Compilers

3^{er} course
Spring Term

Alfonso Ortega: alfonso.ortega@uam.es
Enrique Alfonseca: enrique.alfonseca@uam.es

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 $X \rightarrow B$,

3

FIRST relationship

Definition

- Formally,

Let $G=\{\Sigma_N, \Sigma_T, P, A\}$ be a context-independent grammar.

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

In the case that the empty string λ can also be derived from α, then we shall say that λ also belongs to **first(α)**

4

FIRST relationship

Algorithm

- The following is an algorithm for calculating $\texttt{first(X)}\ \forall\ X \in \Sigma_N \cup \Sigma_T$:

> The following rules will be applied, until we cannot add any new non-terminal symbols, or λ
> 1. Initialise `first(X)` as an array.
> 2. If $X \in \Sigma_T$ then $\texttt{first(X)}=\{X\}$. Return it.
> 3. If $X \rightarrow^* \lambda \in P$ then insert λ in `first(X)`.
> 4. If $X \rightarrow Y_1 Y_2 ... Y_k \in P$ then
>   - For `i` from 1 to k,
>   - Every element from $\texttt{first}(Y_i)$ has to be in `first(X)`
>   - If $Y_i \rightarrow^* \lambda \in P$ then continue the loop;
>   - Otherwise, break.

5

FIRST relationship

Algorithm

- Description of the algorithm to calculate $\texttt{first}(\alpha)\ \forall\ \alpha \in (\Sigma_N \cup \Sigma_T)^*$:

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

6

NEXT relationship

Definition

- Formally,

Let $G=\{\Sigma_N, \Sigma_T, P, A\}$ be a context independent grammar.
For any non-terminal symbol X, we shall call **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)**

7

NEXT relationship

Algorithm

- The following pseudo-code calculates next(X) $\forall\ X \in \Sigma_N$:

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

8

FIRST and NEXT sets

Exercise

- Given the following grammar,

```
G=< {E,E',T,T',F},
    {+,*,(,),id}
    {   E   → TE'
        E'  → +TE'  | λ
        T   → FT'
        T'  → *FT'  | λ
        F   → (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.

9

FIRST and NEXT sets

Exercise

- Given the following grammar,

```
G=< {E,A,B,C,D,E,F},
    {h,i,j,k,l}
    {   E   → A | B
        A   → D | ED
        D   → CE | λ
        B   → BB | F | k | λ
        F   → hi | kkl
        C   → j | λ
    },
    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.

10