

Aufgabe 2: Grammatik vereinfachter boolescher Ausdrücke

Erstellen Sie eine Grammatik boolescher Ausdrücke, die als Operanden Namen (*ident*) sowie die Konstanten *true* und *false* haben und als Operatoren *&&*, *||* und *!* (wobei *!* Vorrang vor *&&* und *&&* Vorrang vor *||* haben soll). Teilausdrücke sollen auch geklammert werden können. Orientieren Sie sich dabei an der Grammatik arithmetischer Ausdrücke aus Abschnitt 1.4. Beispiel:

(big || small) && ready || big && ! ready

Lösung

Wie bei arithmetischen Ausdrücken brauchen wir eine oberste Ebene für *||*-Verknüpfungen, eine darunterliegende Ebene für *&&*-Verknüpfungen und eine unterste Ebene für die Operanden und die Negation (*!*). Damit ergibt sich folgende Grammatik:

```

BoolExpr = BoolTerm {"||" BoolTerm}.
BoolTerm = BoolFactor {"&&" BoolFactor}.
BoolFactor =
  ident
  | "true"
  | "false"
  | "!" BoolFactor
  | "(" BoolExpr ")".
    
```

Durch die Reihenfolge der Produktionen sind auch die Vorrangregeln der Operatoren definiert. Der Operator *!* liegt auf unterster Ebene der Grammatik und hat daher Vorrang vor dem Operator *&&*, der auf mittlerer Ebene liegt, und dieser hat wiederum Vorrang vor dem Operator *||*, der auf höchster Ebene in der Grammatik steht. Der Ausdruck

(big || small) && ready || big && ! ready

wird wie in folgendem Syntaxbaum dargestellt erkannt.

