

# Überprüfung von Klammersausdrücken

Der eigentliche Inhalt (z.B. arithmetisch, algebraisch) wird ignoriert:

$$(a / (b+c) - (a \cdot (c-b))) \rightarrow (( ( ( ( ( ) ) ) ) )$$

Hier: **nichtleere** Klammersausdrücke, also **nicht**  $b+c \rightarrow \varepsilon$ .

(Geht aber auch ähnlich mit Einschluss von  $\varepsilon$ .)

**Welche Eigenschaften kennzeichnen die korrekten Klammersausdrücke?**

**Wie lautet eine einfache Grammatik für sie?**

**Wie kann man sie mit einem (ND-)Kellerautomaten prüfen?**

**Wie kann man sie mit einem deterministischen Kellerautomaten prüfen?**

**Überprüfung von Klammerausdrücken wie ( ( ) ( ( ) ) )**

**Welche Eigenschaften kennzeichnen die korrekten Klammerausdrücke?**

# Überprüfung von Klammerausdrücken wie ( ( ) ( ( ) ) )

## Welche Eigenschaften kennzeichnen die korrekten Klammerausdrücke?

1. Insgesamt gleich viele ( und ).  
Aber nicht in beliebiger Reihenfolge: )( ist inkorrekt.
2. Mit ( kann stets ein korrekter Ausdruck fortgesetzt werden, was aber später ein entsprechendes ) erfordert.
3. Bis zu jeder Stelle gilt bzw. jedes Präfix hat: höchstens so viele ) wie (.
4. Mit ) kann ein korrekter Ausdruck genau dann fortgesetzt werden, wenn damit nicht gegen (3) verstoßen wird.

# Überprüfung von Klammerausdrücken wie ( ( ) ( ( ) ) )

## Welche Eigenschaften kennzeichnen die korrekten Klammerausdrücke?

1. Insgesamt gleich viele ( und ). Aber nicht in beliebiger Reihenfolge: )( ??
2. Mit ( kann stets ein korrekter Ausdruck fortgesetzt werden, was aber später ein entsprechendes ) erfordert.
3. Bis zu jeder Stelle gilt bzw. jedes Präfix hat: höchstens so viele ) wie (.
4. Mit ) kann ein korrekter Ausdruck genau dann fortgesetzt werden, wenn damit nicht gegen (3) verstoßen wird..

1&3 sind offenbar hinreichend und notwendig

**Überprüfung von Klammerausdrücken wie ( () ( () ) )**

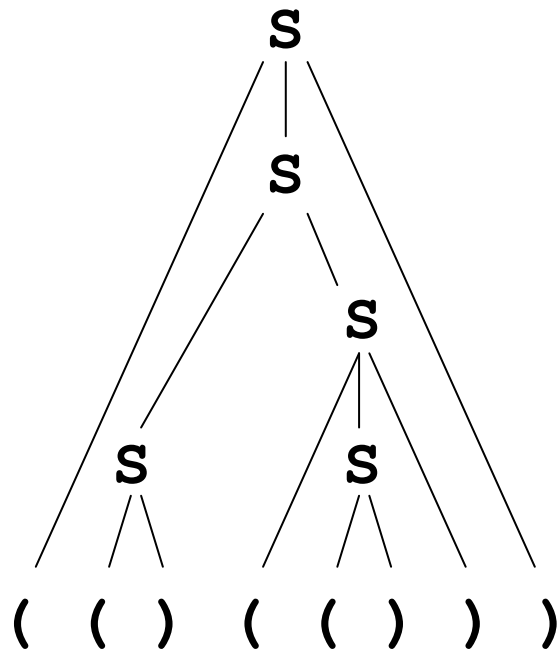
**Wie lautet eine einfache Grammatik für sie?**

# Überprüfung von Klammerausdrücken wie ( ( ) ( ( ) ) )

Wie lautet eine einfache Grammatik für sie?

- $\Sigma = \{ (, ) \}$
- $V = \{ S \}$
- Start: S
- Regeln:  
 $S \rightarrow ( ) \mid SS \mid (S)$

# Überprüfung von Klammerausdrücken – manuell, gemäß Grammatik:



## Überprüfung von Klammerausdrücken – (ND-) Kellerautomat, gemäß Rezept aus Grammatik:

- $Z = \{ z_0, z, z^* \}; \Gamma = V \cup \Sigma; F = \{ z^* \}$
- $\Delta$  enthält folgende Transitionen:
  - $(z_0, -, \$, z, S)$  und  $(z, -, \$, z^*, \epsilon)$
  - $(z, x, x, z, \epsilon)$  für alle  $x \in \Sigma$
  - $(z, x, V, z, r)$  für alle Regeln  $V \rightarrow r$  in  $R$



## Überprüfung von Klammersausdrücken – (ND-) Kellerautomat, gemäß Rezept aus Grammatik:

- $Z = \{ z_0, z, z^* \}; \Gamma = V \cup \Sigma; F = \{ z^* \}$
- $\Delta$  enthält folgende Transitionen:
  - $(z_0, -, \$, z, S)$  und  $(z, -, \$, z^*, \varepsilon)$
  - $(z, x, x, z, \varepsilon)$  für alle  $x \in \Sigma$
  - $(z, x, V, z, r)$  für alle Regeln  $V \rightarrow r$  in  $R$

$\Delta$ :

$(z_0, -, \$, z, S)$	$(z, \varepsilon, S, z, \text{“SS“})$
$(z, -, \$, z^*, \varepsilon)$	$(z, \varepsilon, S, z, \text{“()“})$
$(z, \text{‘(‘}, \text{‘(‘}, z_1, \varepsilon) \}$	$(z, \varepsilon, S, z, \text{“(S)“})$
$(z, \text{‘)‘}, \text{‘)‘}, (z, \varepsilon) \}$	

$z^*$  akzeptiert.

# Überprüfung von Klammersausdrücken – deterministischer Kellerautomat, ähnlich manueller Methode:

$(z_0, '(', \$, (z_1, '(')$   
 $(z_1, '(', '(', z_1, "(((" )$   
 $(z_1, ')', '(', z_1, \epsilon)$   
 $(z_1, -, \$, z_2, \epsilon)$   
 $(z_2, '(', \$, z_1, '(')$

$z^*$  akzeptiert.

Idee:

Im Stapel wird einfach festgehalten, um wie viele die  
öffnenden Klammern zurzeit den schließenden Klammern  
voraus sind.

# Überprüfung von Klammersausdrücken – deterministischer Kellerautomat, Beispiel:

( ( ) ( ( ) ) ) OK?

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

( ( ) ( ( ) ) )



auf den Stapel

-

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

( ) ( ( ) ) )

(  
-

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

( ) ( ( ) ) )



(  
\_

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

) ( ( ) ) )

(  
(  
-

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

) ( ( ) ) )

(  
(  
-



# Überprüfung von Klammersausdrücken – deterministischer Kellerautomat, Beispiel:

( ( ) )

(  
-

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

( ( ) )



(  
\_

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

( ) )

(  
(  
-

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

( ) ) )



(  
(  
-

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

) ) )

(  
(  
(  
-

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

) ) )

(  
(  
(  
-

# Überprüfung von Klammersausdrücken – deterministischer Kellerautomat, Beispiel:

) )

(  
(  
-

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

) )

(  
(  
-



# Überprüfung von Klammersausdrücken – deterministischer Kellerautomat, Beispiel:

)

(  
-

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

)

(  
-

# Überprüfung von Klammerausdrücken – deterministischer Kellerautomat, Beispiel:

$\varepsilon$

-

OK!

# Übungen

- Formulieren Sie die Arbeitsweise des deterministischen Kellerautomaten programmiersprachlich.

Verwenden Sie dabei:

- `pop(st)` – der neue kleinere Stapel, nachdem das oberste Stapелеlement von Stapel *st* entfernt wurde; verändert nichts wenn *st* leer.
- `push(e!,st)` – der neue größere Stapel, nachdem *e!* auf Stack *st* gelegt wurde.
- `top(st)` – (liest) das oberste Element von Stapel *st*, (ohne es zu entfernen); ersatzweise \$, wenn *st* leer.  
\$ sei keines der zulässigen Elemente.

Was tut Ihr Programm, wenn die Wörter `((` bzw. `)` geprüft werden?

- Erweitern Sie das Verfahren auf Klammersausdrücke mit `,` `[` und `]`. Testen Sie es mindestens mit `( [ ] ( [ ] ) )` (korrekt!) und `( [ ] ]` (falsch!).  
Grammatik?