

SWP Assignment 1

Alexander Frewein (01430019)
Klaus Fabian Frühwirth (01131523)
Stephany Amizic (01331786)

Institute of Software Technology
`alexander.frewein@student.tugraz.at`
`fabian.fruehwirth@student.tugraz.at`
`stephany.amizic@student.tugraz.at`

Beispiel 1

a.)

$$L = \{\underline{a}(\underline{aa|b})^* \underline{c}\}$$

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

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

$$S \rightarrow \underline{a}C$$

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

$$A \rightarrow \underline{c}D$$

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

$$B \rightarrow \underline{c}D$$

$$C \rightarrow \underline{c}D$$

$$D \rightarrow \epsilon$$

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

b.)

$$L = \{\underline{a}^{(2n)} \underline{b} \underline{c}^* (\underline{bb|d}) \mid n > 0\}$$

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

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

$$A \rightarrow \underline{b}C$$

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

$$A \rightarrow \underline{b}D$$

$$B \rightarrow \underline{b}E$$

$$B \rightarrow \epsilon$$

$$C \rightarrow \underline{c}C$$

$$C \rightarrow \underline{c}B$$

$$C \rightarrow \underline{c}D$$

$$D \rightarrow \underline{d}D$$

$$D \rightarrow \epsilon$$

$$E \rightarrow \underline{b}B$$

Beispiel 2

a.)

Dies ist eine **allgemeine** Grammatik da $|\alpha| \leq |\beta|$ **nicht** gilt und somit keine Restriktion $\alpha \rightarrow \beta$ gilt

b.)

Dies ist eine **reguläre** Grammatik da $|\alpha| \leq |\beta|$, $\alpha \in V_N$ β hat form aA oder a , mit $a \in V_T \cup \{\epsilon\}$, $A \in V_N$

c.)

Dies ist eine **kontextfreie** Grammatik da $|\alpha| \leq |\beta|$, $\alpha \in V_N$

d.)

Dies ist keine **gültige** Grammatik da $R \rightarrow Qy$ nicht laut Definition $\alpha\beta \in (V_N \cup V_T)$ diese Form nicht in der Grammatik definiert ist.

e.)

Dies ist eine **allgemeine** Grammatik da $|\alpha| \leq |\beta|$ **nicht** gilt und somit keine Restriktion $\alpha \rightarrow \beta$ gilt

f.)

Dies ist keine **gültige** Grammatik da $\underline{num} \rightarrow \underline{var}$ nicht in den Grammatik definiert ist

Beispiel 3

First und Follow Mengen:

	FIRST	FOLLOW
S	<u>a</u> <u>b</u> <u>c</u> <u>d</u> <u>e</u>	\$
A	<u>a</u> <u>b</u> <u>c</u>	<u>d</u> <u>e</u>
B	<u>d</u> <u>e</u>	<u>c</u> <u>d</u> <u>e</u>
C	<u>b</u>	<u>c</u> <u>d</u> <u>e</u>
D	<u>c</u>	<u>d</u> <u>e</u>
E	<u>d</u> <u>e</u>	<u>c</u> <u>d</u> <u>e</u>

LL(1) Tabelle

	<u>a</u>	<u>b</u>	<u>c</u>	<u>d</u>	<u>e</u>	\$
S	S → AB	S → AB	S → AB	S → AB	S → AB	S → AB
A	A → <u>a</u> A	A → CD	A → CD			
B				B → E	B → E	
C		C → <u>b</u> B	C → ε	C → ε	C → ε	
D			D → <u>c</u> C	D → ε	D → ε	
E				E → <u>d</u>	E → <u>e</u>	

Beispiel 4

Gegeben ist die folgende LL(1) Tabelle, welche eine grobe Abstraktion der Variablendeklaration in Scala beschreibt. Die unterstrichenen Zeichenketten in den Spalten der ersten Zeile stellen jeweils ein Terminalsymbol dar.

	<u>var</u>	<u>val</u>	<u>one</u>	<u>two</u>	<u>String</u>	<u>Int</u>	<u>0</u>	<u>1</u>	<u>"</u>	<u>=</u>	<u>\$</u>
S	AB	AB									
A	CN _:	CN _:									
B					String="V"	Int=U					
C	<u>var</u>	<u>val</u>									
N			<u>one</u>	<u>two</u>							
U							<u>0</u> V	<u>1</u> V			
V							<u>0</u> V	<u>1</u> V	ε	ε	

Überprüfen Sie mittels der gegebenen LL(1) Tabelle ob folgende Ausdrücke gültige Sätze der definierten Grammatik sind: a) var one : String = 1

b) val two : int = 10

Die Lösung für die Unterpunkte a und b soll im folgenden Format erarbeitet und abgegeben werden:

Stack	Input	Produktion/Kommentar
\$S	val one: Int= "11"\$...

a.)

Stack	Input	Produktion/Kommentar
\$S	var one : String = 1 \$	S:= AB
\$BA	var one : String = 1\$	A:= CN _:
\$B _: NC	var one : String = 1 \$	C:= <u>var</u>
\$B _: N var	var one : String = 1\$	
\$B _: N	one : String= 1\$	N:= <u>one</u>
\$B _: one	one : String= 1\$	
\$B	String = 1\$	B:= <u>String="V"</u>
\$"V" <u>=String</u>	<u>String</u> =1\$	
\$"V"	1\$	Keine Regel

Diese Satz ist nicht Valid!

b.)

Stack	Input	Produktion/Kommentar
\$S	val two : int = 10\$	S:= AB
\$BA	val two : int= 10\$	A:= CN ₂
\$B ₂ NC	val two : int= 10\$	C:= val
\$B ₂ Nval	val two : int = 10\$	
\$B ₂ N	two : int = 10\$	N:= <u>two</u>
\$B ₂ two	two : int = 10\$	
\$B	int= 10\$	B:= <u>int</u> =U
\$U= int	int = 10\$	
\$U	10\$	U:= <u>1</u> V
\$V 1	10 \$	
\$V	0\$	V:= <u>0</u> V
\$V0	0 \$	
\$V	\$	V:= ϵ
\$	\$	Valid

Diese Satz ist Valid!

Beispiel 5

Die gegebene Gramatik ist keine LL(1) Gramatik weil:

- X ist Mehrdeutig (Linksfaktorisierung)
- Y ist indirekt Links recursiv (über Z)

Äquivalente LL(1) Grammatik:

$$\begin{aligned}
 S &\rightarrow XY \\
 V &\rightarrow \underline{x} \\
 V &\rightarrow \underline{=} \\
 W &\rightarrow \underline{0}W \\
 W &\rightarrow \underline{1}W \\
 W &\rightarrow \epsilon \\
 X &\rightarrow \underline{x}X_2 \\
 X_2 &\rightarrow V \\
 X_2 &\rightarrow W \\
 Y &\rightarrow \underline{+}Y_2 \\
 Y_2 &\rightarrow \underline{y}WY_2 \\
 Y_2 &\rightarrow \epsilon
 \end{aligned}$$

LL(1) Tabelle

	<u>x</u>	<u>=</u>	<u>0</u>	<u>1</u>	<u>±</u>	<u>y</u>	\$
S	$S \rightarrow XY$						
V	$V \rightarrow X$	$V \rightarrow \underline{=}$					
W			$W \rightarrow \underline{0}W$	$W \rightarrow \underline{1}W$	$W \rightarrow \epsilon$	$W \rightarrow \epsilon$	$W \rightarrow \epsilon$
X	$X \rightarrow \underline{x}X_2$						
X ₂	$X_2 \rightarrow V$	$X_2 \rightarrow V$	$X_2 \rightarrow W$	$X_2 \rightarrow W$	$X_2 \rightarrow W$		
Y					$Y \rightarrow \underline{+}Y_2$		
Y ₂						$Y_2 \rightarrow \underline{y}WY_2$	$Y_2 \rightarrow \epsilon$

Beispiel 6

a.)

```
object SimpleParserA extends JavaTokenParsers {
  def A: Parser[Any] = "aa" ~ A | "c" ~ ""
  def B: Parser[Any] = "b" ~ B | "c" ~ ""
  def AB: Parser[Any] = A | B

  def S: Parser[Any] = "a" ~ AB | "ac"
  def apply(s: String) = parseAll(S, s)
}
```

a.)

```
object SimpleParserB extends JavaTokenParsers {
  def N: Parser[Double] = "[1-9][0-9]*|0|-[1-9][0-9]*".r
    ^^ {i => i.toDouble}
  def S: Parser[Double] = N ~ " * 10 ^ " ~ N ^^ {case a ~ _ ~ b => a * math.pow(10, b)}
  def apply(s: String) = parseAll(S, s)
}
```