Lösungen zu den Übungsaufgaben im Buch *Automaten und Sprachen: Theoretische Informatik für die Praxis* von Andreas Müller, ISBN 978-3-662-70145-4 (Softcover), ISBN 978-3-662-70146-1 (eBook), https://link.springer.com/book/10.1007/978-3-662-70146-1. Website zum Buch: https://autospr.ch

Kapitel 5: Kontextfreie Grammatiken und Sprachen

5.1. Formulieren Sie eine kontextfreie Grammatik für die folgenden Sprachen über den Terminalsymbolen $\Sigma = \{\texttt{0}, \texttt{1}\}$:

a) $L = \Sigma^*$

b) Wörter, die mit dem gleichen Symbol enden, mit dem sie beginnen.

c) L besteht aus den Wörtern gerader Länge.

d) $L = \{\texttt{0}^n\texttt{1}^m \mid n > m\}$

Lösung. a) Jedes Wort kann man erhalten, indem man dem leeren Wort ε wiederholt $\texttt{0}$ oder $\texttt{1}$ anhängt.

$$\begin{aligned} W &\to W\texttt{0} \\ &\to W\texttt{1} \\ &\to \varepsilon \end{aligned}$$

b) Sei E eine Variable, die für Wörter mit gleichem Anfangs- und Endzeichen steht. Dazu gehören ε, $\texttt{0}$ und $\texttt{1}$. Längere Wörter kann man aus beliebigen Wörtern W in Σ^* erzeugen, indem man das gleiche Symbol voranstellt und anhängt. Für beliebige Wörter kann man die Produktionsregeln aus a) verwenden.

$$\begin{aligned} E &\to \varepsilon \\ &\to \texttt{0} \\ &\to \texttt{1} \\ &\to \texttt{0}W\texttt{0} \\ &\to \texttt{1}W\texttt{1} \\ W &\to W\texttt{0} \\ &\to W\texttt{1} \\ &\to \varepsilon \end{aligned}$$

c) Die Variable G steht für Wörter gerader Länge. Ein Wort gerader Länge kann man bilden, indem man einem Wort gerader Länge ein Zeichenpaar anhängt. Sei P eine Variable, die für ein Zeichenpaar steht. Die Variable Z soll für ein einzelnes Zeichen stehen, die Produktionsregeln der Grammatik sind damit:

$$G \to GP$$

$$
\begin{aligned}
&\to \varepsilon \\
P &\to ZZ \\
Z &\to \mathtt{0} \\
&\to 1
\end{aligned}
$$

d) Das kürzeste Wort der gesuchten Art ist $\mathtt{0}$. Sei S eine Variable, die für Wörter der Sprache steht. Wörter in L kann man aus bereits produzierten Wörtern in L produzieren, indem man eine $\mathtt{0}$ voranstellt, und optional eine 1 anhängt. Dies führt auf die Produktionsregeln

$$
\begin{aligned}
S &\to \mathtt{0}S \\
&\to \mathtt{0}S\,1 \\
&\to \mathtt{0}
\end{aligned}
$$

○

5.2. Für arithmetische Ausdrücke mit den Grundoperationen + und - stehen die folgenden zwei kontextfreie Grammatiken zur Auswahl. Einerseits eine vereinfachte Variante der in Abschnitt 5.3.1 besprochenen Expression-Term-Faktor-Grammatik:

$$
\begin{aligned}
\text{Ausdruck} &\to \text{Zahl '+' Ausdruck} \\
&\to \text{Zahl '-' Ausdruck} \\
&\to \text{Zahl,}
\end{aligned}
$$

die Rechtsrekursion statt Linksrekursion verwendet. Andererseits die folgende Grammatik:

$$
\begin{aligned}
\text{Ausdruck} &\to \text{Term Termsequenz} \\
\text{Termsequenz} &\to \text{'+' Term Termsequenz} \\
&\to \text{'-' Term Termsequenz} \\
&\to \varepsilon \\
\text{Term} &\to \text{Zahl.}
\end{aligned}
$$

Die beiden Grammatiken verwenden natürlich die gleichen Regeln für das nichtterminale Symbol "Zahl".

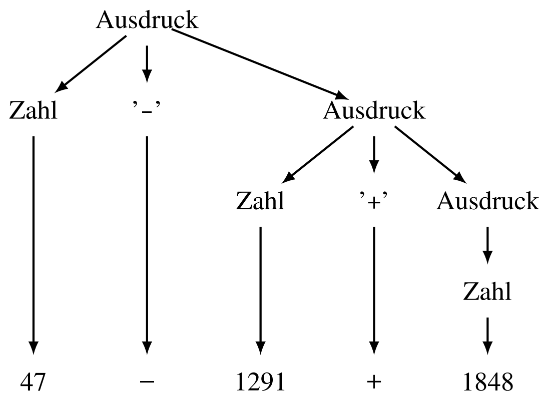
a) Erstellen Sie für jede Grammatik den Syntaxbaum für den Ausdruck

$$47 - 1291 + 1848$$

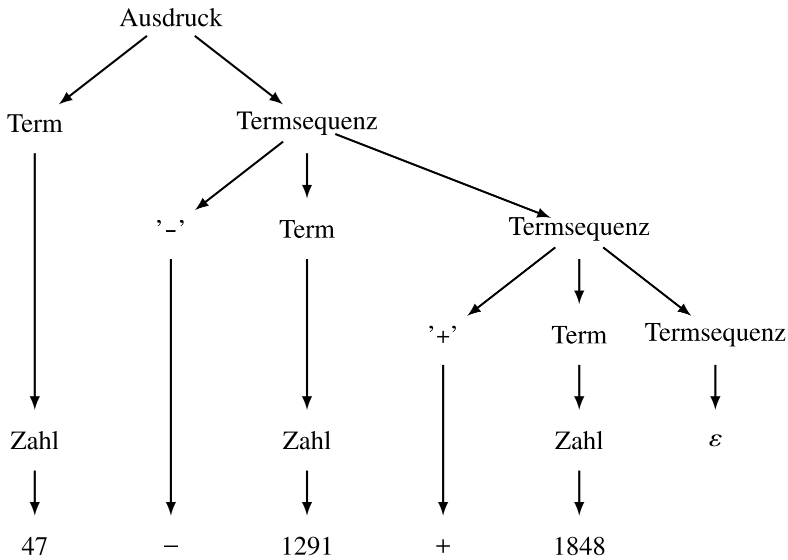
b) Formulieren Sie Regeln, wie der arithmetische Ausdruck auf der Basis des Syntaxbaums auszuwerten ist.

c) Welche Grammatik ist vorzuziehen?

Lösung. a) Die erste Grammatik erzeugt



Bei Verwendung der zweiten Grammatik bekommt man jedoch



b) Um mit einem Syntaxbaum einen Ausdruck auszuwerten, muss man für jede Regel festlegen, wie sie in die Wertberechnung übersetzt werden muss.

Um für die erste Grammatik festzulegen, wie ein Ausdruck auszuwerten ist, muss man zuerst den Ausdruck auf der rechten Seite der Regel auswerten, und dann die arithmetische Operation mit der Zahl entsprechend dem Operationszeichen durchführen.

Die Auswertung einer Regel nach der zweiten Grammatik erfolgt anders: zum Wert der Termsequenz wird der Wert des Terms hinzuaddiert, mit dem Vorzeichen, das in der Regel gewählt worden ist.

c) Für die erste Grammatik wird zuerst die Summe $1291 + 1848 = 3139$ gebildet, die anschließend von 47 subtrahiert, was $47 - (1291 + 1848) = -3092$ ergibt. Der erste Syntaxbaum ergibt also das falsche Resultat.

Das richtige Resultat erhält man jedoch mit dem Stanxbaum nach der zweiten Grammatik. Von unten nach oben ausgewertet bedeutet das wieder, dass zuerst die Termsequenz in der dritten Zeile zu 1848 evaluiert wird. Die Termsequenz in der zweiten Zeile kann dann als $-1291 + 1848$ ausgerechnet werden. Der Ausdruck ist dann 47 plus das bisherige Resultat, also 604, das richtige Resultate. Daher ist die zweite Grammatik vorzuziehen. ○

5.3. In einer SQL INSERT Query müssen Spaltennamen der Datenbank und Werte in gleicher Zahl angegeben werden:

```
INSERT INTO blubb(a1, b2, c3) VALUES('wert1', 'wert2', 1291);
```

Es stellt sich die Frage, ob man bereits durch die Grammatik sicherstellen kann, dass gleich viele Spaltennamen wie Werte angegeben werden. Betrachten Sie dazu die Sprache über dem Alphabet $\Sigma = \{\texttt{(}, \texttt{)}, \texttt{,}\}$, bestehend aus Wörtern der Form

$$w = \texttt{(}\underbrace{\texttt{,}\,\texttt{,}\,\dots\,\texttt{,}}_{n}\texttt{)}\texttt{(}\underbrace{\texttt{,}\,\texttt{,}\,\dots\,\texttt{,}}_{n}\texttt{)}$$

wobei sich zwischen den beiden Klammern gleich viele Kommata befinden. Ist diese Sprache kontextfrei?

Lösung. Diese Sprache ist kontextfrei, denn man kann diese Wörter mit folgender Grammatik erzeugen

$$\begin{aligned} S &\to \text{‘}\texttt{(}\text{’}\ V\ \text{‘}\texttt{)}\text{’} \\ V &\to \text{‘}\texttt{,}\text{’}\ V\ \text{‘}\texttt{,}\text{’} \\ &\to \text{‘}\texttt{)(}\text{’} \end{aligned}$$

○

5.4. Betrachten Sie die kontextfreie Grammatik

$$\begin{aligned} R &\to \texttt{'a'} \mid \texttt{'b'} \mid \texttt{'c'} \mid \dots \mid \texttt{'z'} \mid \texttt{'.'} \\ &\to R\ \texttt{'*'} \\ &\to R\ \texttt{'|'}\ R \\ &\to RR \\ &\to \texttt{'('}\ R\ \texttt{')'} \end{aligned}$$

über dem Alphabet bestehend aus allen Zeichen, die in der Grammatik durch einfache Anführungszeichen als Terminalsymbole gekennzeichnet wurden. Finden Sie die Ableitung für die folgenden Wörter mit dieser Grammatik:

a) `.*`

b) `(a|b)`

c) `(ab)*|(cd)*`

Lösung. a) $R \to R\texttt{*} \to \texttt{.*}$

b) $R \to \texttt{(}R\texttt{)} \to \texttt{(}R\texttt{|}R\texttt{)} \to \texttt{(a|}R\texttt{)} \to \texttt{(a|b)}$

c)

$$\begin{aligned} R &\to R\,|\,R \to R^*\,|\,R \to R^*\,|\,R^* \to (R)^*\,|\,R^* \to (R)^*\,|\,(R)^* \to (RR)^*\,|\,(R)^* \\ &\to (RR)^*\,|\,(RR)^* \to (\mathtt{a}R)^*\,|\,(RR)^* \to (\mathtt{ab})^*\,|\,(RR)^* \to (\mathtt{ab})^*\,|\,(\mathtt{c}R)^* \\ &\to (\mathtt{ab})^*\,|\,(\mathtt{cd})^* \end{aligned}$$

○

Diskussion. Diese Grammatik beschreibt eine Teilmenge der regulären Ausdrücke.

5.5. Die kontextfreie Grammatik

$$\begin{aligned} S &\to A \\ &\to B\mathtt{c} \\ A &\to AB \\ &\to ABC \\ &\to \mathtt{a} \\ B &\to \mathtt{b} \\ C &\to \mathtt{cc} \end{aligned}$$

hat nicht Chomsky-Normalform. Finden Sie eine äquivalente Grammatik in Chomsky-Normalform.

Lösung. Die Ausgangsgrammatik ist

$$\begin{aligned} S &\to A \mid B\mathtt{c} \\ A &\to AB \mid ABC \mid \mathtt{a} \\ B &\to \mathtt{b} \\ C &\to \mathtt{cc} \end{aligned}$$

Elimination der Unit rule $S \to A$:

$$\begin{aligned} S &\to B\mathtt{c} \mid AB \mid ABC \mid \mathtt{a} \\ A &\to AB \mid ABC \mid \mathtt{a} \\ B &\to \mathtt{b} \\ C &\to \mathtt{cc} \end{aligned}$$

Ersetzung der ABC-Regeln mithilfe einer zusätzlichen Variablen U:

$$\begin{aligned} S &\to B\mathtt{c} \mid AB \mid AU \mid \mathtt{a} \\ A &\to AB \mid AU \mid \mathtt{a} \\ U &\to BC \\ B &\to \mathtt{b} \\ C &\to \mathtt{cc} \end{aligned}$$

Elimination der Terminal-Symbole

$$S \to BC' \mid AB \mid AU \mid \mathtt{a}$$

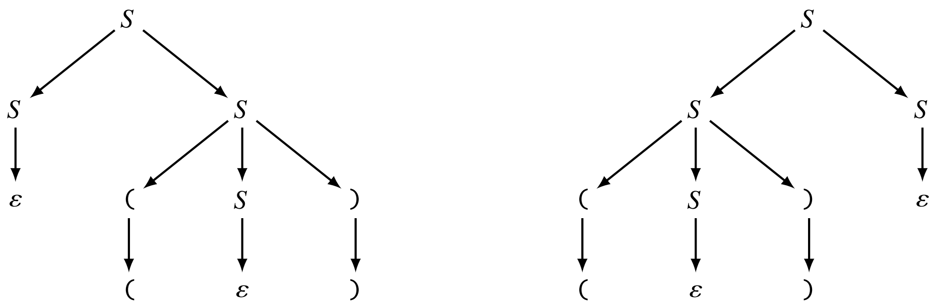
$$
\begin{aligned}
A &\to AB \mid AU \mid \mathtt{a} \\
U &\to BC \\
B &\to \mathtt{b} \\
C &\to C'C' \\
C' &\to \mathtt{c}
\end{aligned}
$$

Diese Grammatik hat offenbar Chomsky-Normalform. ○

5.6. Für die Sprache der korrekt geschachtelten Klammerausdrücke wurde die Grammatik

$$
S \to \mathtt{(}S\mathtt{)} \mid SS \mid \varepsilon \tag{1}
$$

gefunden, die sich jedoch als nicht eindeutig herausgestellt hat. Das Wort `()` hat zum Beispiel die beiden völlig verschiedenen Parse-Trees



Betrachten Sie als Alternative die Grammatik

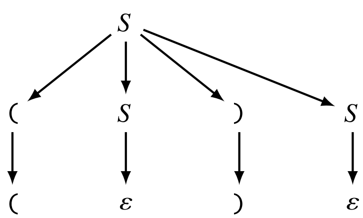
$$
\begin{aligned}
S &\to \mathtt{(}S\mathtt{)}S \\
&\to \varepsilon
\end{aligned}
\tag{2}
$$

a) Finden Sie den Parse-Tree von `()` in der Grammatik (2).

b) Ist die Grammatik (2) eindeutig?

c) Finden Sie die Chomsky-Normalform dieser Grammatik.

Lösung. a) Der Parse Tree ist



b) Die Grammatik ist eindeutig, weil für jede öffnende Klammer die erste Regel genau einmal angewendet werden muss. Die Grammatik (1) dagegen lässt immer die Option offen, die Regel $S \to SS$ anzuwenden, was zur Zweideutigkeit führt.

c) Zunächst muss sichergestellt werden, dass die Startvariable auf der rechten Seite nicht vorkommt:

$$\begin{aligned} S_0 &\to S \\ S &\to (S)S \mid \varepsilon \end{aligned}$$

Jetzt müssen ε-Regeln entfernt werden:

$$\begin{aligned} S_0 &\to S \mid \varepsilon \\ S &\to (S)S \mid ()S \mid (S) \mid () \end{aligned}$$

Jetzt müssen Unit-Rules entfernt werden:

$$\begin{aligned} S_0 &\to (S)S \mid ()S \mid (S) \mid () \mid \varepsilon \\ S &\to (S)S \mid ()S \mid (S) \mid () \end{aligned}$$

Jetzt müssen nur noch die verbleibenden Sequenzen aufgeteilt werden:

$$\begin{aligned} S_0 &\to AX \mid AY \mid AZ \mid AB \mid \varepsilon \\ S &\to AX \mid AY \mid AZ \mid AB \\ A &\to (\\ B &\to) \\ X &\to SY \\ Y &\to BS \\ Z &\to SB \end{aligned}$$

Damit ist Chomsky-Normalform erreicht. ○