

Syntaxanalyse

Bestimmung der Predict Mengen

Definition:

Ist $N \rightarrow s$, $s \in A^*$ eine Produktion, dann heisst die Entscheidungsmenge:

$\text{predict}(N \rightarrow s) = \text{first}(s \text{ follow}(N))$

$\text{predict}(N \rightarrow s)$ ist die Menge der Terminalsymbole, mit denen eine Satzform bei Verwenden der Produktion $N \rightarrow s$ beginnen kann.

Syntaktische Analyse

Prof. Dr. U. Göhner HS Kempten 85

Syntaxanalyse

Bestimmung der Predict Mengen

Satz:

Eine kontextfreie Grammatik ist genau dann eine LL(1)-Grammatik, wenn für alle Produktionen $N \rightarrow s_1 | s_2 \dots | s_n$ gilt:

$\text{Predict}(N \rightarrow s_1) \cap \text{Predict}(N \rightarrow s_2) = \emptyset$, $\text{Predict}(N \rightarrow s_1) \cap \text{Predict}(N \rightarrow s_3) = \emptyset, \dots$

Syntaktische Analyse

Prof. Dr. U. Göhner HS Kempten 86

Syntaxanalyse

LL(1)-Grammatik

Satz:

Eine kontextfreie Grammatik G ist genau dann eine LL(1)-Grammatik, wenn gilt:

G erlaubt eine Sackgassenfreie Top-Down-Analyse von links nach rechts, d.h. wenn es alternative Produktionen $N \rightarrow s_1 | s_2 \dots$ gibt, dann kann mit Kenntnis des nächsten Symbols die richtige Ableitung gewählt werden.

2 Fälle:

$$1. s \neq \epsilon : \text{predict}(N \rightarrow s) = \text{First}(s)$$

$$2. s \rightarrow \epsilon : \text{predict}(N \rightarrow s) = \text{First}(s) \setminus \{\epsilon\} \cup \text{Follow}(N)$$

Beispiel: $S \rightarrow AB$;
 $A \rightarrow aA \mid \epsilon$;
 $B \rightarrow bB \mid \epsilon$;

$$\text{predict}(A \rightarrow aA):$$

$$\text{Fall 1.} : \text{predict}(A \rightarrow aA) = \text{First}(aA) = \{a\}$$

$$\text{predict}(A \rightarrow \epsilon):$$

$$\text{Fall 2} : \text{predict}(A \rightarrow \epsilon) = \text{Follow}(A) = \text{First}(B) \setminus \{\epsilon\} \cup \text{Follow}(S) = \{b, \$\}$$

$$\text{predict}(B \rightarrow bB):$$

$$\text{Fall 1} : \text{predict}(B \rightarrow bB) = \text{First}(bB) = \{b\}$$

$$\text{predict}(B \rightarrow \epsilon):$$

$$\text{Fall 2} : \text{predict}(B \rightarrow \epsilon) = \text{Follow}(B) = \text{Follow}(S) = \{\$\}$$

LL(1)-Eigenschaft ist erfüllt

$$\text{predict}(S \rightarrow AB)$$

$$\text{Fall 2} : \text{predict}(S \rightarrow AB) = \text{First}(AB) \setminus \{\epsilon\} \cup \text{Follow}(S) = \{a, b, \$\}$$

LL(1)-Grammatik

Satz:

G erlaubt eine Sackgassenfreie Top-Down-Analyse von links nach rechts, d.h. wenn es alternative Produktionen $N \rightarrow s_1 \mid s_2 \dots$ gibt, dann kann mit Kenntnis des nächsten Symbols die richtige Ableitung gewählt werden.

Fall 2: $\text{predict}(S \rightarrow AB) = \text{first}(AB) \setminus \{\epsilon\} \cup \text{Follow}(S) = \{a, b, \$\}$

Prof. Dr. U. Göhner HS Kempten 87

Algorithmus des rekursiven Abstieg bei LL(1)-Grammatiken:

Für jedes Nichtterminal A mit den Produktionen $A \rightarrow s_1 | s_2 | \dots | s_n$ erzeugt man eine Funktion $A()$, die in Abhängigkeit vom zuletzt gelesenen Token die richtige Produktion auswählt und das Verarbeiten der rechten Seite übernimmt.

Forstabelle aufbauen mit predict-Mengen;
Übernahme $S \rightarrow AB$ für alle $a \in \text{predict}(S \rightarrow AB)$
 $A \rightarrow aA$ " $a \in$ " $(A \rightarrow aB)$

	a	b	\$
S	$S \rightarrow AB$	$S \rightarrow A\cancel{B}$	$S \rightarrow A\cancel{B}$
A	$A \rightarrow aA$	$A \rightarrow \epsilon$	$A \rightarrow \epsilon$
B	error	$\cancel{B} \rightarrow bB$	$\cancel{B} \rightarrow \epsilon$

Prof. Dr. U. Göhner HS Kempten 88