Workshop I *Progress in Generative Grammar: Its Characterization and Assessment*
November 10 (Saturday), 2007.

Hypotheses in Generative Grammar

Ayumi Ueyama (Kyushu University, JAPAN)

This presentation is concerned with the questions in (1) in the context of Generative Grammar. It addresses, and suggests ways to grapple with, the difficulties in answering (1).

(1) a. What kind of generalization has to be explained in Generative Grammar?
    b. What is a testable manifestation of a prediction/consequence that is deducible from a theory of the Computational System?
    c. How can a prediction be tested?

Before going into each of (1), I briefly state why such issues have to be taken up.

1. Why do we need such discussion?

Generative grammar draws a clear distinction between (linguistic) competence and performance, and declares that the object of investigation is competence. The model for linguistic competence is often called *Computational System*; it is a set of operations and it generates abstract representations that underlie or correspond to a so-called sentence. The sentences which can be derived by this system are *grammatical*, while those which cannot are *ungrammatical*. Thus, if the distinction between grammatical and ungrammatical sentences *were* directly observable, the investigation of competence *could* ideally be carried out as follows:

(2) 1. Identify some grammatical and ungrammatical sentences.
    2. Hypothesize the Computational System so that it can derive the former but not the latter.
    3. Deduce a prediction based on the hypothesized Computational System.
    4. Test the prediction.

The data in actual research in generative grammar, however, is based on acceptability judgments on a given sentence (under a specified interpretation). As is well-known, acceptability judgments do not necessarily coincide with (un)grammaticality, and they are quite easy to be affected by factors of performance. This makes it imperative that we provide a theory of how a hypothesis can be tested, by specifying the relationship between acceptability and grammaticality, and this is what Y. Deguchi has recently dubbed as *the need of phenomenology in generative grammar*.

Since the 1960's, Chomsky's attitude towards the phenomenology for Generative Grammar has been that there is no need of methodology as such, insofar as each researcher is rational enough. His opinion might have been appropriate in the 1960s, but the situation is no longer the same now since the theory of competence has been drastically changed compared to the one that was assumed in the 1960s. The most fundamental difference lies in the disposal of Phrase Structure Rules and the introduction of Numeration/Merge that has taken over the former. When Phrase Structure Rules were assumed, grammar was a static system that defined the properties of grammatical sentences, and the actual linguistic activity was expected to make use of this knowledge. The relation between the grammatical knowledge and the actual linguistic activity, however, remained too abstract to construct a theory for it. But once Numeration/Merge has been introduced, the Computational System (CS) can be regarded as a dynamic system which actually constructs 'sentences', making it possible to construct a model of the actual linguistic activity with the CS embedded at the center of it, i.e., making the phenomenology for Generative Grammar a real possibility.

2. Model of acceptability judgments

There is a huge and in fact fundamental great difference between acceptability and grammaticality. For example, making an acceptability judgment (under a specified interpretation) can be schematized as in (3). Grammaticality, on the other hand, is a notion having to do with whether the CS generates an output or not.

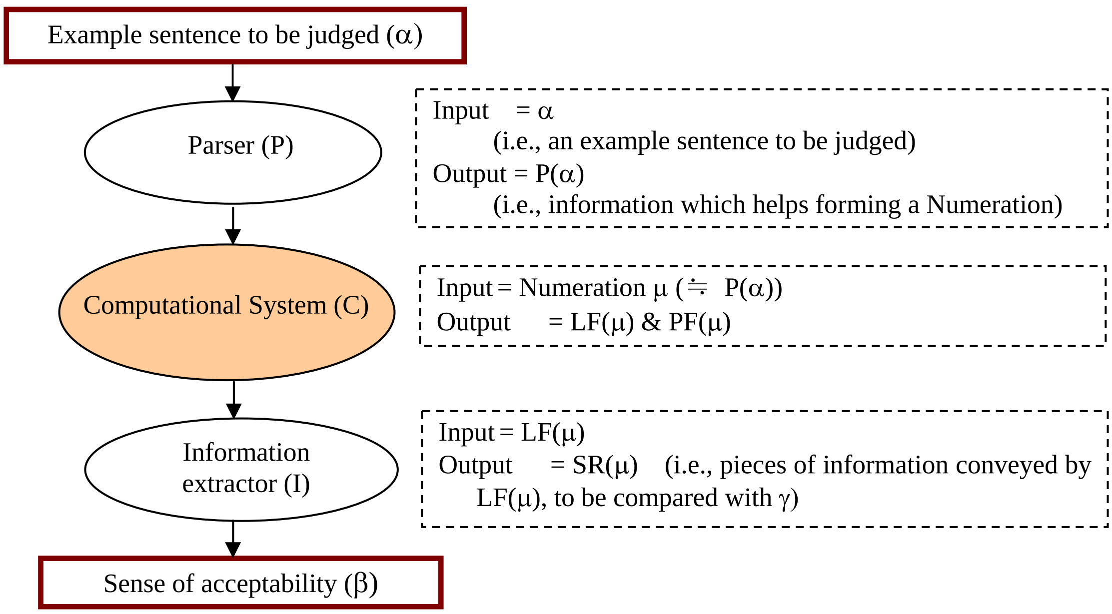
(3) Making acceptability judgments (under a specified interpretation)
Input: an example sentence
Output: the sense of acceptability

(4) Computational System:
Input: Numeration
Output: LF & PF

But (4) must be embedded in the activity in (3); otherwise, it would not make sense to check the speaker intuitions in an attempt to discover the CS-related properties. If the effects of the factors that are not directly related to the CS can be controlled, it should be possible to address grammaticality on the basis of acceptability, hence extract the former from the latter, so to speak.

Suppose one is asked how acceptable a sentence α is under a specified interpretation γ. The model of acceptability judgments that we assume can be outlined as in (5).

(5)



The activity of acceptability judgment consists of checking each of (6a) and (6b).

(6) a. whether PF(μ) is nondistinct from α
b. whether SR(μ) satisfies the conditions (that must in turn be satisfied in order) for γ (to be possible).

The sense of acceptability β can thus be formalized as follows.

(7) Sense of acceptability:
β = [C] − [P] − [I] + error, where
[C] is 1, if both (6a) and (6b) are confirmed; otherwise, [C] is 0.
[P] is some value (between 0 and 1) which expresses the difficulty in Parsing.
[I] is some value (between 0 and 1) which expresses the difficulty in Information extraction.

3. Towards a phenomenology for Generative Grammar

Let us return to the questions in (1) in light of the model of acceptability judgment presented above.

3.1. Generalizations that have to be explained

(1) a. What kind of generalization has to be explained in generative grammar?

Given that the main object of investigation of generative grammar is the mechanism of the Computational System, the answer to (1a) has to be something which is explained in terms of (8).

(8) a. the properties of a lexical item
b. the applicability of an operation on a lexical item
c. the output conditions on LF or PF

Here we should not fail to recognize the difference between Phrase Structure Rules (=PSRs) and Merge. PSRs define the patterns of grammatical sentences, while Merge does not define anything: it is merely an operation that combines two things to form one. Suppose one has recognized the existence of a certain pattern/construction. This might have been an object of explanation under the theory which contains PSRs, since some rule or other may be needed to generate the new pattern. On the other hand, the mere existence of a particular pattern is not an object of explanation by itself under the theory with Merge, since almost every combination of words can be generated from this theory; rather, the appropriate constraints are needed so that ungrammatical 'sentences' may not be generated by this mechanism. Therefore, the generalization to be explained is an observation such as (9):

(9) *Schema:
A sentence is unacceptable as long as it fails to satisfy condition X.

In order to make sure that the unacceptability in question is indeed due to the failure of "condition X" to be satisfied, (9) has to be accompanied by (10).

(10) ^{ok}Schema:
A sentence can be acceptable if it satisfies condition X.

Thus, our answer to (1a) is as in (11).

(1) a. What kind of generalization has to be explained in Generative Grammar?
(11) Answer to (1a):
An observation that can be stated as a pair of *Schema and ^{ok}Schema.

3.2. Predictions to be tested

Let us move on to (1b).

(1) b. What is a testable manifestation of the prediction/consequence that is deducible from a theory of Computational System?

The answer is quite straightforward given our formula of the sense of acceptability in (7) repeated here, and an independent assumption that the degrees below 0 cannot be distinguished from one another.

(7) Sense of acceptability:
$\beta = [C] - [P] - [I] + \text{error}$, where
[C] is 1, if both (6a) and (6b) are confirmed; otherwise, [C] is 0.
[P] is some value (between 0 and 1) which expresses the difficulty in Parsing.
[I] is some value (between 0 and 1) which expresses the difficulty in Information extraction.

If [C] is 0, the value of β does not get affected by [P] and [I], given the independent assumption that the degrees below 0 cannot be distinguished from one another. (And this means that ungrammaticality necessarily manifests itself as unacceptability.) When [P] and [I] are considerably large, the value of β can easily get close to 0, regardless of the value of [C]. (And this means that grammaticality does not necessarily manifest itself as acceptability, so to speak.) Thus the value of β being close to 0 may be due to [C] being 0, but it may instead be due to [P] and/or [I] being considerably large. It thus follows that the contrast in [C] can be observed clearly only if [P] and [I] are held to be low enough.

Our answer to (1b) is thus as in (12).

(1) b. What is a testable manifestation of the prediction/consequence that is deducible from a theory of Computational System?

(12) Answer to (1b):
a. The sense of acceptability of a sentence of *Schema should always be 0 (or lower).
b. The contrast between *Schema and ^{ok}Schema is observable only if [P] and [I] are held low enough.

3.3. Experiments and the evaluation of their results

Our answer to (1c) is as in (13).

(1) c. How can a prediction be tested?

(13) Answer to (1c):
By checking the informant judgment on a *Schema and its corresponding ^{ok}Schema.

The sense of acceptability of a *Schema can vary depending on the following two factors.

(14) a. an actual example sentence which realizes the *Schema
b. a person who makes the judgment

In order to obtain a representative value for a *Schema (or for an ^{ok}Schema), we need to know how to get the value, including how to arrive at it statistically covering both the informants and the different instantiations of the same *Schema and of the same ^{ok}Schema. Suppose that the representative value for a *Schema is close to 0 and that for the corresponding ^{ok}Schema is far better than 0. We shall say that such a pair of *Schema and ^{ok}Schema constitutes a *licensed generalization*, and we want it to follow from the CS-related properties that we hypothesize.

4. Progress in Generative Grammar

What has prompted the preceding considerations is the concern for ensuring progress in generative grammar. In light of the preceding remarks, it is possible to characterize *progress in generative grammar* as follows.

(15) a. The more *licensed generalizations*, the better.
b. The more unlicensed generalization get identified as such and expelled from the arena of theoretical inquiry into the nature of the CS, the better.

(16) The more *licensed generalizations* are expressed within the theory, i.e., in terms of the postulated concepts and relations, the better.

(17) The fewer theoretical concepts and relations are needed for expressing the *licensed generalizations*, the better.

As (17) gets achieved, many *licensed generalizations* will likely fall under a more abstract generalization. But no matter how abstract a generalization we might end up entertaining, it will be based ultimately on *licensed generalizations*, whose validity can be tested as suggested above.