Technische Universität München
Institut für Informatik
PD Dr. Markus Holzer
Dipl.-Inf. Michael Tautschnig

WS 2007/2008
Übungsblatt 9
Mittwoch, 19. Dezember 2007

Übung zur Vorlesung Automaten, Formale Sprachen und Berechenbarkeit

Lösungshinweise

Aufgabe 1 *Kleensche Hülle*

(i) $A^0 = \{\lambda\}$, $A^1 = \{\lambda, a\}$, $A^2 = \{\lambda, a, aa\}$, $A^3 = \{\lambda, a, aa, aaa\}$. Daher enthält A^n genau die Elemente $\lambda, a, \ldots, a^n$; A^n enthält $n+1$ Elemente (Beweis durch Induktion).

(ii) Wir zeigen durch Induktion, daß für alle $i \geq 2$ gilt, daß $A^i = A^{i-1}$. Daraus folgt sofort $A = A^*$. Der Induktionsanfang $A^2 = A$ gilt nach Voraussetzung. Nehmen wir als Induktionsannahme an, daß $A^{n-1} = A^n$. Für A^{n+1} gilt dann: $A^{n+1} = A \cdot A^n = A \cdot A^{n-1} = A^n$.

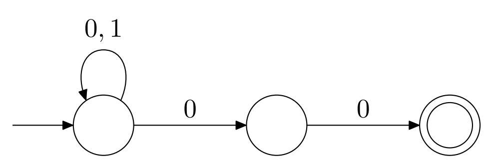
Aufgabe 2 Tunix *Pfade*

```
^([a-zA-Z0-9]+/)*[a-zA-Z0-9]+(\.[A-Z][A-Z][A-Z])?$
```

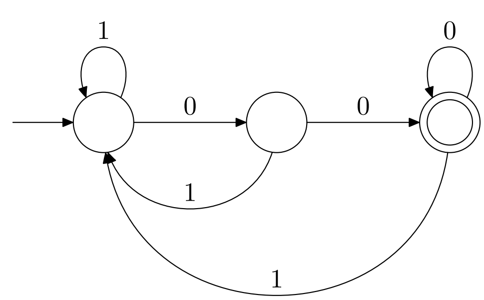
Aufgabe 3 *Determinisierung und Minimierung von endlichen Automaten*

Gute Beschreibungen des zugrundeliegenden Algorithmus findet man etwa in [2], Seite 46f oder [1], Seite 154ff.

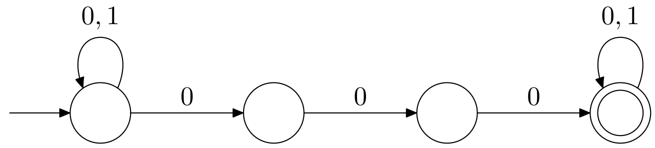
(i) Ein nichtdeterministischer Automat sieht folgendermaßen aus:



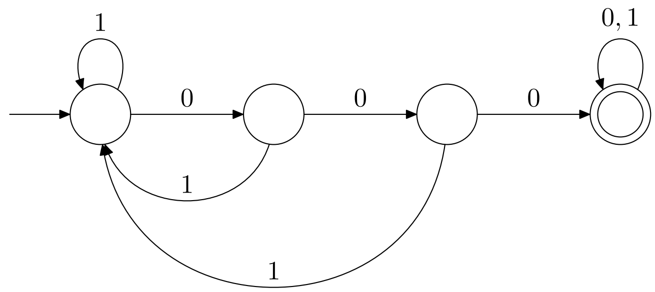
Determinisierung liefert den folgenden endlichen Automaten:



(ii) Ein nichtdeterministischer Automat sieht folgendermaßen aus:

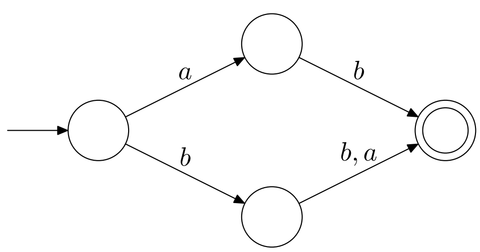


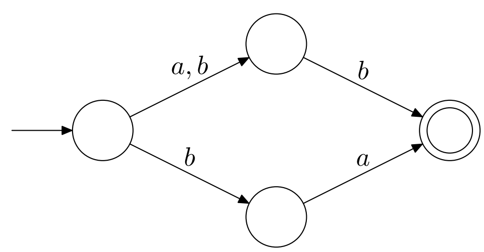
Determinisierung und Minimierung liefert den folgenden endlichen Automaten:



Aufgabe 4 *Verschiedenes*

(i) Falsch; man betrachte die Sprache $\{ab, bb, ba\}$; diese Sprache wird von zwei nicht-isomorphen nichtdeterministischen Automaten mit gleicher Zahl von Zuständen akzeptiert:





(ii) Falsch; das angegebene Problem reduziert sich auf das Erreichbarkeitsproblem in einem Graphen (ist von einem gegebenen Zustand des Automaten ein Endzustand erreichbar). Dieses Problem ist sicher entscheidbar.

(iii) Man betrachte Automaten, die die Sprachen $L_1 = (a^m)^*$ und $L_2 = (a^n)^*$ (wobei m und n beliebige verschiedene Primzahlen sind) akzeptieren. Klarerweise benötigen Automaten, die diese Sprachen akzeptieren $O(m)$ bzw. $O(n)$ Zustände, da diese Automaten bis m bzw. n zählen müssen. Nun gilt aber $L_1 \cap L_2 = (a^{mn})^*$; jeder Automat, der diese Sprache akzeptiert muss bis mn zählen können und benötigt daher $O(mn)$ Zustände.

Literatur

[1] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, New York, second edition, 2001.

[2] U. Schöning. *Theoretische Informatik - kurzgefasst.* Spektrum Akademischer Verlag, Heidelberg, Berlin; Germany, 4 edition, 2001.