$LR(k)$ **Grammars**

Adding Lookahead

- Goal: Similar to $LL(k)$ for NTA, we want to resolve nondeterminism in NBA by allowing a lookahead of k input symbols

- The start-separated CFG $G = < \Sigma, N, P, S >$ is an $LR(k)$ grammar for a given $k \in \mathbb{N}$ if for all rightmost derivations of the form

$$S \begin{cases} \Rightarrow_r^* \alpha A w \Rightarrow_r \alpha\beta w \\ \Rightarrow_r^* \gamma B x \Rightarrow_r \alpha\beta y \end{cases} \text{such that } first_k(w) = first_k(y)$$

it follows that $\alpha = \gamma, A = B$, and $x = y$

$LR(0)$ Grammars

- In contrast to $LL(0)$ grammars, $LR(0)$ grammars are actually interesting
- G is an $LR(0)$ grammar if for all rightmost derivations of the form

$$S \begin{cases} \Rightarrow_r^* \alpha A w \Rightarrow_r \alpha\beta w \\ \Rightarrow_r^* \gamma B x \Rightarrow_r \alpha\beta y \end{cases}$$

  it follows that $\alpha = \gamma, A = B$, and $x = y$

- For the automaton, this means that the decision to reduce or shift in the state (w, δ, z) only depends on δ, without needing to look ahead into the input w

Example of an $LR(0)$ grammar

- Start-separated grammar

$$
\begin{array}{ll}
S' \rightarrow S & (0) \\
S \rightarrow B|C & (1,2) \\
B \rightarrow aB|b & (3,4) \\
C \rightarrow aC|c & (5,6)
\end{array}
$$

- Input ab
  - Initial state $(ab, \varepsilon, \varepsilon)$
  - Shift is only possible action (b, a, ε)
  - Shift is only possible action $(\varepsilon, ab, \varepsilon)$
  - Only one rule to reduce b $(\varepsilon, aB, 4)$
  - Only one rule to reduce aB $(\varepsilon, C, 43)$
  - Only one rule to reduce C $(\varepsilon, S, 431)$
  - Final state $(\varepsilon, S', 4310)$