Formal Description of Grammar and Translation (II)

By Akihiro NOZAKI

Institute of Mathematics, College of General Education, University of Tokyo

(Received February 4, 1966)

Introduction

As it is well known, the procedure of generating sentences in a language is often simplified by relevant stratification of the grammar of the language. If the stratification is well matched with the language, it will be also useful for simplifying other automatic processing of the language, e. g., automatic translation to or from the language, with due alternation of details.

In the following, we shall describe a two-level stratification of translation procedure as a practical model. To show its feasibility, a part of the procedure will be illustrated in an algorithmic language.

1. *Formal Description of Levels*

A grammar G is a triple $(B, \sum, F)$ consisting of a set B of basic symbols, a set $\sum$ of symbols and a set F of rewriting rules (CF. [1].)

It is often convenient to assume that the set F is partially ordered and that every derivation

$$\varDelta : \ \alpha_0 \xrightarrow{f_1} \alpha_1 \xrightarrow{f_2} \cdots\cdots \xrightarrow{f_n} \alpha_n$$

should satisfy the following condition:

If $f_i<f_j$, *then* $i<j$.

In other words, if $f<g$ then the rule g should not be, or at least need not be, applied previously to the rule f.

Under such restriction on derivations, the set $D(G)$ of derived strings and the set $A(G)$ of admissible strings are re-defined. We call the grammar G as a *grammar with partial order.*

When the set F is partially ordered, its dual F^* will have the dual order as following:

$$f^* < g^* \quad \text{if and only if} \quad f > g.$$

Then the duality between G and G^* holds also for the grammars with partial orders.

Let $F_1, \cdots, F_n$ be some subsets of F satisfying the following conditions:

(1) $\qquad F=F_1 \cup F_2 \cup \cdots\cdots \cup F_n$

(2) if $g \in F_i$, $h \in F_j$ and $i<j$, then always $g<h$,

or

(3) if $g \in F_i$, $h \in F_j$ and $i<j$, then always $g>h$.

We shall call these subsets as *levels* of the grammar G.
The family $[F_1, \cdots, F_n]$ of these levels is called a *stratification* of G and G is called a *grammar with* n *levels.*

There is no essential difference between (2) and (3). For convenience, we assume the condition (2) for generative grammars while (3) for analytical grammars.

A grammar $G_1=(B, \Sigma, F_1)$ is called the *kernel* of G with respect to the stratification.

If $L=D(G)$, i.e. G is a generative grammar of L, then every sentence of L can be derived from a sentence in $D(G_1)$ using the rules in $F_2, \cdots, F_n$. If $L=A(G)$, then every sentence of L can be reduced to a sentence in $A(G_1)$.

2. *General Scheme of Two-level Translation*

Let us consider translation from L to L'. We assume that L has its analytical grammar G with two levels F_1 and F_2, and that L' has its generative grammar also with two levels F'_1 and F'_2. Then each derivation in G or G' can be splitted as following:

$$\Delta: \underbrace{\alpha_0 \to \alpha_1 \to \cdots \to \alpha_j}_{\Delta_2}\underbrace{\to \cdots \to \alpha_n}_{\Delta_1}=S$$

where $\alpha_j \in A(G_1)$.

$$\Delta': \underbrace{\beta_m \leftarrow \beta_{m-1} \leftarrow \cdots \leftarrow \beta_k}_{\Delta'_2}\underbrace{\leftarrow \cdots \leftarrow \beta_0}_{\Delta'_1}=S$$

where $\beta_k \in D(G'_1)$.
The translation procedure $\Phi: \Delta \to \Delta'$ will also be splitted:

$$\Delta'_1=\Phi_1(\Delta_1, \Delta_2)$$

$$\Delta'_2=\Phi_2(\Delta_1, \Delta_2)$$

These mappings are expected to be rather simple by setting adequate levels in G and G'.

To continue our investigation more substantially, it seems very convenient to employ the following hypotheses:
Working Hypothesis I.

We can take suitable phrase structure grammars G_1 and G'_1 so that the mapping Φ_1 can be given by a contravariant functor $\varphi: F_1 \to F_2$:

$$\Delta_1: \alpha_j \xrightarrow{f_{j+1}} \cdots\cdots \xrightarrow{f_n} \alpha_n=S$$

$$\Delta'_1: \beta_k \xleftarrow{\varphi(f_{j+1})} \cdots\cdots \xleftarrow{\varphi(f_n)} \beta_0=S$$

Consequently $k=n-j+1$. Δ'_1 depends only on Δ_1.

Working Hypothesis II

We can write down a "program" for the function Φ_2 in an algorithmic language.

The input data for the program will be accumulated through the process of analyzing the original sentence according to G.

Let us start with an attempt of constructing the program for Φ_2

3. *A Program for Φ_2*

In this section, we consider English as an example of the target language L'. As a grammar G' of L', we take the grammar described in [2]. Then F'_2 contains the following rules which are not of phrase structure type:

(1) Passive Transformation

$$T_p : NP+AUX+V+NP' \rightarrow NP'+AUX+\text{be}+\text{en}+V+\text{by}+NP.$$

(2) Negative Transformation

$$T_n :\ NP+C+\left\{\begin{matrix} V \\ M \\ \text{have} \\ \text{be} \end{matrix}\right\} \longrightarrow NP+C+n't+\left\{\begin{matrix} V \\ M \\ \text{have} \\ \text{be} \end{matrix}\right\}$$

(3) Interrogative Transformation

$$T_q :\ NP+C+\left\{\begin{matrix} V \\ M \\ \text{have} \\ \text{be} \end{matrix}\right\} \longrightarrow C+NP+\left\{\begin{matrix} V \\ M \\ \text{have} \\ \text{be} \end{matrix}\right\}$$

These are optional transformations. Therefore enough information should have been accumulated for the program of Φ_2 to determine whether each of these rules have to be applied or not.

The grammar G of the source language will probably contain an analytical passive transformation "T^p." Then it will be one reasonable resolution to associate an assignment statement such as

"passive :=*true*"

to the rule T^p in order to provide good information for Φ_2.

The identifier "passive" denotes of course a Boolean variable indicating whether the original sentence (or the subsentence under analysis) is passive (*true*) or not (*false*).

The value of the variable is initially set to be *false*. It is turned to *true* only by the above mentioned assignment statement, or in other words only when the rule T^p is applied.

We assume that other Boolean variables "negative" and "interrogative" are also defined in a similar way.

We can now illustrate a part of the desired program using these variables.

```
If passive=true then apply ($T^p$);
apply ($T^{ob}_{sep}$);
If negative=true then apply ($T_n$);
If interrogative=true then
    begin apply ($T_q$);
        $X+WH+NP+Y \longrightarrow WH+NP+X+Y$;
        $WH+\begin{Bmatrix}\text{ANIMATE NOUN}\\ \text{INANIMATE NOUN}\end{Bmatrix} \longrightarrow \begin{Bmatrix}\text{WHO}\\ \text{WHAT}\end{Bmatrix}$        end
```

Variables utilized above are independent parameters of the function Φ_2. Suppose that there are n variables which are sufficient to describe a complete program for a simple sentence containing one single predicate.

Then every particular combination of values of these variables is represented by an n-tuple, or an n-dimensional vector $(a_1, \cdots, a_n)$.

If an input sentence contains more than two predicates, then there must be the same number of such vectors. Every predicate (or rather, subsentence) P has its own vector $(x_1, \cdots, x_n)$, which is hereafter called the *structural index* $S(P)$ of P.

The program for a simple sentence will then be set to work recursively to every subsentence.

As the components of structural index $S(P)=(x_1, \cdots, x_n)$, the following items seem to be important:

subordinate clause or not,
interrogative or not,
negative or not,
past or not,
progressive or not,
passive or not,
perfect or not,
conditional or not,
imperative or not,
exclamatory or not,
person of the subject,
gender of the subject,
number of the subject,

These items make clear the requirement to the construction of an analytical grammar G of L.

4. *Construction of Kernels*

When a grammar G' of the target language L' is quite different from a grammar G of the source language L, many problems are involved in realizing a contravariant functor $\varphi : F_1 \rightarrow F'_1$. First of all, we have to check every *PS* rule

f in F_1 if there exists a relevant *PS* rule $\varphi(f)$ in F'_1. If not, the rule f has to be removed from F_1 and added to F_2, even though the rule f itself is expressed as a simple *PS* rule.

A good example is found in Japanese-English translation. In Japanese, it is very easy to change an active sentence into passive. We have only to add a small word "RERU" at the end of the sentence. Such a simple rule $f: S \rightarrow S + \text{RERU}$, however, doesn't correspond to a simple *PS* rule in English grammar. Thus the rule f has to be excluded from F_1 so that the hypothesis I could be satisfied rigidly.

Of course it is not at all easy to construct an adequate pair of grammars (G, G') for every pair of languages (L, L'). We should therefore be satisfied with a sufficiently simple kernel K, for instance, of generative type which might be extremely simple in general but could apply to many translations.

Such a kernel K of a grammar G will not contain the rules to specify the modification and/or the transformation indicated by structural indices described in the preceding section. It will contain truely basic rules as following:

$S \rightarrow \text{GERM} + \text{INDEX}$

$\text{GERM} \rightarrow \text{Subject} + M + V + \text{Object}$, etc.

$\text{INDEX} \rightarrow \text{Past, Passive, Past} + \text{Passive}, \cdots$

Subject, Object$\rightarrow NP$

$NP \rightarrow \text{Noun}, AP + NP$

$AP \rightarrow \text{Adjective}, S + \text{Subordinate} + \text{INDEX}$

Generated sentences by K may not be a sentence in $D(G)$. They shall be further modified and/or transformed by the rules in G, for example;

T_s (English): $S + \text{Subordinate} + NP \rightarrow NP + S$

T_s (Japanese): $\text{Subordinate} \rightarrow \phi$

T_s (German): $(X + V + Y) + \text{Subordinate} + NP$

$\rightarrow NP + (X + Y + V)$.

The kernel K will generate only "basic patterns of sentences," or "skeletons," i.e. combinations of verbs and nouns in essence. Any change in word order are in principle specified by the rules which don't belong to K.

If the translation from $D(G)$ is required, the dual grammars G^* and K^* will be useful.

For the sake of clarity, an example of translation utilizing K is shown below. Many items of structural index are omitted in the following for the simplicity of description.

$L = \text{German}$, $L' = \text{English}$.

(1) Analytical Derivation

$$\left.\begin{array}{l}
\text{Es war ein König, der drei Töchter hatte.}\\
\rightarrow \text{Es}+\text{Sein}+\text{König}+(\text{der}+\text{Tochter}+\text{Haben}+\text{Past})+\text{Past.}\\
\rightarrow \text{Es}+\text{Sein}+(\text{der}+\text{Haben}+\text{Tochter}+\text{Past})+\text{Subordinate}\\
\quad +\text{König}+\text{Past.}\\
\quad \vdots\\
\rightarrow \text{Es}+\text{Sein}+S+\text{Subordinate}+NP+\text{Past.}\\
\rightarrow \text{Es}+\text{Sein}+AP+NP+\text{Past.}\\
\rightarrow \text{Es}+\text{Sein}+NP+\text{Past.}\\
\rightarrow \text{Es}+\text{Sein}+NP+\text{INDEX.}\\
\rightarrow \text{GERM}+\text{INDEX}\\
\rightarrow S.
\end{array}\right\}\ \begin{array}{c}\text{derivation}\\ \text{by } K_L\end{array}$$

(2) Generative Derivation

$$\left.\begin{array}{l}
\text{S}\\
\rightarrow \text{GERM}+\text{INDEX}\\
\rightarrow \text{There}+\text{Be}+NP+\text{INDEX.}\\
\rightarrow \text{There}+\text{Be}+NP+\text{Past.}\\
\rightarrow \text{There}+\text{Be}+AP+NP+\text{Past.}\\
\rightarrow \text{There}+\text{Be}+\ S\ +\text{Subordinate}+NP+\text{Past.}\\
\quad \vdots\\
\rightarrow \text{There}+\text{Be}+(\mathit{WHO}+\text{Have}+\text{Daughter}+\text{Past})+\text{King}+\text{Past.}
\end{array}\right\}\ \begin{array}{c}\text{derivation}\\ \text{by } K_{L'}\end{array}$$

$$\begin{array}{l}
\rightarrow \text{There}+\text{Be}+\text{King}+(\mathit{WHO}+\text{Have}+\text{Daughter}+\text{Past})+\text{Past.}\\
\quad \vdots\\
\rightarrow \text{There was a king who had three daughters.}
\end{array}$$

Remark

The main problem left to be solved in this connection is precise description and classification of the rules each of which rewrites "GERM" into a particular combination of a verb and nouns. Roughly speaking, it could be regarded as the problem of classifying predicates, and larger part of the classification could be made on the ground of semantics, under little influence of singularity of individual syntax.

References

[1] A. NOZAKI, Formal Description of Grammar and Translation (I). *Sci. Pap. Col. Gen. Educ., Univ. of Tokyo*, **15**, 29-33. (1965)

[2] N. CHOMSKY, Syntactic Structures, *Mouton* & *Co., the Hague.* (1957)