

Conjunto FIRST

Definição

Seja α uma seqüência qualquer gerada por G. Definimos como sendo **first**(α) o conjunto de símbolos terminais que iniciam α ou seqüências derivadas (direta ou indiretamente) de α .

Observações: Se $\alpha = \varepsilon$ ou $\alpha \xRightarrow{*} \varepsilon$, então $\varepsilon \in \mathbf{first}(\alpha)$.

Algoritmo

Para calcular **first**(X) para todo $X \in V_n \cup V_t$, aplicamos as seguintes regras

- a) se $X \in V_t$, então **first**(X) = {X};
- b) se $X \in V_n \wedge X \rightarrow a\alpha \in P$, então coloque a em **first**(X); da mesma forma, se $X \rightarrow \varepsilon \in P$, coloque ε em **first**(X);
- c) se $X \rightarrow Y_1 Y_2 \dots Y_k \in P$, então, para todo $i \mid Y_1 Y_2 \dots Y_{i-1} \in V_n \wedge \mathbf{first}(Y_j)$, para $j = i, i-1$, contenha ε , adicione **first**(Y_i) – { ε } em **first**(X).

Em outras palavras

- 1 – coloque **first**(Y_1), exceto ε , em **first**(X);
- 2 – se $\varepsilon \in \mathbf{first}(Y_1)$ então coloque **first**(Y_2), exceto ε em **first**(X);
- 3 – se $\varepsilon \in \mathbf{first}(Y_2)$ então
.....
até Y_k .
- 4 – finalmente, se para todo i (de 1 a k) **first**(Y_i) contém ε , então adicione ε em **first**(X).

Conjunto FOLLOW

Definição

Definimos **follow**(A), para todo $A \in V_n$, como sendo o conjunto de símbolos terminais que podem aparecer imediatamente após A em alguma forma sentencial de G.

Algoritmo

Para todo $A \in V_n$, aplique as regras abaixo, até que **follow**(A) esteja completo (isto é, não sofra nenhuma alteração)

- 1 – coloque \$ (a marca de final de sentença) em **follow**(S), onde S é o símbolo inicial da gramática em questão;
- 2 – se $A \rightarrow \alpha B \beta \in P \wedge \beta \neq \epsilon$,
então adicione **first**(β), exceto ϵ , em **follow**(B);
- 3 – se $A \rightarrow \alpha B$ (ou $A \rightarrow \alpha B \beta$, onde $\epsilon \in \mathbf{first}(\beta) \in P$),
então adicione **follow**(A) em **follow**(B).

Tabela de Parsing

Construção da tabela de parsing do Parser Preditivo

Idéia Geral

Se $A \rightarrow \alpha \in P \wedge \underline{a} \in \mathbf{first}(\alpha)$, então, se \underline{A} está no topo da pilha e \underline{a} é o próximo símbolo da entrada, devemos expandir (derivar) \underline{A} , usando a produção $\underline{A} \rightarrow \alpha$. Mas, e se $\alpha = \varepsilon$ ou $\alpha \xrightarrow{*} \varepsilon$ – note que ε nunca aparecerá na entrada – neste caso, se $\underline{a} \in \mathbf{follow}(A)$ expandimos (derivamos) \underline{A} através da produção $A \rightarrow \alpha$.

Algoritmo

- 1 – para cada produção $A \rightarrow \alpha \in P$, execute os passos 2 e 3;
- 2 – para todo $\underline{a} \in \mathbf{first}(\alpha)$, exceto ε ,
coloque o número da produção $A \rightarrow \alpha$ em $M(A, a)$;
- 3 – se $\varepsilon \in \mathbf{first}(\alpha)$, coloque o número da produção $A \rightarrow \alpha$ em $M(A, b)$, para todo $b \in \mathbf{follow}(A)$;
- 4 – as posições de M que ficarem indefinidas, representarão as situações de erro.

Considerações Gerais

A tabela de parsing dos analisadores preditivos deve possuir a propriedade de que, em cada entrada da tabela M exista no máximo uma produção (seu número); isto viabiliza a análise determinística da sentença de entrada.

Para que esta propriedade se verifique, a gramática deverá satisfazer as seguintes condições

- 1 – não possuir recursão à esquerda;
- 2 – estar fatorada;
- 3 – para todo $A \in V_n \mid A \xrightarrow{*} \varepsilon$, $\mathbf{first}(A) \cap \mathbf{follow}(A) = \varnothing$

As GLCs que satisfazem estas condições são denominadas GLC. **LL(K)** – isto é GLC. que podem ser analisadas deterministicamente da esquerda para a direita (**L**eft-to-right) e o analisador construirá uma derivação mais à esquerda (**L**eftmost derivation), sendo necessário a cada passo o conhecimento de **K** símbolos de *lookahead* (símbolos de entrada que podem ser vistos para que uma ação seja determinada).

Somente GLCs LL(K) podem ser analisadas pelos analisadores preditivos (as demais causam conflitos na construção da tabela de parsing ou fazem com que o analisador entre loop); por isso, os analisadores preditivos são também denominados analisadores LL(K) – na prática usa-se $K = 1$, obtendo-se desta forma **analisadores LL(1)** para **GLC LL(1)**.