HPSG-like manner: agreement means selection of a head with a particular (e.g.) gender/number, case assignment is selection of an NP with a certain case feature. So we might have a schema like this:

- S/NP: run, walk, jog, laugh, ...
- **Lexical Rule**: \( S/NP \implies S/LNP[NOM] \)

or, when we’re not trying to be parsimonious, we might just write down lexical entries like

- \( S/LNP[SG,NOM] \): runs, walks, jogs, laughs, ...

Presumably number agreement should be introduced through a lexical rule which simultaneously adjusts phonology and syntactic category. (Specifically?)

Argument-marking prepositions, as in *give the book to Mary*: hopefully there are only a few such items. To handle them, we introduce a special feature for each, e.g. \([\text{TO}]\). *Give* will then select for a second object which is a PP with the \([\text{TO}]\) feature.

- \( ([\text{to}], PP[TO]/RNP, I) \), where \( I \) is the identity function
- \( ([\text{give}], ((S/LNP)/RPP[TO])/RNP, [\text{give}]) \)

References