language alone. The distinction between grammar and language is extremely important when semantics is being considered as well as syntax.

The grammar

$$S \rightarrow aAd,\ S \rightarrow bAB,\ A \rightarrow cA,\ A \rightarrow c,\ B \rightarrow d \tag{8}$$

has the sentential forms $\{ac^nAd\} \cup \{ac^{n+1}d\} \cup \{bc^nAB\} \cup \{bc^nAd\} \cup \{bc^{n+1}B\} \cup \{bc^{n+1}d\}$. In the string $bc^{n+1}d$, d must be replaced by B, while in the string $ac^{n+1}d$, this replacement must not be made; so the decision depends on an unbounded number of characters to the left of d, and the grammar is not of bounded context (nor is it translatable from right to left). On the other hand this grammar is clearly LR(1) and in fact it is of bounded right context since the handle is immediately known by considering the character to its right and two characters to its left; when the character d is considered the sentential form will have been reduced to either aAd or bAd.

The grammar

$$S \rightarrow aA,\ S \rightarrow bB,\ A \rightarrow cA,\ A \rightarrow d,\ B \rightarrow cB,\ B \rightarrow d \tag{9}$$

is not of bounded right context, since the handle in both ac^nd and bc^nd is "d"; yet this grammar is certainly LR(0). A more interesting example is

$$S \rightarrow aAc,\ S \rightarrow b,\ A \rightarrow aSc,\ A \rightarrow b. \tag{10}$$

Here the terminal strings are $\{a^nbc^n\}$, and the b must be reduced to S or A according as n is even or odd. This is another LR(0) grammar which fails to be of bounded right context.

In Section III we will give further examples and will discuss the relevance of these concepts to the grammar for ALGOL 60. Section IV contains a proof that the existence of k, such that a given grammar is LR(k), is recursively undecidable.

Ginsburg and Greibach (1965) have defined the notion of a *deterministic language*; we show in Section V that these are precisely the languages for which *there exists* an LR(k) grammar, and thereby we obtain a number of interesting consequences.

II. ANALYSIS OF LR(k) GRAMMARS

Given a grammar $\mathcal{G}$ and an integer $k \geqq 0$, we will now give two ways to test whether $\mathcal{G}$ is LR(k) or not. We may assume as usual that $\mathcal{G}$ does