

3. Se β puder derivar ϵ ($\beta \xRightarrow{*} \epsilon$), $b \in \text{follow}(A)$ e α puder derivar b ($\alpha \xRightarrow{*} b\gamma$), então tanto $A ::= \alpha$ quanto $A ::= \beta$ poderão ser escolhidas para expandir A quando b for o símbolo corrente da entrada. É claro que a utilização de $A ::= \alpha$ seria correta, já que $b \in \text{first}(\alpha)$. Para enxergar porque a utilização de $A ::= \beta$ produziria o resultado correto, considere o seguinte exemplo: $\gamma A \phi$ está sendo expandido para reconhecer a entrada, b é o token corrente, γ já foi reconhecido, A deve ser derivado, $b \in \text{first}(\phi)$ e, portanto, $b \in \text{follow}(A)$. Escolhendo $A ::= \beta$ e $\beta \xRightarrow{*} \epsilon$, teríamos $\gamma A \phi \xRightarrow{+} \gamma \phi$ onde b seria reconhecido por ϕ .

Nem todas as gramáticas são ou podem ser transformadas em LL(1). Este é o principal obstáculo ao uso de analisadores preditivos. Felizmente, o problema de entradas na tabela com mais de um elemento pode ser resolvido escolhendo-se um deles, removendo assim a ambigüidade.

Analisadores ascendentes (*botton-up*), que derivam o símbolo inicial a partir dos terminais, podem ser utilizados com um número maior de gramáticas do que analisadores descendentes preditivos. Estes analisadores são, em geral, gerados automaticamente por *geradores de analisadores sintáticos* a partir da gramática da linguagem. Analisadores ascendentes empregam grandes tabelas que seriam difíceis de serem gerados por pessoas sem o auxílio de programas. Como exemplos de geradores de analisadores sintáticos, temos o YACC e o Bison.