On the Ambiguity Problem of Backus Systems*

David G. Cantor

Princeton University, Princeton, New Jersey

Backus [1] has developed an elegant method of defining well-formed formulas for computer languages such as Algol. It consists of (our notation is slightly different from that of Backus):

(I) A finite alphabet: $a_1, a_2, \cdots, a_t$;

(II) Predicates: $P_1, P_2, \cdots, P_s$;

(III) Productions, either of the form (a) $a_j \in P_i$;

or of the form (b) $P_{i_1}P_{i_2} \cdots P_{i_t} \rightarrow P_j$.

A *word* is a finite sequence of letters from the alphabet. Then IIIa states that certain words (containing only one letter) belong initially to some of the predicates, and IIIb states that if words $W_1, W_2, \cdots, W_t$ belong to the predicates $P_{i_1}, P_{i_2}, \cdots, P_{i_t}$ respectively, then the concatenation $W_1W_2 \cdots W_t$ belongs to P_j. We call this a *Backus* system.

A simple example of such a system is:

Alphabet: a, b;

Predicates: P, Q, R;

Productions: $a \in P$, $b \in Q$, $PQ \rightarrow R$, $QP \rightarrow R$; $RR \rightarrow R$, $PRQ \rightarrow R$, $QRP \rightarrow R$.

Then P and Q contain only the words a and b, respectively, while R contains all words which have the same number of a's and b's.

In the above example, $abab$ belongs to R and can be produced in two ways. Namely, as $ab \in R$ and $RR \rightarrow R$, $abab \in R$; also as $ba \in R$ and $PRQ \rightarrow R$, $abab \in R$. We call a Backus system *ambiguous* if one of its predicates contains a word which can be produced in more than one way. As, in practice, the meaning of a word is determined by the way it is produced, an ambiguous Backus System must be avoided.

As the following example illustrates, Algol 60 [3] is ambiguous:

if $B \wedge C$ **then for** $I := 1$ **step** 1 **until** N **do if** $D \vee E$ **then** $A[I] := 0$ **else** $K := K + 1$; $K := K - 1$

In fact, both

for $I := 1$ **step** 1 **until** N **do if** $D \vee E$ **then** $A[I] := 0$

and

for $I := 1$ **step** 1 **until** N **do if** $D \vee E$ **then** $A[I] := 0$ **else** $K := K + 1$

are valid **for** statements of Algol 60. Combining the first with

if $B \wedge C$ **then** $\cdots$ **else** $K = K + 1$;

or the second with

if $B \wedge C$ **then** $\cdots$

* Received March, 1962 This research was supported in part by the Office of Naval Research Logistics Project, Princeton University.

gives rise to the above example, and these two methods of construction correspond to the two possible meanings of the example.

D. Dahm and H. Trotter, in a private communication, have raised the question: "Does there exist an algorithm to determine whether a Backus system is ambiguous?" We call this the *ambiguity problem*. The purpose of this paper is to show that no such algorithm exists, i.e., that the ambiguity problem is unsolvable.

We first define a *normal system*. It consists of:

(I) A finite alphabet: $a_1, a_2, \cdots, a_t$;

(II) A finite collection of ordered pairs: $(g_1, \bar{g}_1), (g_2, \bar{g}_2), \cdots, (g_r, \bar{g}_r)$, where the g_i and $\bar{g}_i$ are words.

(III) An axiom A which is some fixed word.

If U and V are words, we say $U \rightarrow V$ if U is of the form gP and V is of the form $P\bar{g}$ where $(g, \bar{g})$ is one of the ordered pairs. We also write, in this case, $g_i P \rightarrow P\bar{g}_i$. Also, if $U_1, U_2, \cdots, U_n$ are words with $U_i \rightarrow U_{i+1}$, $1 \leqq i \leqq n-1$, then $U_1 \rightarrow U_n$, and we say U_n is derived from U_1. The words which may be derived from the axiom A are called *theorems*.

A normal system is called *undecidable* if there does not exist an algorithm for determining whether a word is a theorem of the system. It is implicit in [2, sec. 6.5] that there exists an undecidable normal system, which we denote by NS, with the property that in each ordered pair $(g, \bar{g})$, the words g and $\bar{g}$ have no common letters.

LEMMA. *If U and V are words of NS, then $U \rightarrow V$, if and only if there exists indices $j_1, j_2, \cdots, j_m$ such that*

$$U\bar{g}_{j_1}\bar{g}_{j_2} \cdots \bar{g}_{j_m} = g_{j_1}g_{j_2} \cdots g_{j_m}V.$$

PROOF. Suppose the equality holds. As $\bar{g}_{j_1}$ and g_{j_1} have no common letters, U is of the form $g_{j_1}R_1$; let $U_1 = R_1\bar{g}_{j_1}$. Then we have $U \rightarrow U_1$ and $U_1\bar{g}_{j_2} \cdots \bar{g}_{j_m} = g_{j_2}g_{j_3} \cdots g_{j_m}V$. Proceeding inductively, we obtain a sequence of words, $U, U_1, U_2, \cdots, U_m = V$ with $U \rightarrow U_1 \rightarrow \cdots \rightarrow U_m$; hence $U \rightarrow V$. Conversely, if $U \rightarrow V$, then there exist words $U_0, U_1, \cdots, U_m$ with $U_0 = U$ and $U_m = V$, and indices $j_1, j_2, \cdots, j_m$ such that $U_{i-1}\bar{g}_{j_i} = g_{j_i}U_i$, $1 \leqq i \leqq m$. Then $U_0\bar{g}_{j_1} = g_{j_1}U_1$ or $U_0\bar{g}_{j_1}\bar{g}_{j_2} = g_{j_1}U_1\bar{g}_{j_2} = g_{j_1}g_{j_2}U_2$. By induction the proof is complete.

THEOREM. *The ambiguity problem is unsolvable.*

PROOF. We describe certain predicates and Backus systems; to save space we omit the formal definitions. It is easy to construct predicates and systems with the required properties. We use as alphabet the alphabet $a_1, a_2, \cdots, a_t$ of NS and in addition the letters $b_1, b_2, \cdots, b_r$, one for each ordered pair $(g_i, \bar{g}_i)$ of NS. If A is the axiom of NS, form the predicate P which contains all words of the form

$$b_{j_m}b_{j_{m-1}} \cdots b_{j_1}A\bar{g}_{j_1}\bar{g}_{j_2} \cdots \bar{g}_{j_m};$$

if W is any word on the alphabet $a_1, a_2, \cdots, a_r$, let Q_W be the predicate containing all words of the form

$$b_{j_m} b_{j_{m-1}} \cdots b_{j_1} g_{j_1} g_{j_2} \cdots g_{j_m} W.$$

It is possible to construct the predicates P and Q_W so that there is no ambiguity in their definition, and we assume that this is done. Then form the Backus system B_W which contains the predicates P, Q_W, and S_W, where S_W is defined by $P \rightarrow S_W$ and $Q_W \rightarrow S_W$.

Now, in order for B_W to be ambiguous, B_W must contain a predicate which contains a word which comes about in two ways. The predicates P and Q_W, and all predicates used in their definition, do not have this property. Thus B_W is ambiguous if and only if S_W contains a word which comes about in two ways. From the definition of S_W, it is clear that B_W is ambiguous if and only if P and Q_W have a word in common. Observing the form of the words in P and Q_W we see that B_W is ambiguous if and only if there exists indices $j_1, j_2, \cdots, j_m$ such that $b_{j_m} \cdots b_{j_1} A \bar{g}_{j_1} \cdots \bar{g}_{j_m} = b_{j_m} \cdots b_{j_1} g_{j_1} \cdots g_{j_m} W$. By the lemma, this is true if and only if $A \rightarrow W$. Thus if the ambiguity problem for Backus systems were solvable, then the decision problem for NS would be solvable, which is not the case. Hence the ambiguity problem is unsolvable.

REFERENCES

1. Backus, J. W. ACM-GAMM Conference ICIP, June 1959.
2. Davis, Martin. Computability and unsolvability. New York, 1958.
3. Naur, Peter, et al. Report on the algorithmic language ALGOL 60. *Comm. ACM 3*, 5 (May 1960).