Lösung zur Klausur „Grundlagen der Theoretischen Informatik“ im WiSe 2003/2004

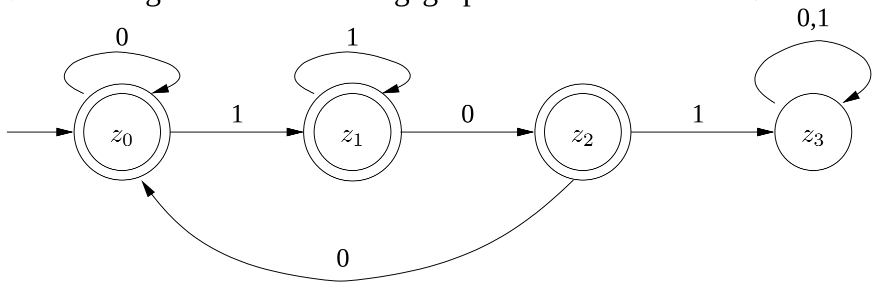
1. Geben Sie einen deterministischen endlichen Automaten an, der die Sprache aller Wörter über dem Alphabet $\{0,1\}$ akzeptiert, die nicht das Teilwort 101 enthalten, also die Sprache

$$L_1 = \left\{ w \in \{0,1\}^* \,\middle|\, \text{es gibt keine Wörter } u, v \in \{0,1\}^* \text{ sodass } w = u101v \right\}.$$

Lösung: Folgender DEA akzeptiert L_1:

$$M = (\{z_0, z_1, z_2, z_3\}, \{0,1\}, \delta, \{z_0, z_1, z_2\}),$$

wobei δ durch folgenden Überführungsgraphen beschrieben wird:



Der Automat sucht nach der Folge 101 in einem Wort. Solange er sie noch nicht gefunden hat, befindet er sich in einem der akzeptierenden Zustände z_0, z_1 oder z_2. Dabei gilt: M ist in z_0, wenn die beiden letzten Zeichen 00 waren oder noch keine 1 gelesen wurde, er ist in z_1 wenn das letzte Zeichen eine 1 war, und er ist in z_2, wenn die beiden letzten Zeichen 10 waren.

Findet M die Folge 101, so geht er in den ablehnenden Zustand z_3 über. Diesen Zustand verlässt M bis zum Ende des Wortes dann nicht mehr. $\square$

2. Geben Sie eine kontextfreie Grammatik an, die folgende Sprache L_2 erzeugt:

$$L_2 = \left\{ a^i b^j c^k \,\middle|\, i = j \text{ oder } j = k;\ i, j, k \geq 0 \right\}.$$

Lösung:

Folgende Grammatik erzeugt L_2:

$$G = (\{S, A, C, X, Y\}, \{a, b, c\}, P, S),$$

wobei R folgendermaßen definiert ist:

$$\begin{array}{rclccccccccc} P & = & \{ & S & \rightarrow & AX, & S & \rightarrow & A, & S & \rightarrow & X, \\ & & & A & \rightarrow & aA, & A & \rightarrow & a, & & & \\ & & & X & \rightarrow & bXc, & X & \rightarrow & bc, & & & \\ & & & S & \rightarrow & YC, & S & \rightarrow & Y, & S & \rightarrow & C, \\ & & & Y & \rightarrow & aYb, & Y & \rightarrow & ab, & & & \\ & & & C & \rightarrow & Cc, & C & \rightarrow & c, & & & \\ & & & S & \rightarrow & \varepsilon & \} & & & & & \end{array}$$

G ist kontextfrei, da auf der linken Seite der Ableitungsregeln immer nur eine einzige Variable steht, $S \rightarrow \varepsilon$ die einzige ε-Regel ist und S nie auf der rechten Seite einer Regel vorkommt.

G muss genau die Wörter der Form $a^n b^m c^m$ mit $n, m \geq 0$ und die Wörter $a^n b^n c^m$ mit $n, m \geq 0$ erzeugen. Wir zeigen nun, dass dies der Fall ist:

Es gilt: Aus der Variablen A kann ein Block a^n mit $n \geq 1$ erzeugt werden; aus X kann ein Block $b^n c^n$ mit $n \geq 1$ erzeugt werden, aus Y kann ein Block $a^n b^n$ mit $n \geq 1$ erzeugt werden und aus C kann ein Block c^n mit $n \geq 1$ erzeugt werden.

Also kann aus AX ein Block $a^n b^m c^m$ mit $n, m \geq 1$ erzeugt werden, und aus YC kann ein Block $a^n b^n c^m$ mit $n, m \geq 1$ erzeugt werden.

Nun müssen nur noch Wörter erzeugt werden, bei denen eines der n oder m (oder beide) gleich 0 sind. Dies wird mit den Regeln $S \rightarrow A$, $S \rightarrow X$, $S \rightarrow Y$, $S \rightarrow C$ und $S \rightarrow \varepsilon$ erreicht.

$\square$

3. Zeigen Sie, dass folgende Sprache L_3 nicht regulär ist:

$$L_3 = \{\, 0^i 1^j \mid i > j \geq 0 \,\}.$$

Lösung:

Beweis mit Pumping-Lemma:

Angenommen, die Sprache L_3 wäre regulär. Dann gibt es eine Zahl n, sodass sich alle Wörter $x \in L$ mit $|x| \geq n$ zerlegen lassen in $x = uvw$, sodass folgende Eigenschaften gelten:

(i) $|v| > 0$,

(ii) $|uv| \leq n$,

(iii) $uv^iw \in L$ für alle $i \geq 0$.

Sei nun n beliebig. Wähle $x = 0^{n+1}1^n$. Es gilt $|x| \geq n$. Sei $x = uvw$ eine beliebige Zerlegung von x, die (i) und (ii) erfüllt. Wegen (ii) folgt, dass uv nur aus Nullen besteht. Sei $l = |v|$. Wegen (i) folgt, dass $l > 0$. Nun gilt: $uv^0w = uw = 0^{n+1-l}1^n \notin L_3$, da $l \geq 1$.

Dies ist ein Widerspruch zu (iii); also kann die Sprache L_3 nicht regulär sein.

$\square$

4. Beweisen Sie: Sind L und L' reguläre Sprachen, dann ist $L \cup L'$ wieder regulär.

Lösung:

Seien L und L' regulär. Dann gibt es zwei reguläre Grammatiken $G = (N, \Sigma, R, S)$ und $G' = (N', \Sigma', R', S')$ mit $L(G) = L$ und $L(G') = L'$. Wir können O.B.d.A. annehmen, dass $N \cap N' = \emptyset$ und damit auch $R \cap R' = \emptyset$ gelten.

Wir wollen nun eine neue reguläre Grammatik G'' angeben, die $L \cup L'$ erzeugt. Ihre Ableitungen geschehen nach folgendem Prinzip:

Sie enthält alle Variablen und (fast) alle Regeln von G und G'. Außerdem besitzt sie noch einen Startzustand S'', der weder in N noch in N' enthalten ist. Vom Startzustand S'' entscheidet sich G'' für eine der Grammatiken G oder G' und simuliert diese dann. So können alle von G und G' erzeugten Wörter auch von G'' erzeugt werden.

Formal:

$$G'' = (N \cup N' \cup \{S''\}, \Sigma \cup \Sigma', R'', S''),$$

wobei für R'' gilt:

$$R'' = (R \setminus \{S \to \varepsilon\}) \cup (R' \setminus \{S' \to \varepsilon\}) \cup \{\, S'' \to w \mid S \to w \in R \text{ oder } S' \to w \in R' \,\}$$

Die neue Grammatik ist regulär, denn sie enthält nur Regeln der regulären Grammatiken G und G' und Regeln, die durch Austausch einer Variablen aus Regeln in G oder G' entstanden sind; außerdem ist $S'' \to \varepsilon$ die einzige mögliche ε-Regel und S'' kommt niemals auf der rechten Seite einer Regel vor.

G'' erzeugt alle Worte aus L, denn falls $x \in L$, dann gilt $S \Rightarrow_G y \Rightarrow_G^* x$ und damit $S'' \Rightarrow_{G''} y \Rightarrow_{G''}^* x$. Mit einem analogen Argument gilt auch $L' \subseteq L(G'')$.

Es werden nur Wörter aus $L \cup L'$ erzeugt, da sich G'' im ersten Erzeugungschritt für eine Grammatik G oder G' entscheidet und dann, aufgrund der disjunkten Variablenmengen, nur noch Regeln dieser Grammatik benutzt.

$\square$

5. Die Firma Big Computer Corp. möchte die sinkenden Marktanteile durch die Herstellung einer High-Tech-Version der Turing-Maschine, GPTM genannt, steigern. Eine GPTM enthält alle Komponenten einer gewöhnlichen Turingmaschine und ist zusätzlich mit einer *Glocke* und einer *Pfeife* ausgestattet. Jeder Zustand einer GPTM ist entweder als ein „Glockenzustand" oder als ein „Pfeifenzustand" ausgezeichnet. Die Arbeitsweise einer GPTM entspricht der einer gewöhnlichen TM mit folgender zusätzlicher Funktionalität: Immer wenn die GPTM einen Zustandsübergang durchführt, läutet sie entweder die Glocke oder bläst die Pfeife, je nach Typ des neuen Zustandes.

   Beweisen Sie, dass es unentscheidbar ist, ob eine gegebene GPTM M bei gegebener Eingabe w jemals pfeift.

   Lösung:

   Es ist zu zeigen, dass unentscheidbar ist, ob eine gegebene GPTM bei gegebener Eingabe einen Pfeifenzustand erreicht. Wir wissen, dass unentscheidbar ist, ob eine TM M bei Eingabe x einen Endzustand erreicht (allgemeines Halteproblem). Wir machen nun aus M eine GPTM M', indem wir alle Endzustände zu Pfeifenzuständen machen. Dann hält also M bei Eingabe x gdw. M' bei Eingabe x irgendwann pfeift. Also ist unentscheidbar, ob eine GPTM jemals pfeift.

   Formal: Sei $P = \{\, w\#x \mid M_w \text{ bei Eingabe } x \text{ pfeift} \,\}$. Zu zeigen ist also, dass P (das „Pfeifenproblem") unentscheidbar ist.

   Wir konstruieren dazu eine Reduktion vom allgemeinen Halteproblem H auf P; wir zeigen also $H \leq P$. Definiere die Funktion f wie folgt:

   $$f(w\#x) = w'\#x,$$

   wobei w' die Kodierung folgender GPTM M' ist: M' arbeitet genau wie die Turingmaschine M_w. Die Endzustände von M_w sind die Pfeifenzustände von M', alle anderen Zustände von M' sind Glockenzustände. Die Funktion f ist total und berechenbar.

   Es gilt: $w\#x \in H$ gdw. M_w bei Eingabe x hält
   gdw. M' bei Eingabe x pfeift irgendwann
   gdw. $w'\#x \in P$
   gdw. $f(w\#x) \in P$,

   also ist $H \leq P$, und da H unentscheidbar ist, folgt, dass P unentscheidbar ist.

   □

6. Welche der folgenden Aussagen sind korrekt? Bitte kreuzen Sie die korrekten Aussagen in den dafür vorgesehenen Kästchen an.

☐ Jede Teilmenge einer regulären Sprache ist regulär.

Begründung: Die Sprache $L_1 = \{a, b\}^*$ ist regulär (sie wird von einem DEA mit Eingabealphabet $\{a, b\}$ erkannt, der nur Endzustände besitzt). Die Sprache $L_2 = \left\{\, a^n b^n \;\middle|\; n \geq 0 \,\right\}$ ist eine Teilmenge von L_1, aber nicht regulär.

☐ Jede reguläre Sprache besitzt eine echte Teilmenge, die wieder regulär ist.

Begründung: Die leere Menge ist eine reguläre Sprache (sie wird von einem DEA, der keine Endzustände besitzt, erkannt), besitzt aber keine echte Teilmenge, also auch keine echte reguläre Teilmenge.

☒ Jede Sprache besitzt eine Teilmenge, die regulär ist.

Begründung: Die leere Menge ist regulär und Teilmenge jeder beliebigen Menge.

☒ Jede Sprache ist Teilmenge einer regulären Sprache.

Begründung: Sei L eine Sprache über dem Alphabet Σ. Dann ist Σ^* eine reguläre Sprache, und L ist Teilmenge von Σ^*.