

## Klammerausdrücke

In Programmierumgebungen kommen Abfolgen von Klammern, runde, eckige und geschweifte, vor. Diese müssen in der richtigen Abfolge auf- bzw. geschlossen werden. Eine korrekte Abfolge von Klammern wäre zum Beispiel:

1 { [] () { () { [] ( [] ) } } } { }

- (a) Entwerfen Sie eine Grammatik, die die korrekte Abfolge solcher Klammerfolgen beschreibt.

Erläuterungen zu den Variablennamen:

**S** Start

**G** geschweifte Klammern ( {} )

**E** eckige Klammern ( [] )

**R** runde Klammern ( () )

**V** Verkettung

```
1 S -> G | E | R | V | EPSILON
2 G -> { S } | { }
3 E -> [ S ] | [ ]
4 R -> ( S ) | ( )
5 V -> S S
```

Die Produktionsregeln kann man auch zu einer Regel zusammenfassen.

```
1 S -> { S } | { } | ( S ) | ( ) | [ S ] | [ ] | S S | EPSILON
```

Damit eine „abwechslungsreichere“ Ableitung entsteht, wir hier die auf mehrere Regeln aufgeteilte Grammatik präferiert.

- (b) Geben Sie eine Ableitung für den oben angegebenen Klammerausdruck an.

Vor den Knoten/Variablen G E R V müsste eigentlich jedesmal ein S stehen. Da der Baum / die Ableitung sonst zu unübersichtlich wird, wurde auf die S-Knoten verzichtet. Auch auf die Epsilon-Übergänge wurde aus denselben Grund verzichtet.

```

1  S ->
2  V ->
3  GG ->
4  { V }{} ->
5  { E R G }{} ->
6  {[ ]({ V })}{} ->
7  {[ ]({ R G })}{} ->
8  {[ ]({({ V })})}{} ->
9  {[ ]({({ E R })})}{} ->
10 {[ ]({({({ E })})})}{} ->
11 {[ ]({({({({ })})})})}{}

```

