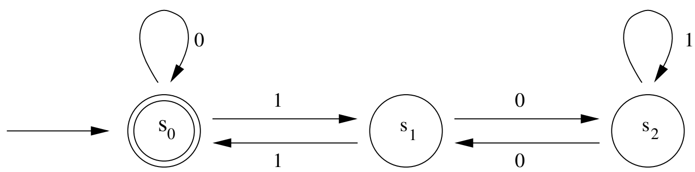
Technische Universität München
Fakultät für Informatik
Prof. Dr. M. Broy

SS 2003
Übungsblatt 3
2. Mai 2003

Übungen zur Vorlesung Einführung in die Informatik IV

Aufgabe 9 (H) Endliche Automaten und reguläre Grammatiken

(a) Gegeben sei der folgende endliche Automat A:



Geben Sie eine reduktive linkslineare Grammatik G an, so dass $L(G) = L(A) \backslash \varepsilon$.

(b) Gegeben sei folgende Grammatik $G = (T, N, \to, Z)$ mit

$$
\begin{aligned}
T &= \{0,1\} \\
N &= \{s_0, \ldots, s_2, Z\} \\
\to &= \{0 \to s_0,\ 1 \to s_1,\ s_0 0 \to s_0,\ s_0 1 \to s_1,\ s_1 0 \to s_1,\ s_1 1 \to s_2,\ s_2 0 \to s_2,\ s_2 \to Z\}
\end{aligned}
$$

Geben Sie einen endlichen Automaten A an, so dass $L(G) = L(A)$.

Aufgabe 10 Endliche Automaten und reguläre Ausdrücke

(a) Vereinfachen Sie die folgenden regulären Ausdrücke:

(i) $\varepsilon | s | ss^* s$

(ii) $t | ss^* t$

(iii) $t | ts^* s$

(iv) $(ts^* s)\{\}$

(v) $\{\} | (ts^* s)$

(b) Gegeben sei der endliche Automat A aus Aufgabe 9 (a). Geben Sie einen regulären Ausdruck an, der von diesem Automaten akzeptiert wird. Vereinfachen sie den regulären Ausdruck so weit wie möglich!

(c) **(Optional)** Schreiben Sie ein (Gofer-)Programm, das zu einem Automaten den äquivalenten regulären Ausdruck berechnet.

Aufgabe 11 **Reguläre Ausdrücke und Automaten**

In der vorhergehenden Aufgabe wurde ein deterministischer Automat in einen regulären Ausdruck umgewandelt. Nun soll die Umkehrung diskutiert werden. Diese Umwandlung erfolgt in mehreren Teilschritten. Gegeben sei der reguläre Ausdruck:

$$[0|1[01^*0]^*1]^*.$$

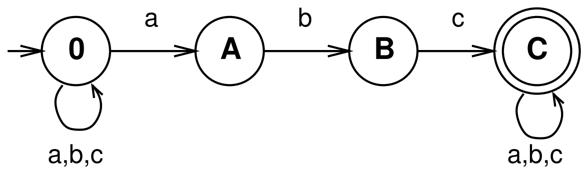
(a) Wandeln Sie diesen regulären Ausdruck in einen nichtdeterministischen endlichen Automaten mit ε- Übergängen um.

(b) Eliminieren Sie die ε-Übergänge!

(c) Minimalisieren Sie den Automaten.

Aufgabe 12 **NEA – DEA**

Gegeben sei folgender nichtdeterministische endliche Automat:



(a) Geben Sie einen deterministischen endlichen Automaten mit gleichem Sprachschatz an! Verwenden Sie dazu das aus der Vorlesung bekannte Potenzmengenverfahren!

(b) (**H**) Minimalisieren Sie den Automaten aus Teilaufgabe a).

Aufgabe 13 **(P)** **Reguläre Grammatik, endlicher Automat**

In dieser Aufgabe soll ein Algorithmus entwickelt werden, der aus einer beliebigen regulären rechtslinearen Grammatik einen endlichen Automaten generiert. Dieser Automat akzeptiert genau die von der Grammatik erzeugte Sprache und kann somit in bekannter Weise zur Verarbeitung strukturierter Informationen eingesetzt werden.

(a) Entwickeln Sie in Java eine allgemein verwendbare Klasse `Automaton`, die beliebige endliche Automaten auf geeignete Weise repräsentiert. Beachten Sie hierbei die Besonderheiten einer möglicherweise nichtdeterministischen Zustandsübergangs*relation* sowie die gebotene Flexibilität gegenüber späteren Änderungen der Struktur zur Laufzeit. Implementieren Sie typische Funktionalität, wie Erzeugen der Zustände und Übergänge oder Ausgabe des aktuellen Zustands sowie der bei Fortschaltung erhaltenen Folgezustände.

(b) Definieren Sie ein geeignetes Format, um die Produktionen einer rechtslinearen regulären Grammatik in einer Datei abzulegen, beispielsweise zeilenweise in der Form[1]
<nonterminal> -> <epsilon> | terminal | terminal <nonterminal>
Implementieren Sie eine Methode `readGrammar()` der Klasse `Automaton`, die eine derartige Datei einliest und die erforderlichen Zustände und Zustandsübergänge erzeugt. Dokumentieren Sie die korrekte Umsetzung durch mindestens 3 verschiedene Testfälle.

[1]Das Axiom der Grammatik kann hierbei durch ein ausgezeichnetes Symbol <S> dargestellt werden.

Hinweis: Die Java-Klasse `java.util.HashMap` bzw. die von ihr implementierte Schnittstelle `java.util.Map` ermöglicht die einfache Zuordnung zwischen Zeichenketten und Objektinstanzen zur Laufzeit. Diese Funktionalität kann bei Konstruktion des Automaten aus einer Grammatik vorteilhaft eingesetzt werden. Darüber hinaus wird im WWW eine umfassende Lösungsvorlage bereitgestellt, die Sie als Grundlage einer eigenen Lösung verwenden können.