

# Blatt 01: Reguläre Sprachen

## A1.1: Sprachen von regulären Ausdrücken (1P)

regex:  $\$a \backslash + \backslash a \backslash (a \backslash + \backslash b)^* \backslash a \$$  welche Sprache?

### Antwort:

Das Wort "a" (Teil vor dem +) oder Wörter beginnend mit "a", endend mit "a" und dazwischen beliebig viele oder keine "a" oder "b"

$\{ a \}$  oder  $\{ a (a \text{ oder } b \text{ oder nichts})^* a \}$

## A1.2: Bezeichner in Programmiersprachen (3P)

Bezeichneraufbau:

- Variablennamen beginnen mit **V**(global) oder **v**(lokal)
- Funktions- und Methodenparameter mit **p**, Klassenparameter (Definition von Vererbung) mit **P**
- Weitere müssen mit (a-z, A-Z) beginnen
- folgen dürfen Buchstaben, Ziffern und Unterstrich
- dürfen nicht mit Unterstrich enden
- müssen mindestens zwei Zeichen haben

regulärer Ausdruck + 2 Beispiele

### Antwort:

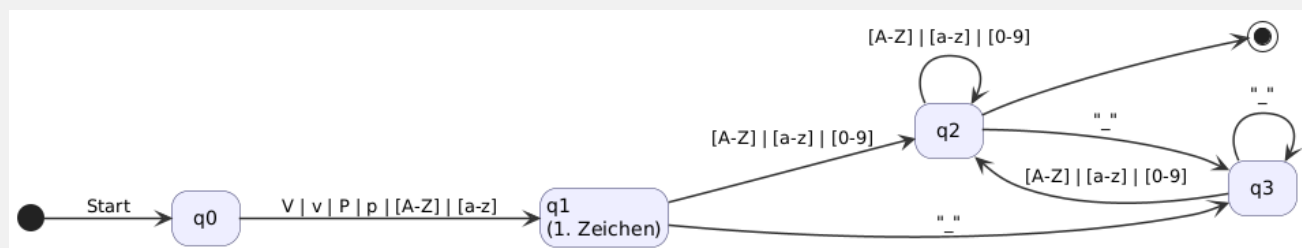
$\$L = (V+v+P+p+a-z+A-Z)(a-z+A-Z+0-9+_)^*(a-z+A-Z+0-9)\$$

### Beispiele:

Bezeichner	Teil	Teil des regex
Vcount1	V	$\$(V+v+P+p+a-z+A-Z)\$$
	count	$\$(a-z+A-Z+0-9+_)^*\$$
	1	$\$(a-z+A-Z+0-9)\$$
p_name_3	p	$\$(V+v+P+p+a-z+A-Z)\$$
	_name_	$\$(a-z+A-Z+0-9+_)^*\$$
	3	$\$(a-z+A-Z+0-9)\$$

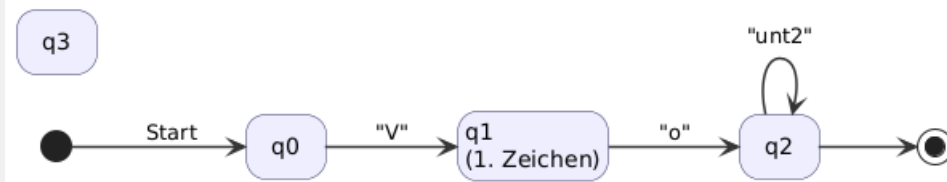
DFA + 2 Beispiele

### Antwort:

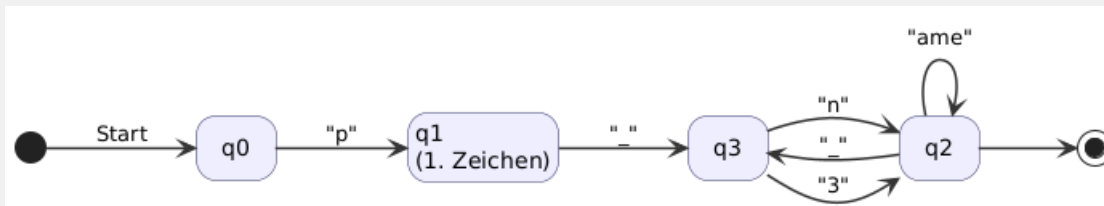


### Beispiele:

- Vcount1 = start -> q0 --"V"--> q1 --"o"--> q2 --"unt2"--> q2 -> ende



- p\_name\_3 = start -> q0 --"p"--> q1 --"\_"--> q3 --"n"--> q2 --"ame"--> q2 --"\_"--> q3 --"3"--> q2 --> ende



## reguläre Grammatik + 2 Beispiele

### Antwort:

$N = \{S, A, B, C\}$

$T = \{\{a-z + A-Z\}: \text{buchstabe}, \{0-9\}: \text{zahl}, \_ : \text{unterstrich}\}$

$S = S$

$P = \{$

- $S \rightarrow fA \text{ f} \in \{V, v, P, p\} \cup \text{buchstabe}$
- $A \rightarrow (\text{buchstabe}) B \mid (\text{zahl}) B \mid (\text{unterstrich}) C$
- $B \rightarrow (\text{buchstabe}) B \mid (\text{zahl}) B \mid \epsilon$
- $C \rightarrow (\text{buchstabe}) B \mid (\text{zahl}) B \mid (\text{unterstrich}) C$

$\}$  (Produktionen)

### Beispiel

Vcount1

$S \rightarrow "V"A \rightarrow "Vc"B \rightarrow "Vco"B \rightarrow "Vcou"B \rightarrow "Vcoun"B \rightarrow "Vcount"B \rightarrow "Vcount1"B \rightarrow "Vcount1\epsilon"$

p\_name\_3

$S \rightarrow "p"A \rightarrow "p\_C \rightarrow "p\_n"B \rightarrow "p\_na"B \rightarrow "p\_nam"B \rightarrow "p\_name"B \rightarrow "p\_name\_C \rightarrow "p\_name\_3"B \rightarrow "p\_name\_3\epsilon"$

## A1.3: Gleitkommazahlen in Programmiersprachen (2P)

Aufbau Gleitkommazahlen in Python und Java.

### Antwort:

Java:

- optionales Vorzeichen
- mind. eine Ziffer vor dem Dezimalpunkt
- optionaler Dezimalteil (Punkt + Nachkommastelle)
- optionaler Exponentialteil (opt. Vorzeichen, Ziffern)

Python: float

- optionales Vorzeichen
- mind. eine Ziffer vor dem Dezimalpunkt
- optionaler Dezimalteil (Punkt + Nachkommastelle)

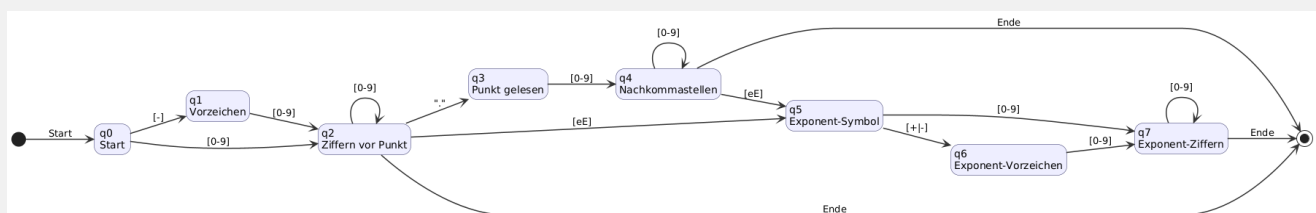
Regex, DFA, Grammatik

Java

- Regex:  $\$[-,+]?[0-9]^+(\.[0-9])?((e+E)[+ -]?[0-9]^+)?\$$
- Beispiele:

Zahl	Teil	Teil des regex
-12.34e+5	-	$[-,+]?$
12		$[0-9]^+$
.34		$(\.[0-9]^+)?$
e+5		$((e+E)([+,-])?[0-9]^+)?$
3.1415	3	$[0-9]^+$
.1415		$(\.[0-9]^+)?$
(kein Exponent)		

- DFA



hier muss bei q0-q1 das plus noch mitgegeben werden

- Beispiele:
  - 3.1415 : start --> q0 --"3"--> q2 --"."--> q3 --"1415"--> q4 --> ende
  - -12.34e+5 : start --> q0 --"1"--> q1 --"2"--> q2 --"."--> q3 --"34"--> q4 --"e"--> q5 --"+"--> q6 --"5"--> q7 --> ende
- Grammatik

$N = \{S, A, B, C, D, E, F\}$

$T = \{+, -\}$ : vorzeichen,  $\{0-9\}$ : zahl,  $\cdot$ : punkt,  $\{e, E\}$ : expo

$S = S$

$P = \{$

- $S \rightarrow (\text{vorzeichen}) A \mid A$
- $A \rightarrow (\text{zahl}) B$
- $B \rightarrow (\text{zahl}) B \mid (\text{punkt}) C \mid (\text{expo}) D \mid \epsilon$
- $C \rightarrow (\text{zahl}) C \mid (\text{expo}) D \mid \epsilon$
- $D \rightarrow (\text{vorzeichen}) E \mid E$
- $E \rightarrow (\text{zahl}) F$
- $F \rightarrow (\text{zahl}) F \mid \epsilon$

$\}$  (Produktionen)

### Beispiel

3.1415

$S \rightarrow A \rightarrow 3 B \rightarrow 3 \cdot C \rightarrow 3 \cdot 1 C \rightarrow 3 \cdot 14 C \rightarrow 3 \cdot 141 C \rightarrow 3 \cdot 1415 C \rightarrow 3 \cdot 1415 \epsilon$

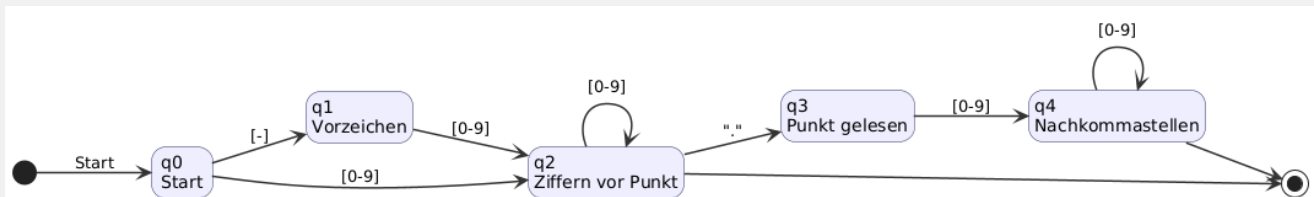
-12.34e+5

$S \rightarrow - A \rightarrow - 1 B \rightarrow - 12 B \rightarrow - 12 \cdot C \rightarrow - 12 \cdot 3 C \rightarrow - 12 \cdot 34 C \rightarrow - 12 \cdot 34 e D \rightarrow - 12 \cdot 34 e + E \rightarrow - 12 \cdot 34 e + 5 F \rightarrow - 12 \cdot 34 e + 5 \epsilon$

- Regex:  $\$-?[0-9]^+([0-9]^+)?\$$
- Beispiele:

Zahl	Teil	Teil des regex
-12.34	-	-?
12		$[0-9]^+$
.34		$([0-9]^+)?$
3.1415	3	$[0-9]^+$
.1415		$([0-9]^+)?$

- DFA



- Beispiele:
  - 3.1415 : start --> q0 --"3"--> q2 --"."--> q3 --"1415"--> q4 --> ende
  - -12.34 : start --> q0 --"1"--> q1 --"2"--> q2 --"."--> q3 --"34"--> q4 --> ende

- Grammatik

- $N = \{S, A, B, C\}$
- $T = \{-\}: \text{vorzeichen}, \{0-9\}: \text{zahl}, \{.\}: \text{punkt}$
- $S = S$
- $P = \{$ 
  - $S \rightarrow (\text{vorzeichen}) A \mid A$
  - $A \rightarrow (\text{zahl}) B$
  - $B \rightarrow (\text{zahl}) B \mid (\text{punkt}) C \mid \epsilon$
  - $C \rightarrow (\text{zahl}) C \mid \epsilon$

} (Produktionen)

### Beispiel

- 3.1415
  - $S \rightarrow A \rightarrow 3 B \rightarrow 3 . C \rightarrow 3 . 1 C \rightarrow 3 . 14 C \rightarrow 3 . 141 C \rightarrow 3 . 1415 C \rightarrow 3 . 1415 \epsilon$
- -12.34
  - $S \rightarrow - A \rightarrow - 1 B \rightarrow - 12 B \rightarrow - 12 . C \rightarrow - 12 . 3 C \rightarrow - 12 . 34 C \rightarrow - 12 . 34 \epsilon$

## A1.4: Mailadressen? (1P)

Warum  $\$(a-z)^+@(a-z) \cdot (a-z)\$$  ungeeignet für emails?

- erlaubt keine großbuchstaben
- keine zahlen
- nur ein kleiner buchstabe nach @ und nach .
- keine sonderzeichen erlaubt
- -> zu wenig möglichkeiten, eine sole email wirds nicht geben

$\$a + b + c + c + \backslash\text{ldots} + z\$$  ist besser, aber immer noch nicht richtig. Warum?

- Weil weiterhin keine Großbuchstaben akzeptiert werden

- c ist doppelt
- theoretisch erlauben wir damit a, b, c, "...", z, also nicht das alphabet

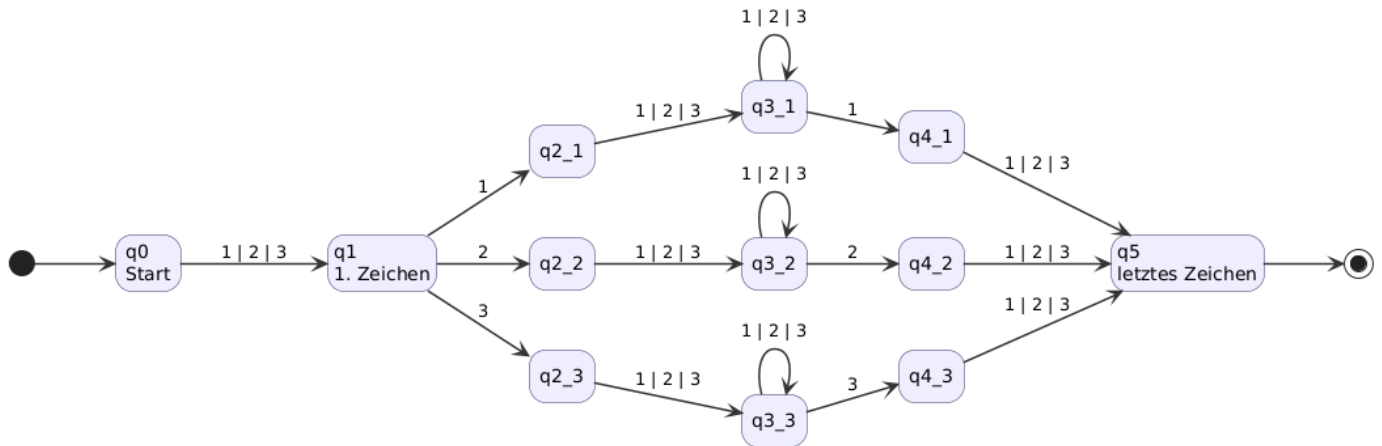
Verbessert

$\$((a-z, A-Z)^+, [0-9]^+)^+ @ (a-z)^+ . (a-z)^+ \$$

weiterhin nicht optimal, da keine sonderzeichen, könnte mit zahlen beginnen, etc

## A1.5: Der zweitletzte Buchstabe (1P)

$\$ \Sigma = \{1,2,3\}^*, \text{wort}[\text{länge}-2] == \text{wort}[1]$



## A1.6: Sprache einer regulären Grammatik (2P)

$S \rightarrow aA$

$A \rightarrow dB \mid bA \mid cA$

$B \rightarrow aC \mid bC \mid cA$

$C \rightarrow \epsilon$

- wörter beginnen mit a
- haben mindestens 2 zeichen
- bestehen aus {a,b,c,d}

$\$(a(b+c)+(dc))^*d(a+b)\$$

