

Aufgaben zu den Vorlesungen “Theoretische Informatik III” und “Compilerbau”

Aufgabe 1: Ein Kassenzettel bestehe aus Zeilen, die in etwa wie folgt aussehen:

<i>Tullamore Dew:</i>	91.50 Euro
<i>Jim Beam:</i>	17.34 Franken
<i>Stroh Rum:</i>	22.33 Euro

Die Preise sind also teilweise in Euro, teilweise in Schweizer Franken angegeben. Entwickeln Sie ein *JFlex*-Programm, das einen Kassenzettel einliest und anschließend den Gesamtpreis in Euro ausgibt. Nehmen Sie dabei an, dass ein Euro einen Wert von 1,20 Franken hat.

Aufgabe 2:

- (a) Geben Sie einen deterministischen endlichen Automaten F an, so dass $L(F)$ aus genau den Wörtern der Sprache $\{a, b\}^*$ besteht, bei denen auf jedes “a” mindestens ein “b” folgt.
- (b) Geben Sie einen regulären Ausdruck an, der diese Sprache beschreibt.
- (c) Berechnen Sie, ausgehend von dem unter (b) angegebenen regulären Ausdruck, einen nicht-deterministischen endlichen Automaten, der dieselbe Sprache erkennt.
Benutzen Sie dabei das in der Vorlesung diskutierte Verfahren.
- (d) Überführen Sie den unter (c) angegebenen Automaten in einen deterministischen endlichen Automaten.
Benutzen Sie dabei das in der Vorlesung diskutierte Verfahren.
- (e) Minimieren Sie die Anzahl der Zustände des in (d) berechneten Automaten.
Benutzen Sie dabei das in der Vorlesung diskutierte Verfahren.

Aufgabe 3: Die Sprache L_P beinhaltet alle Wörter aus der Sprache $\{a, b\}^*$, die Palindrome sind. Definieren wir für einen String $w = c_1c_2 \cdots c_{n-1}c_n$ den String w_r (gelesen: w rückwärts) durch

$$w^r := c_n c_{n-1} \cdots c_2 c_1,$$

so kann die Sprache L_P durch

$$L_P := \{w \in \{a, b\}^* \mid w = w^r\}$$

definiert werden.

- (a) Zeigen Sie mit Hilfe des Pumping-Lemmas für reguläre Sprachen, dass die Sprache L_P der Palindrome keine reguläre Sprache ist.
- (b) Überprüfen Sie, ob die Sprache L_P eine kontextfreie Sprache ist und beweisen Sie Ihre Behauptung.

Aufgabe 4: Es sei $\Sigma = \{a, b, c\}$.

- (a) Die Sprache L_1 werde durch die Gleichung

$$L_1 := \{a^k b^l c^{l+k} \in \Sigma^* \mid k, l \in \mathbb{N}\}$$

definiert. Überprüfen Sie, ob die Sprache L_1 regulär ist und beweisen Sie Ihre Behauptung.

- (b) Überprüfen Sie, ob die oben definierte Sprache L_1 kontextfrei ist und beweisen Sie Ihre Behauptung.

- (c) Die Sprache L_2 werde durch die Gleichung

$$L_2 := \{a^k b^l c^{l \cdot k} \in \Sigma^* \mid k, l \in \mathbb{N}\}$$

definiert. Überprüfen Sie, ob die Sprache L_2 regulär ist und beweisen Sie Ihre Behauptung.

- (d) Überprüfen Sie, ob die oben definierte Sprache L_2 kontextfrei ist und beweisen Sie Ihre Behauptung.

Aufgabe 5: Präfix-Ausdrücke sind Ausdrücke, in denen die arithmetischen Operatoren “+”, “-”, “*” und “/” als zweistellige Präfix-Operatoren benutzt werden. Außer den genannten Operatoren sollen die Ausdrücke nur natürliche Zahlen enthalten. Die Operatoren stehen immer vor ihren Argumenten, die natürlich selber komplexe Präfix-Ausdrücke sein können. Beispielsweise ist

$$+ + 2 * 3 4 - 1 2$$

ein Präfix-Ausdruck, der den selben Wert hat wie der Term

$$(2 + (3 * 4)) + (1 - 2).$$

- (a) Geben Sie eine **Ebnf**-Grammatik für Präfix-Ausdrücke an.
- (b) Geben Sie eine ANTLR-Grammatik an, mit deren Hilfe Sie einen Präfix-Ausdruck auswerten können.
- (c) Geben Sie eine JAVACUP-Grammatik an, mit deren Hilfe Sie einen Präfix-Ausdruck auswerten können. Zusätzlich sollen Sie mit Hilfe von *JFlex* einen Scanner entwickeln, der die in der JAVACUP-Grammatik verwendeten Nicht-Terminale erzeugt.

Aufgabe 6: Betrachten Sie die folgende Sprache:

$$L = \{a^k b^l \mid k, l \in \mathbb{N} \wedge k \neq l\}$$

- (a) Zeigen Sie, dass diese Sprache nicht regulär ist.

Hinweis: Benutzen Sie das Theorem von Nerode.

- (b) Zeigen Sie, dass diese Sprache kontextfrei ist.

Aufgabe 7: Die Grammatik $G = \langle \{A, B\}, \{ \text{"u"}, \text{"x"}, \text{"y"}, \text{"z"} \}, R, A \rangle$ habe die folgenden Regeln:

$$\begin{array}{lcl} A & \rightarrow & B \text{"x"} \\ & | & \text{"y"} B \text{"z"} \\ & | & \text{"u"} \text{"z"} \\ & | & \text{"y"} \text{"u"} \text{"x"} \\ B & \rightarrow & \text{"u"} \end{array}$$

Bearbeiten Sie die folgenden Teilaufgaben:

- Überprüfen Sie, ob diese Grammatik eine LL(1)-Grammatik ist und begründen Sie Ihre Antwort.
- Überprüfen Sie, ob diese Grammatik eine LR-Grammatik ist und begründen Sie Ihre Antwort.

Aufgabe 8: Die Grammatik $G = \langle \{S\}, \{+, -, \mathbf{a}\}, R, S \rangle$ habe die folgenden Regeln:

$$S \rightarrow S S + \mid S S - \mid \mathbf{a}.$$

- Berechnen Sie die Mengen $First(S)$ und $Follow(S)$.
- Berechnen Sie die Menge der SLR-Zustände für diese Grammatik.
- Berechnen Sie die Funktionen $action()$ und $goto()$ für diese Grammatik und stellen Sie das Ergebnis in einer Tabelle dar.
- Untersuchen Sie, ob diese Grammatik mehrdeutig ist.

Aufgabe 9: Nehmen Sie an, dass die im Skript eingeführte Sprache *Integer-C* um eine **do-while**-Schleife erweitert werden soll, deren Syntax durch die folgende Grammatik-Regel gegeben ist:

$$statement \rightarrow \text{"do"} \text{ statement } \text{"while"} \text{ "(" } boolExpr \text{ ")" } .$$

Die Semantik dieses Konstruktes soll mit der Semantik des entsprechenden Konstruktes in der Sprache *C* übereinstimmen. Geben Sie eine Gleichung an, die beschreibt, wie eine **do-while**-Schleife in *Java-Assembler* übersetzt werden kann.