Remarks on ‘Contemporary Linguistics’ 5th edition

Marcus Kracht
Department of Linguistics, UCLA
3125 Campbell Hall
450 Hilgard Avenue
Los Angeles, CA 90095–1543
kracht@humnet.ucla.edu

March 7, 2007

Despite the merits of the book there are a number of things that it either fails to explain or which are inconsistent. I’d like to point out a few of them. These notes are meant to clarify issues that have either come up in discussions or I have noted while reading through the book. I shall keep updating them. Notice that everything I say below is for your benefit only. Wherever I say that things in the book are incorrect I will refrain from checking your knowledge of such matters, thus it will not arise in either the assignments, the midterm or the final. When I elaborate on what the book says, however, this material might come up. Watch the date on this manuscript, as I might update this in the future.

The Feature System

The little circle before the place features, as in [oLABIAL] is an unhelpful way to deal with features. It provides no benefits and is confusing. Basically, rather than working with a distinction between a feature having the values + or −, now we are playing with the distinction between having a feature and not having it.

Here is a way to achieve the same effect. We simply say that not having a feature means that the feature has value −. So, if a sound has labial features, then mark it as [+LABIAL]; if it does not have labial features, mark it as [−LABIAL]. In the latter case, also mark every other feature in that group as minus, for example mark all [−LABIAL] sounds as [−round].

Vowel Quality in American

The vowel system of Standard American is different from that displayed in the handbook of the IPA (collected by Peter Ladefoged, based on a south Californian speaker). The differences are unimportant for the subject matter but may explain why a lot of people have difficulties assessing the identity of certain vowels. The IPA lists in as central vowels [ɹ] (in the place where you find the schwa) and otherwise only [ʌ]. This explains why many of you cannot tell the difference between [ə] and [ʌ] — there is none in the way you speak in case (even and especially) as native Americans. Thus, what elsewhere comes out as [ðə] (`the`) will rather sound [ðʌ]. Similarly, there is no vowel [ɔ] listed in the IPA handbook, only [o] and [ɑ]. Again, this explains why so many have a hard time hearing the difference between vowels that are written as [ɔ] and [ɑ] in the transcription of the words—there is likely to be none in the words.

The status of the glottal stop

The glottal does not appear in the chart of phonemes. Nevertheless, it is part of the sounds that are characterised by the abstract feature system 3.25 of Chapter 3. (Notice that the diphthongs /ow/ and /ej/ have been simplified to /o/ and /e/, respectively.) This is an inconsistency that is not explained anywhere as far as I see.

In present day English every vowel is automatically preceded by a glottal stop *if the onset is empty*. So we have /ʔon/ (`on`), /ʔænsɹ/ (`answer`), but /mek/ (`make`). Another way to view is to say that in English, the onset is never empty, it always consists in at least one sound. By default it is the glottal stop. There is some rationale for this. In Old English verse where the rhyming principle was based on alliteration, which means roughly that the onset is repeated not the rhyme, one possibility for rhyming was to use words with a vowel. However, in Standard

American English, which is what we look at here, none of this can be brought to bear on the analysis.

Finding Syllables

The rules for establishing the syllable boundaries are not laid out explicitly. Let me put them down here for clarity.

1. An English syllable has one and only one vowel. Thus there are as many syllables as there are vowels. (This refers to the phonemes, not the letters, of course.)

2. Vowels and only vowels are nuclear sounds (in English).

3. A *legitimate onset* is any sequence of nonvowels with which a proper English word begins. (Proper words are words of English that are not proper names of people, countries or cities and are not recent loanwords or quotations from other languages.)

4. In a word, the onset belonging to a nucleus is the longest stretch of nonvowels that precedes it in the same word and is a legitimate onset.

5. Any sequence of sounds that spans between a nucleus and the next onset (or the word boundary) is an coda.

Notice that the division into syllables only requires that one knows about legitimate onsets; there is no need to know what the legitimate codas are. Let me stress that this is the algorithm that we are using. It may need refinement, but it is quite accurate. For the purpose of this lecture, please ignore all further complications (for example ambisyllabicity).

Applying Rules

A **rule** is a statement of the form $A \to B/X\_\_Y$. Rules are applied to strings, or—more generally—syntactic representations. Let's focus on rules operating on strings. The part to the left of the slash indicates the change that the rule introduces if it at all applies; the right hand side tells us on what condition it applies.

It does so by telling us what the environment of A must be in order for the rule to apply. To begin with the latter, the parts $X\_\_Y$ says that the rule applies if the parts that undergoes change finds itself immediately to the right of X and immediately to the left of Y. X and Y may be strings, but in general can be properties of strings; they may include abstract symbols, such as # (word boundary) or V (vowel), [+sonorant]. If they are empty, the condition is void. For example, if X is empty the rule applies if A is followed by Y; if Y is empty, the rule applies if A is followed by X. If both are empty it applies to all occurrences of A.

Now let us be given an arbitrary string $\vec{x}$; we call $\vec{x}$ the **input** to the rule. The **output** is what the rule returns. Two things may happen: the does not rule apply, or it does. Moreover, if it applies, it may apply at different places. Let us look at these cases in turn.

① The rules does not apply. In this case the rule does not change $\vec{x}$. $\vec{x}$ is also the **output**.

② The rule does apply. In this case it may choose an occurrence of A in $\vec{x}$ and replace it with B. This gives a new string $\vec{y}$, which is then the output.

Let the rule be `a` $\to$ `A/□__`. This means that the rule applies if it has an occurrence of `a` at its beginning of a word, or more exactly, right next to an occurrence of □.

① The rule does *not* apply to `□cat□`. Thus, if the input is `□cat□`, the output is `□cat□` again.

② The rule does apply to `□apple□`. The only occurrence of `a` is right after □, so the context condition is fulfilled. The rule applies by taking out that occurrence: `□__pple□` and then inserting `A`: `□Apple□`. This is the output.

Since □ is a symbol (it is the word boundary, which we normally write with a blank) we may also create strings containing more instances of it, for example: `□apples□are□healthy□`. If this is the input to the rule, there is a choice as to where it can be applied. There are three occurrences of `a`; the first two are at the beginning of a word (`apple`, `are`), the third one is not. So, applying the rule once gives us two possible outcomes:

❶ The environment of the first occurrence is `□__pples□are□healthy□`. We insert `A` into this environment and get `□Apples□are□healthy□`.

❷ The environment of the second occurrence is `□apples□__re□healthy□`. We insert `A` into this environment and get `□apples□Are□healthy□`.

❸ The environment of the third occurrence is `□apples□are□he__lthy□`. This occurrence may not be chosen, as `e` (which precedes the underscore) is not identical to `□`.

Notice that if *some* environment matches the condition the rule *must* be applied. (Otherwise the rule is called **optional**). It may then be applied to *any occurrence that matches the conditions on the environment*.

It does *not* matter whether we think of the rules as acting on strings of letters or strings of sounds. The concept of a rule stays the same. The only difference is the kind of object to which it applies. Certainly, we will not want to apply a rule designed for pronunciation to be applied to strings of letters and vice versa.

A and B may be strings, but we may also take advantage of the feature decomposition. There are two important cases to look at. Write $\varnothing$ for the empty string. The rule $\varnothing \to B/X\_\_Y$ will, when applicable, *insert* B; the rule The rule $A \to \varnothing/X\_\_Y$ will, when applicable, *delete* A.

Context Free Rules

A special topic, not covered in the book, are the **context free rules**. This is a rule where the context is empty. More exactly, it has this form

(1) $$X \to Y_1 Y_2 \cdots Y_n$$

where there is one letter to the left (X) and one or several letters to the right (the Y_i). We could write this rules as follows.

(2) $$X \to Y_1 Y_2 \cdots Y_n/\_\_$$

But it is generally agreed that the context condition is not mentioned. The format (2) shows us that the rule can be applied to any X, regardless what is found to its left and regardless what is found to its right. (That is why the rules are called context free.)

X-bar syntax can be written using context free rules. Here are some rules of X-bar syntax:

$$\text{(3)} \qquad \begin{array}{l} \text{NP} \rightarrow \text{Det N}' \\ \text{NP} \rightarrow \text{N}' \\ \text{N}' \rightarrow \text{N PP} \\ \text{N}' \rightarrow \text{N} \end{array}$$

We can collapse two rules with the same symbol to the left as follows:

$$\text{(4)} \qquad \begin{array}{l} \text{NP} \rightarrow \text{Det N}' \mid \text{N}' \\ \text{N}' \rightarrow \text{N PP} \mid \text{N} \end{array}$$

The vertical slash indicates a choice between the items on its left and on its right. It is not part of the rule. For example, in the first line the slash separates the string Det N′ and N′. This means that we have *two* rules, one where we replace NP by Det N′, and another where we replace it by N′.

These rules are interpreted as follows. Suppose we have a string NP. Then by the first rule we are allowed to replace this by the sequence Det N′. In this sequence, both Det and N′ are seen as single symbols (just accidentally written using several characters). Following this, we are allows to replace N′ by N PP, for example. This means that the sequence Det N′ becomes Det N PP upon involving that rule. We could use another rule, the last one for example, and get instead Det N. Now we have exhausted the rules, however. There is nothing that we can do using the rules. But notice that the rules above only talk about projections of Ns and—moreover—only with Ps as complement. In particular, nothing is said about PP is general. So, we add another set of rules:

$$\text{(5)} \qquad \begin{array}{l} \text{PP} \rightarrow \text{Deg P}' \\ \text{PP} \rightarrow \text{P}' \\ \text{P}' \rightarrow \text{P NP} \\ \text{P}' \rightarrow \text{P} \end{array}$$

In shorthand we may write this as

$$\text{(6)} \qquad \begin{array}{l} \text{PP} \rightarrow \text{Deg P}' \mid \text{P}' \\ \text{P}' \rightarrow \text{P NP} \mid \text{P} \end{array}$$

And so we can continue the derivation. From Det N PP we can get to Det N Deg P′ and, finally, Det N Deg P NP. At this point we call our rules upstairs and replace NP first by Det N′ and then by Det N. The sequence we get is

(7) Det N Deg P Det N.

We still need to fill in the words, though. There are two ideas. One is to simply use context free rules:

(8)
$$\begin{aligned} \mathrm{N} &\to \texttt{house} \mid \texttt{cat} \mid \texttt{theorem} \mid \texttt{idea} \mid \ldots \\ \mathrm{Det} &\to \texttt{the} \mid \texttt{a} \mid \ldots \\ \mathrm{Deg} &\to \texttt{almost} \mid \texttt{right} \mid \ldots \\ \mathrm{P} &\to \texttt{in} \mid \texttt{under} \mid \ldots \end{aligned}$$

That means that wherever we find N we may choose to replace this either by `house` or by `cat` or by `car` or by whatever follows in this list. Thus, applying the rules we can go on as follows:

(9)
Det N Deg P Det N
Det `house` Deg P Det N
`a house` Deg P Det N
`a house` Deg `under` Det N
`a house right under` Det N
`a house right under the` N
`a house right under the cat`

Make sure you check that in passing from one line to the next I have applied just one rule to one particular item. What I obtained was an understandable, though somewhat unusual, phrase of English.

Typically, X-bar syntax is written in an abstract form as follows.

(10)
$$\begin{aligned} \mathrm{XP} &\to \mathrm{Z\ X'} \\ \mathrm{XP} &\to \mathrm{X'} \\ \mathrm{X'} &\to \mathrm{X\ YP} \\ \mathrm{X'} &\to \mathrm{X} \end{aligned}$$

Here, X, Y and Z are placeholders for categories; X can be A, V, N or P. Y can again by A, V, N or P. Finally, Z depends on what X is. If X = N then Z = Det.

However the rules are more generous than we would like. For example, we can derive the phrase `a idea under the house`. Apart from some morphological problem (that the indefinite `a` occurs when `an` would be appropriate) there is also the problem that `idea` does not take a complement PP of the form `under` NP. Similarly, we need to distinguish verbs that take a direct object (an NP) from those that do not. (The NP complement of a verb is also called its **direct object**.) How can this be achieved? X-bar syntax certainly does not allow us to do this. However, notice that we do not really *have* to use context free rules. We may employ context sensitive (ie non context free) rules if necessary. So, we may say, for example,

(11) $\quad$ V → `devour` | `prove` | . . . /__NP
$\quad$ V → `eat` | `run` | . . .

The first batch of rules is not context sensitive: it says that V may be replaced by `devour` or `prove` (or any of the other elements denoted by the dots) only if followed by NP. Thus, the VP `prove the theorem` has the following derivation.

(12) VP
(13) V′
(14) V NP
(15) `prove` NP
(16) `prove` Det N′
(17) `prove` Det N
(18) `prove the` N
(19) `prove the theorem`

Watch for the transition from (15) to the next line. The context condition is satisfied.

A sequence of strings like the one from (12) to (19) is called a **derivation**. A general definition is this. Let *G* be a set of rules. A derivation in *G* is a sequence of strings such that each string except the first is derived from the one preceding it but applying one of the rules of *G*. This concept does not require the rules to be context free; any set of rules is fine. In principle a derivation can begin with any string. However, a grammar will always have one designated symbol with which a derivation must start. In our case, a derivation must start with CP.

Meaning relations

Let us be given two sentences, S and T. We say that S **entails** T if whenever S is true, T is true as well. An example is: `John lives in Los Angeles` and `John lives in California`. We write as follows:

(20) $$\frac{\texttt{John lives in Los Angeles.}}{\therefore\ \texttt{John lives in California.}}$$

It is also customary to write on the following:

(21) $\texttt{John lives in Los Angeles.} \vdash \texttt{John lives in California.}$

(22) $\texttt{John lives in Los Angeles.} \supset \texttt{John lives in California.}$

(23) $\texttt{John lives in Los Angeles.} \to \texttt{John lives in California.}$

I recommend only the first usage (that is, (21)) in addition to (20). Two things can be immediately deduced. (a) Any sentence entails itself. For if S is true then S is true. Trivial as it may seem, there is a tendency to overlook this fact. (b) If S entails T and T entails U then S entails U. For assume that S is true. Then since S entails T, T is also true. And since T entails U, U is true.

We say that S and T are **synonymous** or **truth-conditionally equivalent** if S entails T and T entails S. We write $S \equiv T$. The following can now be shown:

① For all S: $S \equiv S$.

② For all S and T: if $S \equiv T$ then $T \equiv S$.

③ For all S, T and U: if $S \equiv T$ and $T \equiv U$ then $S \equiv U$.

Meaning relations between words can generally be reduced to meaning relations between sentences. For example, let *w* and *v* be some nouns (for example, `dog` and `animal`). *w* is a **hyponym** of *v* if the sentence `Here is a` *w*`.` entails the sentence `Here is a` *v*`.`. For example, we have a valid argument of the form

(24) $$\frac{\texttt{Here is a dog.}}{\therefore\ \texttt{Here is an animal.}}$$

Therefore, *w* is a hyponym of *v*. The following is then clear from the above: any noun is a hyponym of itself; and if *w* is a hyponym of *v* and *v* a hyponym of *u* then *w* is a hyponym of *u*. Again, *w* and *v* are said to be **synonymous** if *w* is a hyponym of *v* and *v* a hyponym of *w*. This is a particular case of Leibniz' Principle.

> *Leibniz Principle.* Two constituents C and D (which can be words) are *synonymous* iff for any sentence S containing one of them, say C, and S′ the result of replacing an occurrence of C by D, S and S′ have the same truth.

I do not expect you to know (or apply this principle). But it is interesting to note that it reduces synonymy entirely to truth for any two constituents.

Thematics Roles

First of all, thematic roles are generally identified via some semantic criterion. The ideal case is that of a verb. The verb `to hit` denotes an event that involves two participants: one who is doing the hitting, and someone or something that is being hit. The book mentions the following θ-roles: **agent**, **theme**, **location**, **source** and **goal**. I added a sixth: **experiencer**. I should be pretty clear what sort of role that is: someone (human or animal) that is said to be in an emotional state. Notice that the role must be encoded in the verb meaning. If John kicks a stone he certainly has a particular emotion but nothing in the verb ‘to kick’ says that he does. Thus, the subject of `kick` is not an experiencer. But it is an agent since to kick entails to actively do something.

The role of θ-role assignment is somewhat unclearly stated in the book. It is different for verbs and prepositions. A verb gets in NPs and PPs of various forms, and all of them serve a role, that is, they are said to be *assigned a θ-role by the verb*. The preposition `to` does not assign a θ-role to its complement in the same way. In fact, it is the entire expression `to the shop` that is given to the verb, and the verb will assign to it the θ-role goal. So, we may think of `to` as creating a goal from an NP. However, when doing exercises and during the final, such details will not matter, and you may continue to use the notation given in the book.