

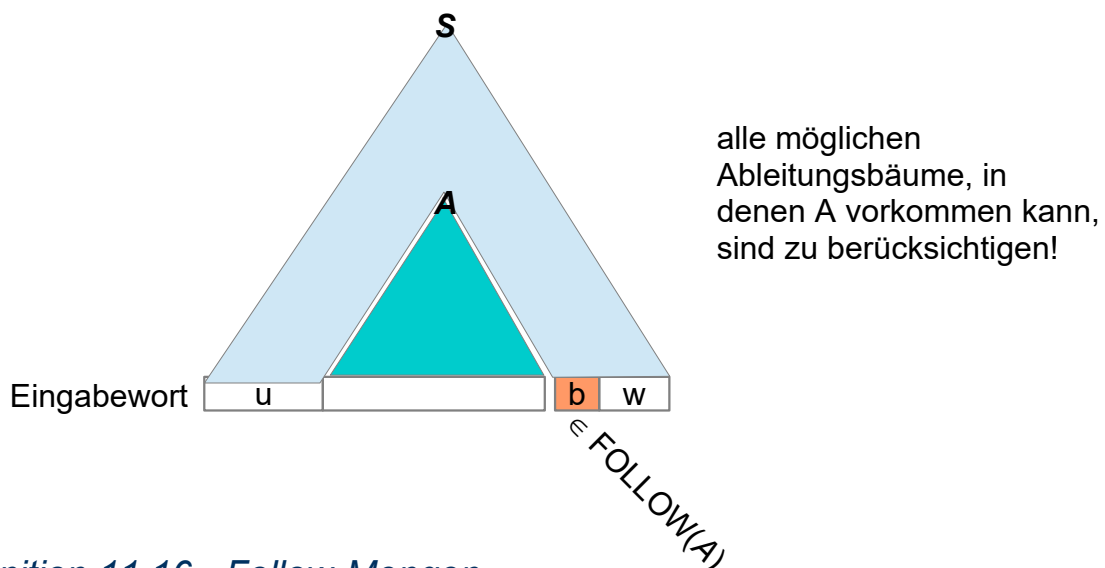
# Lektion 19

Nach der nullable-Eigenschaft und den First-Mengen wird nun mit den Follow-Mengen eine weitere Grammatikeigenschaft betrachtet. Ausgehend von diesen Eigenschaften kann die Vorausschautabelle bestimmt werden, die für grundlegende Information für die effiziente Top-Down-Analyse angibt. Wenn die Grammatik die sog. LL(1)-Eigenschaft hat, d.h. wenn die Vorausschautabelle immer eindeutig eine Produktion zum expandieren angibt, ist eine effiziente Top-Down-Syntaxanalyse damit möglich. Sie lernen zwei Ansätze kennen, wie ausgehend vom Inhalt der Vorausschautabelle die Top-Down-Syntaxanalyse implementiert werden kann.

## 11.2.3 Follow-Mengen

Für die nichtterminalen Symbole  $A$  einer kontextfreien Grammatik ist **Follow( $A$ )** die Menge aller terminalen Symbole, die in Ableitungen unmittelbar auf den Teil folgen können, der aus  $A$  abgeleitet werden kann:

### Veranschaulichung für Follow( $A$ )



### Definition 11.16 - Follow-Mengen

Gegeben sei eine kontextfreie Grammatik  $G = (N, T, P, S)$ . Für die nichtterminalen Symbole  $A \in N$  sind die Follow-Mengen folgendermaßen definiert:

$$\text{Follow}(A) = \{ b \in T \mid \text{es existieren Symbolfolgen } u \in (N \cup T)^*, w \in (N \cup T)^*, \\ \text{so dass } S \Rightarrow^* uAbw \}$$

## Beispiel 11.17 - Follow-Mengen

Wir betrachten folgende Grammatik:

$$S \rightarrow bAc$$

$$A \rightarrow aB \mid bAB$$

$$B \rightarrow bb \mid cc$$

Was lässt sich über  $\text{Follow}(A)$  aussagen?

- ▶ Betrachten wir die rechte Seite der Produktion

$$S \rightarrow b\underline{A}c,$$

dann sehen wir, dass auf ein  $A$  ein  $c$  folgen kann, d.h.  $c \in \text{Follow}(A)$

- ▶ Betrachten wir die rechte Seite der Produktion

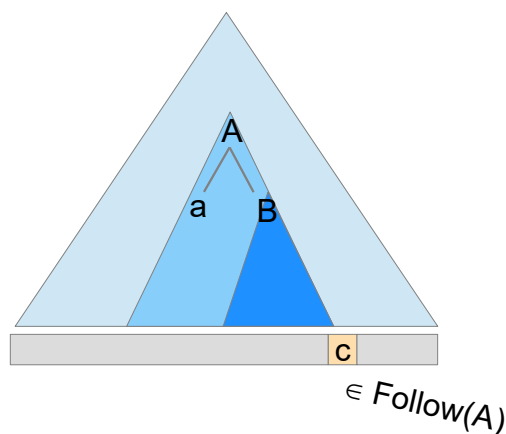
$$A \rightarrow b\underline{A}B,$$

dann ist zu sehen, dass auf  $A$  alles folgen kann, womit  $B$  beginnen kann. D.h. alle Elemente der Menge  $\text{First}(B) = \{b, c\}$  gehören auch zu  $\text{Follow}(A)$  dazu.

- ▶ Was gehört zur Follow-Menge von  $B$ ? Nichtterminal  $B$  wird beispielsweise in der Produktion

$$A \rightarrow a\underline{B}$$

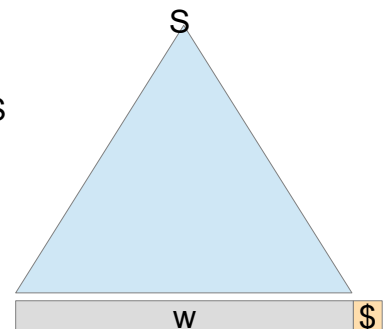
verwendet. In Ableitungsbäumen ergibt sich folgende Konstellation, da  $B$  am Ende der rechten Seite steht:



Wissen wir schon, dass  $c$  auf  $A$  folgen, dann kann auch  $c$  auf  $B$  folgen.

Entsprechend gilt das auch für andere Zeichen aus  $\text{Follow}(A)$ . D.h. da  $B$  am Ende der Produktion steht, sind alle Elemente von  $\text{Follow}(A)$  auch in die Menge  $\text{Follow}(B)$  aufzunehmen.

- ▶ Aus dem Startsymbol  $S$  wird das komplette Eingabewort  $w$  abgeleitet. Nach dem Teil, der aus  $S$  abgeleitet wird, ist das Ende des Eingabeworts erreicht. Um auch den Fall "Eingabeende erreicht" einheitlich behandeln zu können, wird hier das Zeichen  $\$$  als Kennzeichnung für das Eingabeende verwendet.



## Vorbetrachtung - Bestimmung der Follow-Mengen

An diesem Beispiel sind schon die wesentlichen Ansatzpunkte für die Berechnung von Follow-Mengen zu erkennen:

- (1) Das Eingabeende \$ gehört immer zur Follow-Menge des Startsymbols.
- (2) Kommt ein nichtterminales Symbol  $X$  innerhalb der rechten Seite einer Produktion  $A \rightarrow uXv$  vor ("Mittelpositionen"), dann gehört alles, womit der folgende Teil  $v$  beginnen kann, d.h.  $\text{First}(v)$ , zur Follow-Menge von  $X$  dazu.
- (3) Kommt ein nichtterminales Symbol  $X$  am Ende einer Produktion  $A \rightarrow uX$  vor, dann gehören alle Elemente der Follow-Menge der linken Seite  $A$  auch zur Follow-Menge von  $X$  dazu. Sofern auf  $X$  in der Produktion noch etwas folgt, was  $\varepsilon$  ergeben kann, d.h. das nullable ist, dann ist dieses  $X$  auch so zu behandeln, als ob es am Ende der rechten Seite wäre.

## Algorithmus 11.18 - Berechnung der Follow-Mengen

Gegeben sei eine kontextfreie Grammatik  $G = (N, T, P, S)$ . Die Follow-Mengen für die nichtterminalen Symbole  $A \in N$  werden folgendermaßen berechnet:

### (1) Initialisierung:

$\text{Follow}(S) = \{ \$ \}$  für Startsymbol  $S$  (\$ kennzeichnet Eingabeende)

$\text{Follow}(A) = \{ \}$  für alle anderen nichtterminalen Symbole  $A$

### (2) Phase 1, "Mittelpositionen behandeln":

Gehe alle Produktionen durch. Kommt ein Nichtterminal  $X$  in der rechten Seite einer Produktion vor, d.h. hat eine Produktion die Form

$$A \rightarrow uXv \quad \text{wobei } u \in (N \cup T)^*, v \in (N \cup T)^+$$

dann nimm **First(v)** zu **Follow(X)** dazu.

### (3) Phase 2, "Endposition behandeln":

**Wiederhole**

**Gehe alle Produktionen durch:**

Wenn die Produktion die Form

$$A \rightarrow uX$$

hat oder

$$A \rightarrow uXv, \text{ wobei } \text{nullable}(v) \text{ gilt,}$$

dann

nimm alle Symbole von **Follow(A)** zu **Follow(X)** dazu

**solange bis** sich keine Follow-Menge mehr ändert.

## Anmerkung

- ▶ In Phase 1 muss jede Produktion und jede Position innerhalb der rechten Seite der jeweiligen Produktion nur einmal betrachtet werden.
- ▶ In Phase 2 sind ggf. mehrere Iterationen durch alle Produktionen nötig, bis ein stabiles Ergebnis vorliegt.

## Beispiel 11.19 - Follow-Mengen berechnen

Folgende Grammatik, wie in den vorangehenden Beispielen, sei gegeben.

$S \rightarrow d \mid ABS$

$A \rightarrow BC \mid a$

$B \rightarrow \varepsilon \mid bCe$

$C \rightarrow \varepsilon \mid c$

- ▶ **Initialisierung:** nimm \$ zu Follow(S) dazu

	<i>nullable</i>	<i>First</i>	<i>Follow</i>
S	-	a, b, c, d	\$
A	ja	a, b, c	
B	ja	b	
C	ja	c	

- ▶ **Phase 1 "Mittelpositionen":** Betrachte Vorkommen von Nichtterminalen innerhalb der rechten Seite einer Produktion (betrachtete Stelle unterstrichen):

$S \rightarrow \underline{A}BS$  : nimm  $\text{First}(BS) = \{a,b,c,d\}$  zu  $\text{Follow}(A)$  dazu <sup>(1)</sup>

$S \rightarrow A\underline{B}S$  : nimm  $\text{First}(S) = \{a,b,c,d\}$  zu  $\text{Follow}(B)$  dazu <sup>(2)</sup>

$A \rightarrow \underline{B}C$  : nimm  $\text{First}(C) = \{c\}$  zu  $\text{Follow}(B)$  dazu (keine Änderung)

$B \rightarrow b\underline{C}e$  : nimm  $\{e\}$  zu  $\text{Follow}(C)$  dazu <sup>(3)</sup>

	<i>nullable</i>	<i>First</i>	<i>Follow</i>
S	-	a, b, c, d	\$
A	ja	a, b, c	a,b,c,d <sup>(1)</sup>
B	ja	b	a,b,c,d <sup>(2)</sup>
C	ja	c	e <sup>(3)</sup>

- ▶ **Phase 2 (Endpositionen)**

1. Durchgang:

$S \rightarrow AB\underline{S}$  : nimm  $\text{Follow}(S)$  zu  $\text{Follow}(S)$  dazu (keine Änderung)

$A \rightarrow B\underline{C}$  : nimm  $\text{Follow}(A) = \{a,b,c,d\}$  zu  $\text{Follow}(C)$  dazu <sup>(4)</sup>

$A \rightarrow \underline{B}C$  : nullable(C), nimm  $\text{Follow}(A) = \{a,b,c,d\}$  zu  $\text{Follow}(B)$  dazu, keine Änderung

	<i>nullable</i>	<i>First</i>	<i>Follow</i>
S	-	a, b, c, d	\$
A	ja	a, b, c	a,b,c,d <sup>(1)</sup>
B	ja	b	a,b,c,d <sup>(2)</sup>
C	ja	c	e <sup>(3)</sup> , a,b,c,d <sup>(4)</sup>

2. Durchgang:

$S \rightarrow AB\underline{S}$  nimm Follow(S) zu Follow(S) dazu (keine Änderung)

$A \rightarrow B\underline{C}$  : nimm Follow(A) zu Follow(C) dazu (keine Änderung)

$A \rightarrow \underline{B}C$  : nullable (C), nimm Follow(A) zu Follow(B) dazu (keine Änd.)

Die Berechnung ist fertig, da es keine Änderung im 2. Durchgang gibt.

Das folgende Beispiel zeigt, dass das Symbol \$ für das Eingabeende-Symbol nicht nur beim Startsymbol, sondern auch bei anderen Nichtterminalen in der Follow-Menge enthalten sein kann. Es zeigt außerdem, dass zur Follow-Menge des Startsymbols auch andere Symbole außer \$ gehören können.

### Beispiel 11.20 - Follow-Mengen berechnen

$S \rightarrow bSb \mid aA$

$A \rightarrow bB \mid aAa$

$B \rightarrow aB \mid \varepsilon$

► Die Eigenschaften nullable und First sind direkt zu sehen (siehe Tabelle unten)

► **Initialisierung:** nimm \$ zu Follow(S) dazu

	<i>nullable</i>	<i>First</i>	<i>Follow</i>
S	nein	a,b	\$
A	nein	a,b	
B	ja	a	

► **Phase 1 (Mittelpositionen):**

$S \rightarrow b\underline{S}b$  nimm b zu Follow(S) dazu <sup>(1)</sup>

$A \rightarrow a\underline{A}a$  nimm a zu Follow(A) dazu <sup>(2)</sup>

	<i>nullable</i>	<i>First</i>	<i>Follow</i>
S	nein	a,b	\$, b <sup>(1)</sup>
A	nein	a,b	a <sup>(2)</sup>
B	ja	a	

## ► Phase 2 (Endpositionen):

### 1. Durchgang:

$S \rightarrow a\mathbf{A}$  nimm  $\text{Follow}(S) = \{\$, b\}$  zu  $\text{Follow}(A)$  dazu <sup>(3)</sup>  
 $A \rightarrow b\mathbf{B}$  nimm  $\text{Follow}(A) = \{\$, a, b\}$  zu  $\text{Follow}(B)$  dazu <sup>(4)</sup>  
 $B \rightarrow a\mathbf{B}$  nimm  $\text{Follow}(B)$  zu  $\text{Follow}(B)$  dazu, keine Änderung

	<i>nullable</i>	<i>First</i>	<i>Follow</i>
S	nein	a,b	\$, b <sup>(1)</sup>
A	nein	a,b	a <sup>(2)</sup> , \$, b <sup>(3)</sup>
B	ja	a	\$, a, b <sup>(4)</sup>

### 2. Durchgang:

$S \rightarrow a\mathbf{A}$  nimm  $\text{Follow}(S) = \{\$, b\}$  zu  $\text{Follow}(A)$  dazu, keine Änderung  
 $A \rightarrow b\mathbf{B}$  nimm  $\text{Follow}(A) = \{\$, a, b\}$  zu  $\text{Follow}(B)$  dazu, keine Änderung  
 $B \rightarrow a\mathbf{B}$  nimm  $\text{Follow}(B)$  zu  $\text{Follow}(B)$  dazu, keine Änderung

fertig!

## Aufgabe 11.21 - Follow-Mengen berechnen

Berechnen Sie die Follow-Mengen für folgende Grammatik (siehe auch Aufgabe 11.15) :

$S \rightarrow AS \mid d$   
 $A \rightarrow Bc \mid aA$   
 $B \rightarrow \varepsilon \mid b$

## 11.2.4 Bestimmung der Vorausschau-Tabelle

Die Vorausschautabelle enthält die essentiellen Informationen für die effiziente Top-Down-Syntaxanalyse: wann ist welche Produktion für expand-Schritte zu wählen?  
Der Inhalt der Vorausschautabelle kann mit den im vorigen Abschnitt betrachteten Grammatik-Eigenschaften, d.h. der nullable-Eigenschaft und den First- sowie Follow-Mengen, nun berechnet werden.

### Definition 11.22 - Vorausschautabelle für die Top-Down-Analyse

Für eine kontextfreie Grammatik  $G = (N, T, P, S)$  mit den terminalen Zeichen  $T = \{a_1, \dots, a_n\}$  und den nichtterminalen Zeichen  $N = \{A_1, \dots, A_m\}$  ist die Vorausschautabelle folgendermaßen aufgebaut:

- Für jedes nichtterminale Symbol  $A_i$  der Grammatik gibt es eine Zeile.
- Für jedes terminale Symbol  $a_k$  gibt es eine Spalte. Außerdem gibt es eine Spalte für das Eingabeendesymbol \$.

Eingabezeichen	$a_1$	$a_2$	$a_3$	...	$a_n$	\$
Nichtterminale						
$A_1$				...		
$A_2$				...		
...	...	...	...	...	...	
$A_m$				...		

- Eine Produktion  $A \rightarrow w$  wird in der Zeile für das Nichtterminal  $A$  und der Spalte für das Symbol  $x$  (terminales Zeichen oder Eingabeende  $\$$ ) eingetragen
- (1) falls  $x \in \text{First}(w)$
  - (2) falls  $w = \varepsilon$  oder  $\text{nullable}(w)$   
und  $x \in \text{Follow}(A)$

### Beispiel 11.23 - Bestimmung der Vorausschautabelle

Berechnung der Vorausschautabelle für die Grammatik aus den vorangehenden Beispielen:

$S \rightarrow d \mid ABS$

$A \rightarrow BC \mid a$

$B \rightarrow \varepsilon \mid bCe$

$C \rightarrow \varepsilon \mid c$

Die Grammatikeigenschaften wurden schon berechnet:

	<i>nullable</i>	<i>First</i>	<i>Follow</i>
S	-	a, b, c, d	\$
A	ja	a, b, c	a,b,c,d
B	ja	b	a,b,c,d
C	ja	c	a,b,c,d, e

Erstellte Vorausschautabelle:

<i>Eingabe Nichtterminal</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>\$</i>
S	$S \rightarrow ABS^{(2)}$	$S \rightarrow ABS^{(2)}$	$S \rightarrow ABS^{(2)}$	$S \rightarrow d^{(1)}$ $S \rightarrow ABS^{(2)}$	-	-
A	$A \rightarrow BC^{(4)}$ $A \rightarrow a^{(5)}$	$A \rightarrow BC^{(3)}$	$A \rightarrow BC^{(3)}$	$A \rightarrow BC^{(4)}$	-	-
B	$B \rightarrow \varepsilon^{(6)}$	$B \rightarrow \varepsilon^{(6)}$ $B \rightarrow bCe^{(7)}$	$B \rightarrow \varepsilon^{(6)}$	$B \rightarrow \varepsilon^{(6)}$	-	-
C	$C \rightarrow \varepsilon^{(8)}$	$C \rightarrow \varepsilon^{(8)}$	$C \rightarrow \varepsilon^{(8)}$ $C \rightarrow c^{(9)}$	$C \rightarrow \varepsilon^{(8)}$	$C \rightarrow \varepsilon^{(8)}$	-

Erläuterung zum Eintragen der Produktionen in die Vorausschautabelle:

- $S \rightarrow d$        $\text{First}(d) = \{d\}$ , nicht nullable  
 $\Rightarrow$  nur in Spalte d eintragen <sup>(1)</sup>
- $S \rightarrow ABS$        $\text{First}(ABS) = \{a,b,c,d\}$ , nicht nullable  
 $\Rightarrow$  bei a, b, c, d eintragen <sup>(2)</sup>
- $A \rightarrow BC$        $\text{First}(BC) = \{b, c\}$ ,  
 $\Rightarrow$  bei b und c eintragen <sup>(3)</sup>  
 $\text{nullable}(BC)$  gilt,  $\text{Follow}(A) = \{a,b,c,d\}$   
 $\Rightarrow$  bei a, b, c, d eintragen <sup>(4)</sup>
- $A \rightarrow a$        $\text{First}(a) = \{a\}$   
 $\Rightarrow$  bei a eintragen <sup>(5)</sup>
- $B \rightarrow \varepsilon$        $\text{Follow}(B) = \{a,b,c,d\}$   
 $\Rightarrow$  bei a, b, c, d eintragen <sup>(6)</sup>
- $B \rightarrow bCe$        $\text{First}(bCe) = \{b\}$   
 $\Rightarrow$  bei b eintragen <sup>(7)</sup>
- $C \rightarrow \varepsilon$        $\text{Follow}(C) = \{a,b,c,d, e\}$   
 $\Rightarrow$  bei a, b, c, d, e eintragen <sup>(8)</sup>
- $C \rightarrow c$        $\text{First}(c) = \{c\}$   
 $\Rightarrow$  bei c eintragen <sup>(9)</sup>

An diesem Beispiel sind folgende typische Fälle zu sehen:

- ▶ Beginnt die rechte Seite einer Produktion mit einem terminalen Symbol, z.B. wie bei  $B \rightarrow \underline{b}Ce$ , dann wird diese Produktion nur in der Spalte dieses Symbols eingetragen, da alles, was damit abgeleitet werden kann, nur mit diesem Symbol beginnen kann.
- ▶ Bei  $\varepsilon$ -Produktionen, z.B.  $B \rightarrow \varepsilon$ , wird die Produktion bei allen Zeichen der Follow-Menge  $\text{Follow}(B)$  eingetragen, da dann das Eingabezeichen der Vorausschau das ist, was auf B folgt.



- ▶ Ist die rechte Seite einer Produktion nicht  $\varepsilon$ , aber insgesamt nullable, z.B. wie bei  $A \rightarrow BC$ , sind beide Fälle zu berücksichtigen:
  - (1) Die Produktion  $A \rightarrow BC$  ist bei allen Zeichen aus  $\text{First}(BC)$  einzutragen.
  - (2) Da  $BC$  nullable ist, ist die Produktion auch wie eine  $\varepsilon$ -Produktion zu behandeln, d.h. die Produktion  $A \rightarrow BC$  ist auch bei allen Zeichen aus  $\text{Follow}(A)$  einzutragen.

### Aufgabe 11.24 - Bestimmung der Vorausschautabelle

- ▶ Berechnen Sie die Grammatikeigenschaften (nullable, First, Follow) und bestimmen Sie dann die Vorausschautabelle für folgende Grammatik:

$$\begin{array}{lcl} S \rightarrow & AS & | d \\ A \rightarrow & Bc & | aA \\ B \rightarrow & \varepsilon & | b \end{array}$$

### Beispiel 11.25 - Vorausschau-Tabelle

Die Vorausschautabelle für folgende Grammatik ist zu bestimmen.

$$\begin{array}{lcl} S \rightarrow & AB & \\ A \rightarrow & aA & | \varepsilon \\ B \rightarrow & bB & | \varepsilon \end{array}$$

- ▶ Grammatikeigenschaften bestimmen:

	<i>nullable</i>	<i>First</i>	<i>Follow</i>
S	ja	a,b	\$
A	ja	a	b, \$
B	ja	b	\$

Anmerkung: An dieser Grammatik kann man sich auch nochmal die Bedeutung der Follow-Mengen klar machen. Die Grammatik definiert die Sprache  $L(G) = \{a^n b^k \mid n, k \geq 0\}$ . A erzeugt beliebig viele a, B beliebig viele b. Auf einen aus A erzeugten Teil kann somit ein b folgen, aber auch das Eingabeende \$, wenn der B-Teil leer ist.

- ▶ Vorausschau-Tabelle füllen:

	a	b	\$
S	$S \rightarrow AB^{(1)}$	$S \rightarrow AB^{(1)}$	$S \rightarrow AB^{(2)}$
A	$A \rightarrow aA^{(3)}$	$A \rightarrow \varepsilon^{(4)}$	$A \rightarrow \varepsilon^{(4)}$
B	—	$B \rightarrow bB^{(5)}$	$B \rightarrow \varepsilon^{(6)}$

$S \rightarrow AB$ :  $\text{First}(AB) = \{a,b\} \Rightarrow$  bei a und b eintragen<sup>(1)</sup>  
 $\text{nullable}(AB) = \text{true}$ ,  $\text{Follow}(S) = \{\$ \} \Rightarrow$  bei \$ eintragen<sup>(2)</sup>

$A \rightarrow aA:$	$\text{First}(aA) = \{a\}$	$\Rightarrow$ bei a eintragen <sup>(3)</sup>
$A \rightarrow \varepsilon:$	$\text{Follow}(A) = \{b, \$\}$	$\Rightarrow$ bei b und \$ eintragen <sup>(4)</sup>
$B \rightarrow bA:$	$\text{First}(bB) = \{b\}$	$\Rightarrow$ bei b eintragen <sup>(5)</sup>
$B \rightarrow \varepsilon:$	$\text{Follow}(B) = \{\$\}$	$\Rightarrow$ bei \$ eintragen <sup>(6)</sup>

An diesem Beispiel ist zu sehen, dass auch in der Spalte für das Eingabeende \$ Produktionen enthalten sein können.

## 11.3 LL(1)-Grammatiken

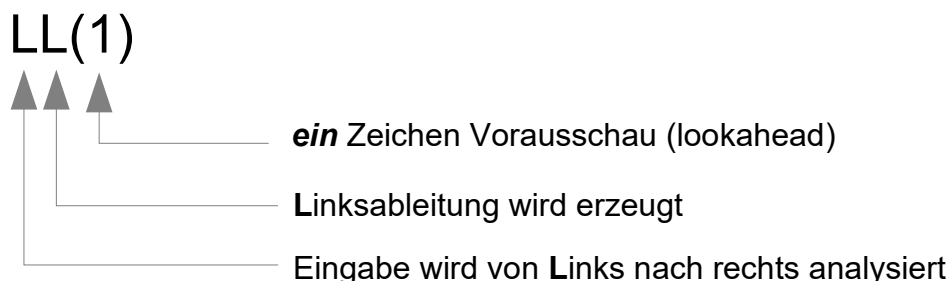
In obiger Aufgabe 11.24 - Bestimmung der Vorausschautabelle ist es so, dass an jeder Position in der Vorausschautabelle maximal eine Produktion eingetragen ist. Bei anderen Grammatik, z.B. in Beispiel 11.23, kommen auch Mehrfacheinträge vor, d.h. für eine Kombination Nichtterminal/Eingabezeichen gibt es mehrere mögliche Produktionen.

### Definition 11.26 - LL(1)-Grammatiken

Grammatiken, deren Vorausschautabelle keine Mehrfacheinträge enthalten, heißen **LL(1)-Grammatiken**.

Hat eine Grammatik die LL(1)-Eigenschaft, ist eine effiziente Top-Down-Syntaxanalyse mit einem Zeichen Vorausschau möglich.

### Namensgebung



### LL(k)-Grammatiken

Analog können Vorausschautabellen für eine längere Vorausschau (*lookahead*) von  $k$  Zeichen ( $k > 1$ ) definiert werden.

Grammatiken, die bei einer Vorausschau von  $k$  Zeichen keine Mehrfacheinträge in der Vorausschautabelle haben, heißen dann **LL(k)-Grammatiken**.

### Anmerkung - LR(k)-Grammatiken

Es gibt neben der Top-Down-Syntaxanalyse auch Verfahren zur Bottom-Up-Analyse. Bei der Bottom-Up-Syntaxanalyse ergeben sich Rechtsableitungen. Entsprechend

dem oben dargestellten Bezeichnungsschema werden diese Verfahren folglich mit LR(1) bzw. LR(k) bezeichnet.

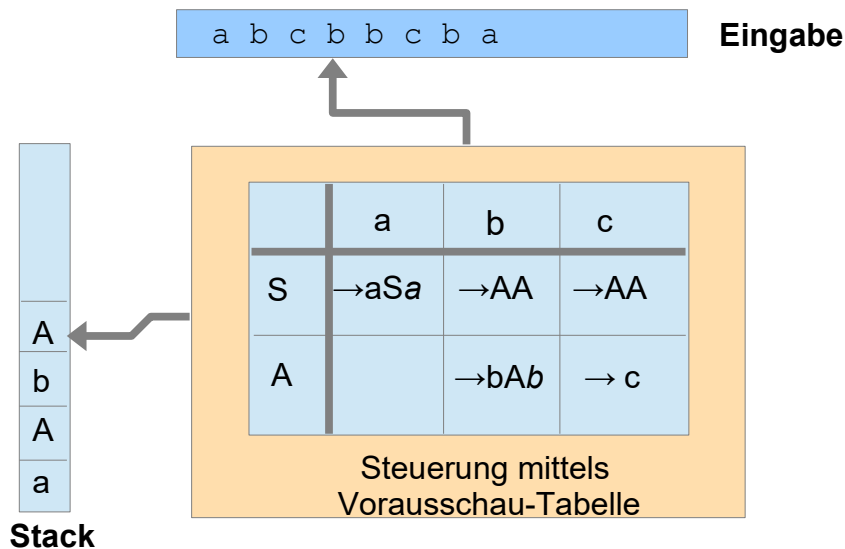
## 11.4 Implementierung effizienter Top-Down-Analyse

Mit Hilfe der in der Vorausschautabelle enthaltenen Information, wann welche Produktion für Expand-Schritte zu verwenden ist, kann die Top-Down-Syntaxanalyse einfach implementiert werden. Es werden hierfür zwei Implementierungsansätze kurz vorgestellt:

- (1) **Tabellengesteuerter** Top-Down-Parser
- (2) Top-Down-Parser mit **rekursivem Abstieg**

### 11.4.1 Tabellengesteuerter Top-Down-Parser mit Vorausschau

#### Aufbau des tabellengesteuerten Parsers



Ein tabellengesteuerter Top-Down-Parser ist im Kern ein Kellerautomat mit Vorausschaumöglichkeit zur Auswahl der Produktion bei den Expand-Schritten.

#### Steuerung der Parsers

- ▶ **Startzustand:** Startsymbol S liegt auf dem Stack, Eingabeposition ist erstes Zeichen.
- ▶ Teilschritte bei der Analyse:

<b>oberstes Stackelement</b>	<b>nächstes Eingabesymbol</b>	<b>Verhalten</b>
Terminal $a$	gleiches Terminal $a$	$a$ vom Stack entfernen, Eingabeposition weitersetzen
Terminal $a$	anderes Terminal $b$	Syntaxfehler melden
Nichtterminal $A$	Terminal $a$	Eintrag zu $A/a$ in Vorausschautabelle suchen <ul style="list-style-type: none"> <li>kein Eintrag vorhanden: Syntaxfehler melden</li> <li>sonst: <math>A</math> vom Stack entfernen, rechte Seite der Regel auf Stack ablegen (erstes Symbol oben)</li> </ul>

- **akzeptierender Endzustand:** Stack und Eingabe leer

### Anmerkung

- Der tabellengesteuerte Parser hat folgende Unterschiede zum Kellerautomat aus Definition 10.2 bzw. Konstruktion 10.14:
- Vorausschau: Parser kann nächstes Eingabesymbol inspizieren, ohne es aus der Eingabe zu entfernen.
  - Eingabe wird durch leeren Keller akzeptiert (nicht durch Endzustand)
- Das Verhalten ist **deterministisch**, sofern die Vorausschautabelle keine Mehrfacheinträge besitzt, d.h. die Grammatik die LL(1)-Eigenschaft hat.

### Beispiel 11.27 - Tabellengesteuerte Top-Down-Parser

Für diese Grammatik

$$S \rightarrow aSd \mid AA$$

$$A \rightarrow bAe \mid c$$

ergibt sich folgende Vorausschautabelle.

	<b><math>a</math></b>	<b><math>b</math></b>	<b><math>c</math></b>	<b><math>d</math></b>	<b><math>e</math></b>	<b><math>\\$</math></b>
<b><math>S</math></b>	$S \rightarrow aSd$	$S \rightarrow AA$	$S \rightarrow AA$	-	-	-
<b><math>A</math></b>	-	$A \rightarrow bAe$	$A \rightarrow c$	-	-	-

Da die Tabelle keine Mehrfacheinträge enthält, hat die Grammatik die LL(1)-Eigenschaft und eine effiziente Top-Down-Analyse ist möglich.

- Tabellengesteuerte Analyse bei Eingabewort `abcecd`:

<b>Stack</b>	<b>Eingabe</b>	<b>Anmerkung</b>
<b>S</b>	abcecd	$S/a$ : wähle $S \rightarrow aSd$
dSa	abcecd	$a/a$ : $a$ entfernen

<b>Stack</b>	<b>Eingabe</b>	<b>Anmerkung</b>
dS	bcecd	S/b: wähle $S \rightarrow AA$
dAA	bcecd	A/b: wähle $A \rightarrow bAe$
dAeAb	bcecd	b/b: b entfernen
dAeA	cecd	A/c: wähle $A \rightarrow c$
dAec	cecd	c/c: c entfernen
dAe	ecd	e/e: e entfernen
dA	cd	A/c: wähle $A \rightarrow c$
dc	cd	c/c: entferne c
d	d	d/d: entferne d
$\varepsilon$	$\varepsilon$	Eingabe o

### Aufgabe 11.28 - Tabellengesteuerte Top-Down-Parser

Vollziehen Sie nach, wie der Parser aus dem vorangehenden Beispiel 11.27 das Wort `baec` analysiert.

## 11.4.2 Top-Down-Analyse mittels rekursivem Abstieg

Eine zweite Möglichkeit, die Top-Down-Analyse zu implementieren, ist die sog. *Syntaxanalyse mit rekursivem Abstieg* (engl. *recursive descend parser*), die folgendermaßen realisiert wird:

- ▶ Für jedes nichtterminale Symbol der Grammatik wird eine Methode implementiert, die prüft, ob der folgende Teil der Eingabe entsprechend dem nichtterminalen Symbol aufgebaut ist.
- ▶ Wird bei der Analyse ein nichtterminales Symbol erwartet, wird die entsprechende Methode aufgerufen. Da kontextfreie Grammatiken rekursiv definiert sein können, ergeben sich dadurch ggf. auch rekursive Methoden.
- ▶ Es werden zwei Hilfsfunktionen verwendet:
  - `lookahead()`: Liefert Symbol an aktueller Eingabeposition (Eingabeposition bleibt aber unverändert)
  - `consume(ch)`: Vergleicht das erwartete Zeichen `ch` mit Zeichen an aktueller Eingabeposition. Bei Übereinstimmung geht die Eingabeposition weiter zum folgenden Zeichen. Andernfalls erfolgt ein Abbruch mit Syntaxfehler.

### Beispiel 11.29 - Top-Down-Analyse mittels rekursivem Abstieg

Folgende Grammatik sei gegeben:

$$\begin{aligned} S &\rightarrow AS \mid d \\ A &\rightarrow Bc \mid aA \\ B &\rightarrow \varepsilon \mid b \end{aligned}$$

In Aufgabe 11.24 wurde bereits die Vorausschautabelle dafür berechnet:

	a	b	c	d	\$
S	$S \rightarrow AS$	$S \rightarrow AS$	$S \rightarrow AS$	$S \rightarrow d$	-
A	$A \rightarrow aA$	$A \rightarrow Bc$	$A \rightarrow Bc$	-	-
B	-	$B \rightarrow b$	$B \rightarrow \varepsilon$	-	-

Für das nichtterminale Symbol S mit den Produktionen

$$\begin{aligned} S &\rightarrow AS \\ &\mid d \end{aligned}$$

ergibt sich folgende Analysemethode.

- ▶ Entsprechend Vorausschautabelle ist bei Vorausschau a, b, und c die Produktion  $S \rightarrow AS$  zu verwenden, d.h. es werden die Methoden für A und S nacheinander aufgerufen.
- ▶ Bei Vorausschau d ist die Produktion  $S \rightarrow d$  anzuwenden, d.h. es wird `consume(d)` aufgerufen.
- ▶ Kommt keines der Zeichen a, b, c oder d in der Eingabe, wird ein Syntaxfehler gemeldet.

```
void S() {  
    switch(lookahead()) {  
        case 'a':  
        case 'b':  
        case 'c':           // Produktion S→AS bei Vorausschau a,b,c  
            A(); S();       // Aufruf der Analysemethoden für A und S  
            break;  
        case 'd':           // Produktion S→d bei Vorausschau d  
            consume('d');   // Zeichen d wird in Eingabe erwartet  
            break;  
        default:            // Syntaxfehler bei unerwartetem Zeichen  
            throw SyntaxError();  
    }  
}
```

In gleicher Weise werden die Methoden für A und B umgesetzt (vollständiger Programmcode siehe Moodle).

### *Anmerkung*

- ▶ Das Verfahren des rekursiven Abstiegs ist nicht nur für kleine Beispiele geeignet, sondern kann auch für reale, komplexe Sprachen, wie z.B. Programmiersprachen, verwendet werden.
- ▶ Bei der Syntaxanalyse mit rekursivem Abstieg wird zwar nicht explizit ein Stack verwendet, aber implizit spielt auch hier ein Stack eine wichtige Rolle - der Aufrufstack für Methoden im Laufzeitsystem der Programmiersprache

## Zusammenfassung zu Lektion 19

---

### *Diese Fragen sollten Sie nun beantworten können*

- ▶ Was bedeuten die Follow-Mengen für nichtterminale Symbole einer Grammatik?
- ▶ Wie können die Follow-Mengen berechnet werden?
- ▶ Wie wird die Vorausschautabelle zu einer kontextfreien Grammatik bestimmt, ausgehend von der nullable-Eigenschaft und den First- und Follow-Mengen?
- ▶ Was bedeutet  $LL(1)$ ?
- ▶ Welche Auswirkungen hat es, wenn eine Grammatik nicht die  $LL(1)$ -Eigenschaft hat?
- ▶ Wie arbeitet ein tabellengesteuerter Top-Down-Parser?
- ▶ Wie arbeitet Top-Down-Analyse mittels rekursivem Abstieg?