methods for translating computer languages (for example, the well-known precedence algorithm, see Floyd (1963)) are based on the fact that the languages being considered have a simple left-to-right structure. By considering all languages that are translatable from left to right, we are able to study all of these special techniques in their most general framework, and to find for a given language and grammar the "best possible" way to translate it from left to right. The study of such languages is also of possible interest to those who are investigating human parsing behavior, perhaps helping to explain the fact that certain English sentences are unintelligible to a listener.

Now we proceed to give precise definitions to the concepts discussed informally above. The well-known properties of *characters* and *strings* of characters will be assumed. We are given two disjoint sets of characters, the "*intermediates*" I and the "*terminals*" T; we will use upper case letters $A, B, C, \cdots$ to stand for intermediates, and lower case letters $a, b, c, \cdots$ to stand for terminals, and the letters X, Y, Z will be used to denote either intermediates or terminals. The letter S denotes the "principal intermediate character" which has special significance as explained below. Strings of characters will be denoted by lower case Greek letters $\alpha, \beta, \gamma, \cdots$, and the *empty string* will be represented by ϵ. The notation α^n denotes n-fold concatenation of string α with itself; $\alpha^0 = \epsilon$, and $\alpha^n = \alpha\alpha^{n-1}$. A *production* is a relation $A \rightarrow \theta$ where A is in I and θ is a string on $I \cup T$; a *grammar* $\mathcal{G}$ is a set of productions. We write $\varphi \rightarrow \psi$ (with respect to $\mathcal{G}$, a grammar which is usually understood) if there exist strings $\alpha, \theta, \omega, A$ such that $\varphi = \alpha A \omega$, $\psi = \alpha\theta\omega$, and $A \rightarrow \theta$ is a production in $\mathcal{G}$. The transitive completion of this relation is of principal importance: $\alpha \Rightarrow \beta$ means there exist strings $\alpha_0, \alpha_1, \cdots, \alpha_n$ (with $n > 0$) for which $\alpha = \alpha_0 \rightarrow \alpha_1 \rightarrow \cdots \rightarrow \alpha_n = \beta$. Note that by this definition it is not necessarily true that $\alpha \Rightarrow \alpha$; we will write $\alpha \equiv> \beta$ to mean $\alpha = \beta$ or $\alpha \Rightarrow \beta$. A grammar is said to be *circular* if $\alpha \Rightarrow \alpha$ for some α. (Some of this notation is more complicated than we would need for the purposes of the present paper, but it is introduced in this way in order to be compatible with that used in related papers.) The *language defined by* $\mathcal{G}$ is

$$\{\alpha \mid S \Rightarrow \alpha \text{ and } \alpha \text{ is a string over } T\}, \tag{1}$$

namely, the set of all terminal strings derivable from S by using the productions of $\mathcal{G}$ as substitution rules. A *sentential form* is any string α for which $S \Rightarrow \alpha$.