A Note on Recursive Languages

TERUAKI FUJITA*

1. *Introduction*

Any programming language is described by (1) a constituent structure grammar which can be stated in Backus normal form (BNF), and (2) supplementary informal rules. For example, the description of ALGOL 60 by BNF in [1] is associated with informally stated "semantical" rules. Among these rules are, e. g., restrictions on the scope of identifiers and labels, which are not semantical but syntactical rules.

To trace a given program, one applies these syntactical rules —formal or informal— and can decide, within a finite time interval, whether it is syntactically correct or not. This fact suggests that actual programming languages, considered as sets of words, will be recursive.

Context-free (CF) languages form a typical family of recursive languages. However, as mentioned above, any existing programming language is a subset of a context-free language. We can not decide whether it is recursive or not, unless the informal syntactical rules are formalized.

The purpose of this paper is to present a general procedure of formalization. ALGOL 60 is adopted as an example, just because it is the only language the CF grammar of which is fairly completely stated in BNF.

2. *Recursiveness of ALGOL 60*

Formalization of ALGOL 60 will be done by constructing a logical system. The construction will be divided into three steps.

(a) Arithmetize ALGOL 60. That is, a recursive one-to-one mapping (Gödel numbering) of the set of sequences of "basic symbols" into the set $\boldsymbol{N}$ of non-negative integers will be defined.

(b) Define predicates each of which "represents" each nonterminal ("metalinguistic variable") of ALGOL 60. That is, for each nonterminal X, we will define a predicate Ξ on the numbers of N, which has the property that, for any terminal string φ

$$\Xi(\bar{\varphi}) \equiv X \overset{*}{\Rightarrow} \varphi,$$

where $\bar{\varphi}$ is the Gödel number of φ. Each Ξ will be proved to be recursive.

(c) Especially the distinguished nonterminal ⟨program⟩ defines a predicate Prog. We define a predicate WFP (well-formed program) as the conjunction of Prog. with predicates corresponding to informally stated syntactical rules. Then WFP is recursive. This means

This paper first appeared in Japanese in Joho Shori (the Journal of the Information Processing Society of Japan), Vol. 8, No. 4 (1967), pp. 188~193.

* Mitsui Mutual Life Insurance Company.

that the language generated by ALGOL 60 grammar is recursive.

3. *Arithmetization of CF language*

Steps (a) and (b) eventually follow a common pattern in every CF language. So we will state the procedure in a form of metatheory for CF languages.

Assume that a CF grammar $G=(V, \Sigma, P, \sigma)$ is given. (Terminology and notation for CF languages in this paper will follow that of [2].) As is well known, we can assume that G satisfies the condition:

(*) G is a reduced grammar, and it contains no production of the form $X \rightarrow X$ $(X \in V - \Sigma)$.

Step (a)

At first we must introduce an appropriate ordering in V and must assign odd numbers 3, 5, 7, ... to the members (characters) of V. To guarantee recursiveness of the predicates to be defined in step (b), some caution is necessary for the ordering. Here we will adopt the "canonical ordering" of Ingerman [3]. In the Ingerman's algorithm, subscripts are assigned to characters of V. Arrange characters of V in the *descending* order of subscripts. Thus σ will be the last character in this ordering. Correspond numbers 3, 5, 7, ... to the resulting sequence of characters.

Let us denote this mapping by 'p'; that is, $p(X)$ is the number corresponding to X, $X \in V$. Given any string $\varphi = X_1 X_2 \ldots X_n \in V^*$, the Gödel number $\bar{\varphi}$ of φ is defined by:

$$\bar{\varphi} = 2^{p(X_1)} 3^{p(X_2)} \ldots p_n^{p(X_n)}$$

where p_i is the i-th prime number.

Step (b)

For any nonterminal X, let $D(X) \subset V^*$ be the set of derivatives of X; that is,

$$D(X) = \{\varphi \mid X \overset{*}{\Rightarrow} \varphi\}.$$

Under the condition (*), D(X) is recursive and thus defines a recursive predicate Ξ on the numbers in $\boldsymbol{N}$. By definition,

$$\Xi(\bar{\varphi}) \equiv X \overset{*}{\Rightarrow} \varphi;$$

thus Ξ "represents" X.

References

[1] Naur, P., Revised Report on the Algorithmic Language ALGOL 60. *Comm. ACM, 6*, 1 (1963), 1-17.

[2] Ginsburg, S., *The Mathematical Theory of Context Free Languages*, McGraw-Hill (1966).

[3] Ingerman, P. Z., *A Syntax-Oriented Translator*, Academic Press (1966).