

## Aufgabe 18

Gegeben sei folgende Grammatik  $G=(N,T,P,S)$ :

$N=\{S,A,B,C\}; T=\{a,b,c\};$

P: S: AbAC ;

A: Baa |  $\epsilon$  ;

B: Bb |  $\epsilon$  ;

C: c ;

### 1. Predict-Mengen

$\text{First}(B) = \{b, \epsilon\}$

$\text{First}(A) = \{a, b, \epsilon\}$

$\text{First}(C) = \{c\}$

$\text{Follow}(A) = \{b, c\}$

$\text{Follow}(B) = \{a, b\}$

$\text{Follow}(C) = \text{Follow}(S) = \{\$ \}$

$X \rightarrow s$	$s \rightarrow \epsilon?$	$\text{First}(s) \setminus \epsilon$	$\text{Follow}(X)$	$\text{Predict}(X \rightarrow s)$
$S \rightarrow AbAC$	no	a,b	-	a,b
$A \rightarrow Baa$	no	a,b	-	a,b
$A \rightarrow \epsilon$	yes	-	$\text{Follow}(A) = \{b\} \cup \{c\}$	b,c
$B \rightarrow Bb$	no	b	-	b
$B \rightarrow \epsilon$	yes	-	$\text{Follow}(B) = \{a\} \cup \{b\}$	a,b
$C \rightarrow c$	no	c	-	c

### 2. Parsertabelle:

	a	b	c	\$
S	$S \rightarrow AbAC$	$S \rightarrow AbAC$		
A	$A \rightarrow Baa$	$A \rightarrow Baa$ $A \rightarrow \epsilon$	$A \rightarrow \epsilon$	
B	$B \rightarrow \epsilon$	$B \rightarrow Bb$ $B \rightarrow \epsilon$		
C			$C \rightarrow c$	

LL(1)-Eigenschaft nicht erfüllt, da in Zeile A Spalte b 2 Produktionen und Zeile A / Spalte b 2 Produktionen sind, ist Regelauswahl nicht eindeutig!

## Aufgabe 19

Gegeben sei folgende Grammatik  $G=(N,T,P,S)$ :

$N=\{S,A,B,C\}; T=\{a,b,c\};$

$P:$       $S: AbCB;$

$A: Ba \mid \varepsilon;$

$B: BACA \mid \varepsilon;$

$C: c;$

### 1. Predict-Mengen

$\text{First}(A) = \{a, c, \varepsilon\}$

$\text{First}(B) = \{a, c, \varepsilon\}$

$\text{First}(C) = \{c\}$

$\text{Follow}(B) = \{a, c\} \cup \text{Follow}(S) = \{a, c, \$\}$

$\text{Follow}(A) = \{b\} \cup \{c\} \cup \text{Follow}(B) = \{a, b, c, \$\}$

$\text{Follow}(C) = \text{First}(A) \setminus \{\varepsilon\} \cup \text{First}(B) \setminus \{\varepsilon\} \cup \text{Follow}(S) \cup \text{Follow}(B)$   
 $= \{a, c\} \cup \{a, c\} \cup \{\$ \} \cup \{a, c, \$\} = \{a, c, \$\}$

$X \rightarrow s$	$s \rightarrow \varepsilon?$	$\text{First}(s) \setminus \varepsilon$	$\text{Follow}(X)$	$\text{Predict}(X \rightarrow s)$
$S \rightarrow AbCB$	no	a,c,b	-	a,b,c
$A \rightarrow Ba$	no	a,c	-	a,c
$A \rightarrow \varepsilon$	yes	-	$\text{Follow}(A) = \{b\} \cup \{c\}$ $\cup \text{Follow}(B) = \{a, b, c, \$\}$	a,b,c,\$
$B \rightarrow BACA$	no	a,c	-	a,c
$B \rightarrow \varepsilon$	yes	-	$\text{Follow}(B) = \{a, c\}$ $\cup \text{Follow}(S) = \{a, c, \$\}$	a,c,\$
$C \rightarrow c$	no	c	-	c

	a	b	c	\$
S	$S \rightarrow AbCB$	$S \rightarrow AbCB$	$S \rightarrow AbCB$	
A	$A \rightarrow Ba$ $A \rightarrow \varepsilon$	$A \rightarrow \varepsilon$	$A \rightarrow Ba$ $A \rightarrow \varepsilon$	$A \rightarrow \varepsilon$
B	$B \rightarrow BACA$ $B \rightarrow \varepsilon$		$B \rightarrow BACA$ $B \rightarrow \varepsilon$	$B \rightarrow \varepsilon$
C			$C \rightarrow c$	

LL(1)-Eigenschaft nicht erfüllt, da in Zeile A Spalte a und c 2 Produktionen und Zeile B / Spalte a und c 2 Produktionen sind, ist Regelauswahl nicht eindeutig!

## Aufgabe 20

Gegeben sei folgende Grammatik:  $G=(N,T,P,S)$ :

$N=\{S,A,B,C\}; T=\{a,b,c\};$

$S: ABbC ;$

$A: bbA \mid bbB ;$

$B: a \mid acC ;$

$C: Cc \mid \varepsilon ;$

- Führen Sie geeignete Linksfaktorisierungen durch und beseitigen Sie Linksrekursionen
- Bestimmen Sie alle predict-Mengen für die neue Grammatik
- Prüfen Sie, ob die Grammatik die LL(1)-Eigenschaft erfüllt.

a)

**Linksfaktorisierung:**

$A:bbA \mid bbB$

Wird ersetzt durch:

$A:bbD$

$D:A \mid B$

$B: a \mid acC$

Wird ersetzt durch

$B: aE$

$E: \varepsilon \mid cC$

**Beseitigung Linksrekursion:**

$C:Cc \mid \varepsilon$

Wird ersetzt durch:

$C:F$

$F:cF \mid \varepsilon$

neue Grammatik:

$S:ABbC$

$A:bbD$

$D:A \mid B$

$B: aE$

$E: \varepsilon \mid cC$

$C:F$

$F:cF \mid \varepsilon$

b)

neue Grammatik:

S:ABbC

A:bbD

D:A|B

B: aE

E:  $\epsilon$  | cC

C:F

F:cF |  $\epsilon$

X→s	s→ $\epsilon$ ?	First(s)\ $\epsilon$	Follow(X)	Predict(X→s)
S→ABbC	no	b	-	B
A→bbD	no	b	-	B
D→A	no	b	-	B
D→B	no	a	-	A
B→aE	no	a	-	A
E→ $\epsilon$	yes	-	Follow(E) =Follow(B) ={b}∪Follow(D) ={b}∪Follow(A) ={b}∪First(B)∪Follow(D) ={b,a}	a,b
E→cC	no	c	-	C
C→F	yes	c	Follow(C) =Follow(S)∪Follow(E) ={\$}∪Follow(B) ={a,b,\$}	a,b,c,\$
F→cF	no	c	-	C
F→ $\epsilon$	yes	-	Follow(F)=Follow(C) ={a,b,\$}	a,b,\$

c)

predict(D→A)={b} und predict(D→B)={a} ist

predict(D→A)∩predict(D→B)={}

predict(E→cC)={c} und predict(E→ $\epsilon$ )={a,b} ist

predict(E→cC)∩predict(E→ $\epsilon$ )={}


predict(F→cF)={c} und predict(F→ $\epsilon$ )={a,b,\$} ist

predict(F→cF)∩predict(F→ $\epsilon$ )={}

also ist für die neue Grammatik LL(1) erfüllt.

d) Erstellen Sie die Parser-Tabelle für die neue Grammatik:

	A	b	c	\$
S		S->ABbC		
A		A->bbD		
B	B->aE			
C	C->F	C->F	C->F	C->F
D	D->B	D->A		
E	E-> $\epsilon$	E-> $\epsilon$	E->cC	
F	F-> $\epsilon$	F-> $\epsilon$	F->cF	F-> $\epsilon$

e) Überprüfen sie die Ergebnisse mit <https://mikedevic.github.io/first-follow/>  
Stimmt 

## Aufgabe 21

Gegeben sei folgende Grammatik:  $G=(N,T,P,S)$ :

$N=\{S,A,B,C\}; T=\{a,b,c\};$

$S: ABbCa ;$

$A: abA \mid abBc ;$

$B \rightarrow Baa \mid C ;$

$C \rightarrow Cc \mid bb ;$

- a) Führen Sie geeignete Linksfaktorisierungen durch und beseitigen Sie Linksrekursionen

a)

**Linksfaktorisierung:**

$A: abA \mid abBc$

Wird ersetzt durch

$A: abD$

$D: A \mid Bc$

**Beseitigung Linksrekursionen**

$B: Baa \mid C$

Wird ersetzt durch:

$B: CE$

$E: aaE \mid \varepsilon$

$C: Cc \mid bb$

Wird ersetzt durch:

$C: bbF$

$F: cF \mid \varepsilon$

Neue Grammatik:

$S: ABbCa$

$A: abD$

$D: A \mid Bc$

$B: CE$

$E: aaE \mid \varepsilon$

$C: bbF$

$F: cF \mid \varepsilon$

b) Bestimmen Sie alle predict-Mengen

Neue Grammatik:

S: ABbCa

A: abD

D: A | Bc

B: CE

E: aaE |  $\epsilon$

C:bbF

F:cF |  $\epsilon$

$X \rightarrow s$	$s \rightarrow \epsilon?$	$\text{First}(s) \setminus \epsilon$	$\text{Follow}(X)$	$\text{Predict}(X \rightarrow s)$
$S \rightarrow ABbCa$	no	a	-	A
$A \rightarrow abD$	no	a	-	A
$D \rightarrow A$	no	a	-	A
$D \rightarrow Bc$	no	b	-	B
$B \rightarrow CE$	no	b	-	B
$E \rightarrow aaE$	no	a		A
$E \rightarrow \epsilon$	yes	-	$\text{Follow}(E)$ $= \text{Follow}(B)$ $= \{b, c\}$	b, c
$C \rightarrow bbF$	no	b	-	B
$F \rightarrow cF$	no	c	-	C
$F \rightarrow \epsilon$	yes	-	$\text{Follow}(F)$ $= \text{Follow}(C)$ $\{a\} \cup \text{First}(E) \setminus \epsilon \cup \text{Follow}(B)$ $\{a\} \cup \{a\} \cup \{b\} \cup \{c\}$ $= \{a, b, c\}$	a, b, c

c)

$\text{predict}(D \rightarrow A) = \{a\}$  und  $\text{predict}(D \rightarrow Bc) = \{b\}$  ist  
 $\text{predict}(D \rightarrow A) \cap \text{predict}(D \rightarrow Bc) = \{\}$

$\text{predict}(E \rightarrow aaE) = \{a\}$  und  $\text{predict}(E \rightarrow \epsilon) = \{b, c\}$  ist  
 $\text{predict}(E \rightarrow aaE) \cap \text{predict}(E \rightarrow \epsilon) = \{\}$

$\text{predict}(F \rightarrow cF) = \{c\}$  und  $\text{predict}(F \rightarrow \epsilon) = \{a, b, c\}$  ist  
 $\text{predict}(F \rightarrow cF) \cap \text{predict}(F \rightarrow \epsilon) = \{c\} \neq \{\}$

also ist für die neue Grammatik LL(1) nicht erfüllt!

c) Erstellen Sie die Parser-Tabelle für die neue Grammatik:

	A	b	c	\$
S	S->ABbCa			
A	A->abD			
B		B->CE		
C		C->bbF		
D	D->A	D->Bc		
E	E->aaE	E-> $\epsilon$	E-> $\epsilon$	
F	F-> $\epsilon$	F-> $\epsilon$	F->cF F-> $\epsilon$	

d) Überprüfen sie die Ergebnisse mit <https://mikedevise.github.io/first-follow/>



## Aufgabe 22

Erstellen Sie Top-Down Parser nach der Methode des rekursiven Abstiegs, die die Sprachen aus den Aufgaben 18-21 erkennen, falls möglich.

Nur möglich für Grammatik aus Aufgabe 20:

```
void S()
{
    if (token=='b') // regel S->ABbC angewandt");
    {
        printf("regel S->ABbC angewandt\n");
        A(); B(); check('b'); C();
    }
    else
    {
        printf("Syntaxfehler: b erwartet\n");
        return;
    }
}
```

```
void A()
{
    if (token == 'b') // regel A->bbD angewandt
    {
        printf("regel A->bbD angewandt\n");
        check('b');
        check('b');
        D();
    }
    else
    {
        printf("Syntaxfehler: b erwartet\n");
        return;
    }
}
```

```
void B()
{
    if (token == 'a') // regel B->aE angewandt
    {
        printf("regel B->aE angewandt\n");
        check('a');E();
    }
    else
    {
        printf("Syntaxfehler: a erwartet\n");
        return;
    }
}
```

```
void C()
{
    if ((token == 'a') || (token == 'b') || (token == 'c') || (token == '\n')) //
regel C->F angewandt
    {
        printf("regel C->F angewandt\n");
        F();
    }
    else
    {
        printf("Syntaxfehler: a,b,c erwartet\n");
        return;
    }
}

void D()
{
    if (token == 'a') // regel D->B angewandt
    {
        printf("regel D->B angewandt\n");
        B();
    }
    else if (token == 'b') // regel D->A angewandt
    {
        printf("regel D->A angewandt\n");
        A();
    }
    else
    {
        printf("Syntaxfehler: a,b erwartet\n");
        return;
    }
}

void E()
{
    if ((token == 'a') || (token == 'b')) // regel E->epsilon angewandt
    {
        printf("regel E->epsilon angewandt\n");
    }
    else if (token == 'c') // regel E->cC angewandt
    {
        printf("regel E->cC angewandt\n");
        check('c'); C();
    }
    else
    {
        printf("Syntaxfehler: a,b,c erwartet\n");
        return;
    }
}
```

```
void F()
{
    if ((token == 'a') || (token == 'b') || (token == '\n')) // regel F->epsilon
angewandt
    {
        printf("regel F->epsilon angewandt\n");
    }
    else if (token == 'c') // regel F->cF angewandt
    {
        printf("regel F->cF angewandt\n");
        check('c'); F();
    }
    else
    {
        printf("Syntaxfehler: a,b,c erwartet\n");
        return;
    }
}
```