Chapter 4: Syntactic Analyser

Part 4d2: First and Next relationships
Symbols that can generate the empty word

Procedure

• We say that a non-terminal symbol X can generate the empty word, and we write it

   \[ X \rightarrow^* \lambda \]

   if:

   • There is a production rule of the form \( X \rightarrow \lambda \).
   • There is a non-terminal symbol B such that
     • \( B \rightarrow^* \lambda \), and
     • There is a production rule of the form \( X \rightarrow B \),

FIRST relationship

Definition

• Formally,

Let \( G = (\Sigma_I, \Sigma_T, P, \Lambda) \) be a context-independent grammar.

If \( \alpha \) is a sentential form of the grammar, \( (\alpha \in (\Sigma_I \cup \Sigma_T)^*) \), we shall call \( \text{first}(\alpha) \) the set of terminal symbols that can be at the beginning of the strings derived from \( \alpha \).

In the case that the empty string \( \lambda \) can also be derived from \( \alpha \), then we shall say that \( \lambda \) also belongs to \( \text{first}(\alpha) \).
The following is an algorithm for calculating $\text{first}(X) \ \forall \ X \in \Sigma_N \cup \Sigma_T$:

1. Initialise $\text{first}(X)$ as an array.
2. If $X \in \Sigma_T$ then $\text{first}(X) = \{X\}$. Return it.
3. If $X \rightarrow ^* \lambda \in P$ then insert $\lambda$ in $\text{first}(X)$.
4. If $X \rightarrow Y_1 Y_2 \ldots Y_k \in P$ then
   - For $i$ from 1 to $k$,
     - Every element from $\text{first}(Y_i)$ has to be in $\text{first}(X)$
     - If $Y_i \rightarrow ^* \lambda \in P$ then continue the loop;
     - Otherwise, break.

If $\alpha = X_1 X_2 \ldots X_n$ then
- If $X_i \rightarrow ^* \lambda \ \forall i \in \{1, 2, \ldots, n\}$ then $\lambda \in \text{first}(\alpha)$.
- If $j \in \{1, 2, \ldots, n\}$ is the first sub-index such that $X_j \rightarrow ^* \lambda$ does not hold, then $\text{first}(Y_j) \subseteq \text{first}(X) \ \forall i \in \{1, 2, \ldots, j\}$.
**NEXT relationship**

**Definition**

- Formally,

Let $G = \langle \Sigma_n, \Sigma_t, P, A \rangle$ be a context independent grammar.

For any non-terminal symbol $X$, we shall call $\text{next}(X)$ the set of terminal symbols that can be generated immediately after $X$.

If $X$ can be generated at the rightmost position in sentential form generated by the grammar, then $\$$ $\in$ next $(X)$.

**Algorithm**

- The following pseudo-code calculates $\text{next}(X)$ for all $X \in \Sigma_n$:

1. Initialise next $(X)$ as an array.
2. If $X = A$ (axiom) add $\$$ to next $(X)$.
3. $\forall Y \rightarrow \alpha X \beta \in P, \beta \neq \lambda \wedge \lambda \notin \text{first}(\beta)$ $\Rightarrow$ first $(\beta) \subseteq \text{next}(X)$.
4. $\forall Y \rightarrow \alpha X \beta \in P, \beta \neq \lambda \wedge \lambda \in \text{first}(\beta)$.
   - Then $\text{first}(\beta) - \{\lambda\} \subseteq \text{next}(X)$.
   - And next $(Y) \subseteq \text{next}(X)$.
5. $\forall Y \rightarrow \alpha X \in P \Rightarrow \text{next}(Y) \subseteq \text{next}(X)$.
**FIRST and NEXT sets**

**Exercise**

- Given the following grammar, 

\[
G=\langle E, E', T, T', F, +, *, (,), id \rangle \\
\{ \quad E \to TE' \\
\quad E' \to +TE' | \lambda \\
\quad T \to FT' \\
\quad T' \to *FT' | \lambda \\
\quad F \to (E) | id \} , \\
E> 
\]

- Find all the symbols which can generate the empty word.
- Find the first set for all the non-terminal symbols.
- Find the next set for all the non-terminal symbols.

---

**FIRST and NEXT sets**

**Exercise**

- Given the following grammar, 

\[
G=\langle E, A, B, C, D, E, F, h, i, j, k, l \rangle \\
\{ \quad E \to A \mid B \\
\quad A \to D \mid ED \\
\quad D \to CE \mid \lambda \\
\quad B \to BB \mid F \mid k \mid \lambda \\
\quad F \to hi \mid kkl \\
\quad C \to j \mid \lambda \\
\} , \\
E> 
\]

- Find all the symbols which can generate the empty word.
- Find the first set for all the non-terminal symbols.
- Find the next set for all the non-terminal symbols.