

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 → <u>C</u> D			
B				B → E	B → E	
C		C → <u>b</u> B	C → <u>ε</u>	C → <u>ε</u>	C → <u>ε</u>	
D			D → <u>c</u> C	D → <u>ε</u>	D → <u>ε</u>	
E				E → <u>d</u>	E → <u>e</u>	

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)
}

```