Syntaxanalyse

Konstruktion der Parsertabelle

- Die Parsertabelle TAB enthält für jedes Nonterminalsymbol A eine Zeile und für jedes Terminalsymbol x eine Spalte.
- Ein Eintrag TAB(A,x) enthält die Regel(n), die für A angewendet werden, falls x das Vorausschau-Token ist.

Algorithmus zur Konstruktion der Parsertabelle

Für jede Ableitungsregel A -> α :

1. Für jedes Terminalsymbol a ∈ FIRST(α) übernehme die Regel A -> α in TAB[A, a]
2. Falls ε ∈ FIRST(α), übernehme
   Regel A -> α in TAB[A, b] für jedes b ∈ FOLLOW(A)
   Falls ε ∈ FIRST(α) und $ ∈ FOLLOW(A), übernehme
   Regel A -> α in TAB[A, $].

Syntaxanalyse

Definition:

Eine kontextfreie Grammatik G heißt **LL(1)-Grammatik**, wenn gilt:

a) Gibt es zwei Produktionen N →s1 und N → s2, dann ist first(s1)∩ first(s2) = ∅.

b) Gibt es ein Nichtterminal N, mit N → ε , dann gilt first(N) ∩ follow(N) = ∅.

L L (1)

Linksableitung, Eingabe von Links gelesen, 1 Zeichen vorauslesen um zu entscheiden, welche Produktion zu verwenden ist

2. Beispiel Parsertabelle erstellen:

$S \to ABC$

$A \to aA \mid \varepsilon$

$B \to bB \mid \varepsilon$

$C \to cC \mid c$

	a	b	c	$
S	S->ABC	S->ABC	S->ABC	error
A	A->aA	A->epsilon	A->epsilon	error
B	error	B->bB	B->epsilon	error
C	error	error	C->cC, C->c	error

$S \to ABC$: $First(ABC) = \{a, b, c\}$

$A \to aA$ $First(aA) = \{a\}$

$A \to \varepsilon$: $Follow(A) = First(BC) = \{b, c\}$

$B \to bB$: $First(bB) = \{b\}$

$B \to \varepsilon$: $Follow(B) = First(C) = \{c\}$

$C \to cC$: $First(cC) = \{c\}$ $\Big\} = \{c\} \neq \emptyset$

$C \to c$: $First(c) = \{c\}$ $\Big\}$

⇒ keine LL(1)-Grammatik

Syntaxanalyse

Definition:
Eine kontextfreie Grammatik G heißt **LL(1)-Grammatik**, wenn gilt:
a) Gibt es zwei Produktionen N →s1 und N → s2, dann ist first(s1)∩ first(s2) = ∅.
b) Gibt es ein Nichtterminal N, mit N → ε , dann gilt first(N) ∩ follow(N) = ∅.
L L (1)
Linksableitung, Eingabe von Links gelesen, 1 Zeichen vorrauslesen um zu entscheiden, welche Produktion zu verwenden ist

Wie könnte man einen Top-down Parser schreiben für diese Sprache?
– 2 Vorausschausymbole verwenden (geht hier, aber nicht immer)
Stattdessen neue Grammatik, die dieselbe Sprache erzeugt:

$S \to ABC$
$A \to aA \mid \varepsilon$
$B \to bB \mid \varepsilon$
$C \to cD$
$D \to cD \mid \varepsilon$

LL(1) -Eigenschaft prüfen:
$D \to cD$ First(D) = {c, ε}
$D \to \varepsilon$ Follow(D) = Follow(C) = Follow(S) = {$} ∩ = ∅

	a	b	c	$
S	S->ABC	S->ABC	S->ABC	error
A	A->aA	A->epsilon	A->epsilon	error
B	error	B->bB	B->epsilon	error
C	error	error	C->cD	error
D	error	error	D->cD	D->epsilon

eindeutige Regelauswahl

Ableitungsbaum für $\omega = abccc\$$