certain form of push-down parsing program from a bounded context
grammar, and Earley (1964) independently developed a somewhat
similar method which was applicable to a rather large number of LR(1)
languages but had several important omissions. Floyd (1964a) also
introduced the more general notion of a bounded right context grammar;
in our terminology, this is an LR(k) grammar in which one knows
whether or not \(X_{r+1} \cdots X_n\) is the handle by examining only a given
finite number of characters immediately to the left of \(X_{r+1}\), as well as
knowing \(X_{n+1} \cdots X_{n+k}\). At that time it seemed plausible that a bounded
right context grammar was the natural way to formalize the intuitive
notion of a grammar by which one could translate from left to right with-
out backing up or looking ahead by more than a given distance; but it
was possible to show that Earley's construction provided a parsing
method for some grammars which were not of bounded right context,
although intuitively they should have been, and this led to the above
definition of an LR(k) grammar (in which the entire string to the left of
\(X_{r+1}\) is known).

It is natural to ask if we can in fact always parse the strings corre-
sponding to an LR(k) grammar by going from left to right. Since there
are an infinite number of strings \(X_1 \cdots X_{n+k}\) which must be used to make
a parsing decision, we might need infinite wisdom to be able to make
this decision correctly; the definition of LR(k) merely says a correct
decision exists for each of these infinitely many strings. But it will be
shown in Section II that only a finite number of essential possibilities
really exist.

Now we will present a few examples to illustrate these notions. Con-
sider the following two grammars:

\[
S \rightarrow aAc, A \rightarrow bAb, A \rightarrow b. \tag{6}
\]

\[
S \rightarrow aAc, A \rightarrow Abb, A \rightarrow b. \tag{7}
\]

Both of these are unambiguous and they define the same language,
\(\{ab^{2n+1}c\}\). Grammar (6) is not LR(k) for any \(k\), since given the partial
string \(ab^n\) there is no information by which we can replace any \(b\) by \(A\);
parsing must wait until the "c" has been read. On the other hand gram-
mar (7) is LR(0), in fact it is a bounded context language; the sentential
forms are \(\{aAb^{2n}c\}\) and \(\{ab^{2n+1}c\}\), and to parse we must reduce a substring
\(ab\) to \(aA\), a substring \(Abb\) to \(A\), and a substring \(aAc\) to \(S\). This example
shows that LR(k) is definitely a property of the grammar, not of the