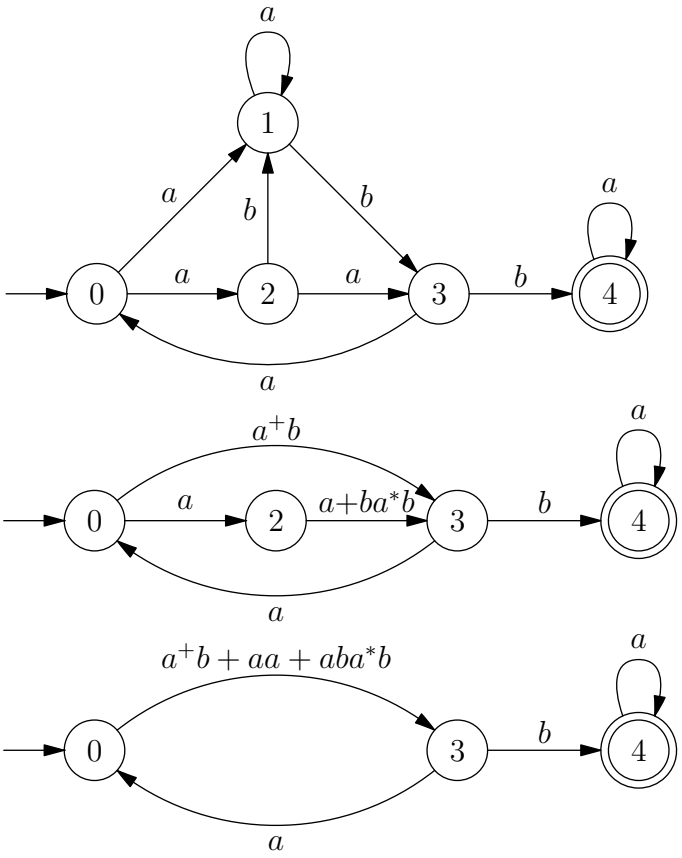
Formale Systeme, Automaten und Prozesse

Lösungsvorschlag

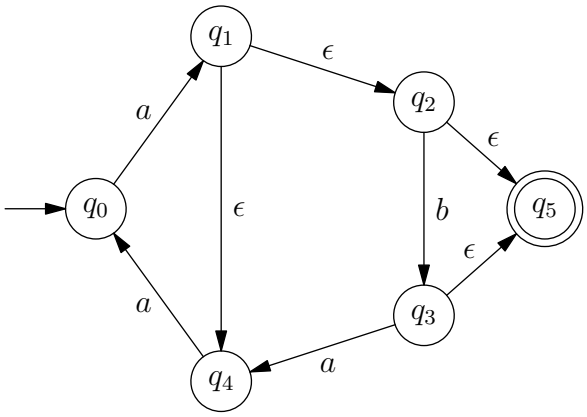
Aufgabe 1 (10 Punkte)
Wir können das Verfahren aus der Vorlesung verwenden, um die Zustände 1 und dann 2 zu eliminieren. (Die umgekehrte Reihenfolge ist vielleicht sogar noch besser.)



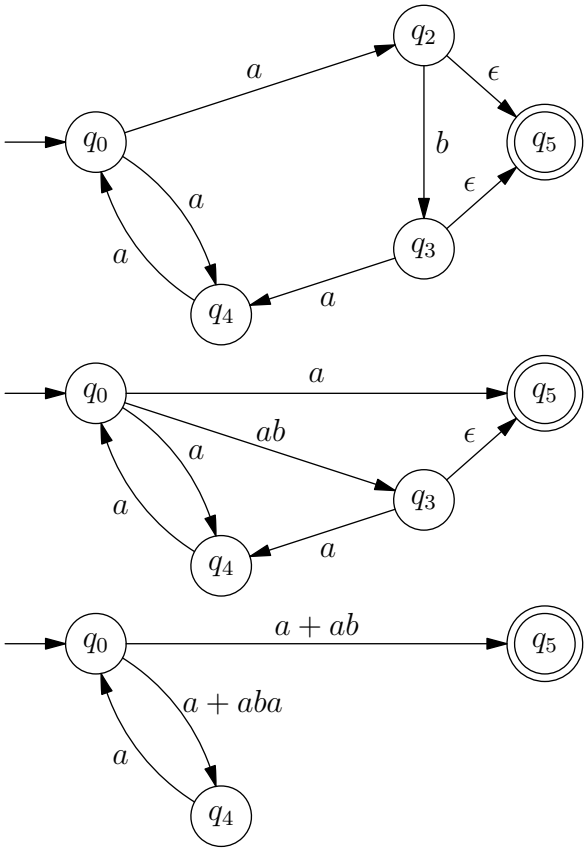
Nun können wir auch noch den Zustand 3 eliminieren oder sofort den regulären Ausdruck ablesen:

$$(a^+ba + aaa + aba^*ba)^*(a^+b + aa + aba^*b)ba^*$$

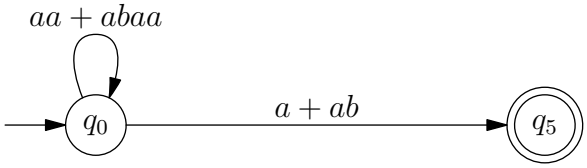
Beim zweiten Automaten stehen wir zunächst vor der Schwierigkeit, daß es zwei Endzustände gibt. Verwenden wir also erst einmal die Standardkonstruktion, welche dieses Problem löst.



Nun eliminieren wir nacheinander die Zustände q_1, q_2 und q_3:



Jetzt können wir den regulären Ausdruck schon bequem ablesen, aber zur Abwechslung eliminieren wir auch noch den Zustand q_4 um einmal stur beim Standardverfahren zu bleiben:



Jetzt ergibt sich der reguläre Ausdruck

$$(aa + abaa)^*(a + ab)$$

und wir sind fertig.

Typische Fehler: Der erste reguläre Ausdruck wurde in der Regel korrekt berechnet. Allerdings gab es hier teilweise Probleme den drittletzten Zustand zu eliminieren. Laut

dem üblichen Vorgehen erzeugt dies eine Schleife an Zustand 0 und eine Kante von 0 nach 4. Oft wurde jedoch die Schleife weggelassen (was möglich ist), dafür aber dann die kompliziertere Beschriftung der Kante von 0 nach 4 falsch angeben.

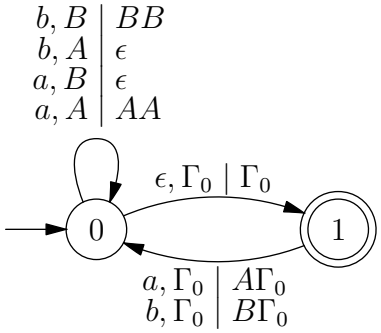
Beim zweiten Automat gab es große Probleme mit den zwei Endzuständen, da das Verfahren aus der Vorlesung nicht angewendet wurde. Oft wurden stattdessen Endzustände einfach eliminiert, beide Endzustände verschmolzen oder versucht, den Ausdruck an einem Automaten mit drei Zuständen abzulesen. In vielen Fällen führte dies zu falschen Lösungen.

Ein weiterer häufiger Fehler trat beim Entfernen der ϵ-Übergänge auf. Dieses Entfernen ist zwar nicht nötig, wurde aber dennoch oft angewendet. Allerdings wurden dabei häufig Kanten vergessen, zum Beispiel die Transition $(1, b, 3)$.

Äußerst umständlich war auch die unnötige Methode, die NFAs vorher mit der Potenzmengenkonstruktion zeitraubend in einen DFA zu überführen, wobei sich häufig Fehler einschlichen.

Aufgabe 2 (10 Punkte)

Wir benötigen einen Zustand 0, in dem wir den Keller benutzen, um die gelesen a's und b's zu zählen. Sobald der Zähler ausgeglichen ist, also nur Γ_0 auf dem Keller liegt, wechseln wir per ϵ-Transition in den akzeptierenden Zustand 1. Sobald in Zustand 1 ein Zeichen gelesen wird, wechseln wir wieder in Zustand 0 um weiter zu zählen. Man beachte das zu jedem beliebigen Zeitpunkt nur A's oder B's auf dem Keller liegen können, nie aber sowohl A's als auch B's.



Typische Fehler:

Sehr häufig wurde ein Automat angegeben, der nicht deterministisch ist. So gab es oft aus einem Zustand sowohl Transitionen der Form $a, \Gamma_0 \mid A\Gamma_0$ (oft auch in der Form $a, X \mid AX$) als auch gleichzeitig die Transition $\epsilon, \Gamma_0 \mid \Gamma_0$. Der Automat entscheidet also nichtdeterministisch ob er ein Zeichen liest oder die ϵ-Transition nutzt.

Dies wurde in der Regel genutzt um das Ende des Wortes zu raten, was deterministisch natürlich nicht möglich ist.

Weiterhin wurde oft ein Automat konstruiert, der mit leerem Keller akzeptiert. In der Regel wurde hier die Akzeptanzbedingung falsch angewendet, da der Automat sofort hält, wenn der Keller leer ist und nicht erst dann, wenn der Keller leer ist und gleichzeitig das Wort zu Ende ist. Auch dieser Ansatz entspricht im wesentlichen dem nichtdeterministischen Raten des Wortendes.

Übrigens ist es unmöglich, diese Sprache mit einem deterministischen Kellerautomaten zu erkennen, der mit leerem Keller akzeptiert, da nach gelesenem Präfix ab nicht klar ist, ob der Automat akzeptieren darf oder nicht.

Aufgabe 3 (10 Punkte)
Wir möchten gerne nachweisen, daß

$$\{\, u\$v\$w \mid u, v, w \in \{a, b\}^* \text{ und } (u = v \text{ oder } u = w) \,\}$$

nicht kontextfrei ist, indem wir zeigen, daß Bob eine Gewinnstrategie beim Pumping-Lemma-Spiel besitzt.

1. Alice wählt also eine beliebige Zahl n
2. Wir lassen Bob das Wort $z = a^n b^n \$ a^n b^n \$$ wählen, das sicher in der Sprache enthalten ist. Bisher hat also Bob nicht verloren.
3. Jetzt wählt Alice Wörter u, v, w, x, y mit $z = uvwxy$, $|vwx| < n$ und $|vx| > 0$.
4. Nun wählt Bob einfach $i = 0$ und wir müssen nachweisen, daß $uv^0wx^0y = uwy$ nicht in der gegebenen Sprache enthalten ist. Dann hat Bob nach den Regeln des Spiels gewonnen.

   Führen wir hierzu eine Fallunterscheidung durch.

   a) Falls vx ein \$ enthält, dann kann uwy nicht zwei \$ enthalten und gehört nicht zur gegebenen Sprache.

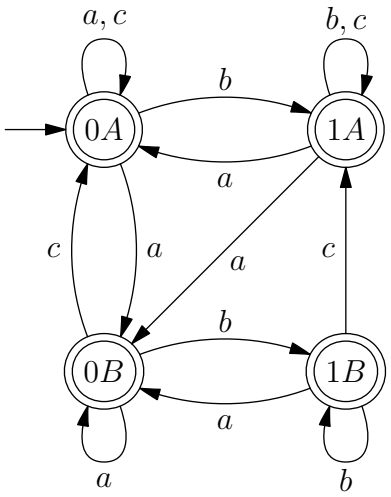
   b) Falls vx kein \$ enthält und vwx als Unterwort von z komplett vor dem ersten \$ in $a^n b^n \$ a^n b^n \$$ liegt, dann ist uv^0wx^0y von der Form $a^j b^k \$ a^n b^n \$$ mit $j < n$ oder $k < n$ und gehört so nicht zur gegebenen Sprache. Ähnlich verhält sich der Fall, daß vwx komplett hinter dem ersten \$ in z liegt.

   c) Falls vx kein \$ enthält und w das erste \$ in z enthält, dann ist $uv^0wx^0y = a^n b^j \$ a^k b^n$ mit $j < n$ oder $k < n$ und gehört so nicht zur Sprache. Falls w das zweite \$ in z enthält, dann ist $x = \epsilon$ und $uv^0wx^0y = a^n b^n \$ a^n b^{n-|v|} \$$ und wieder hat Bob gewonnen.

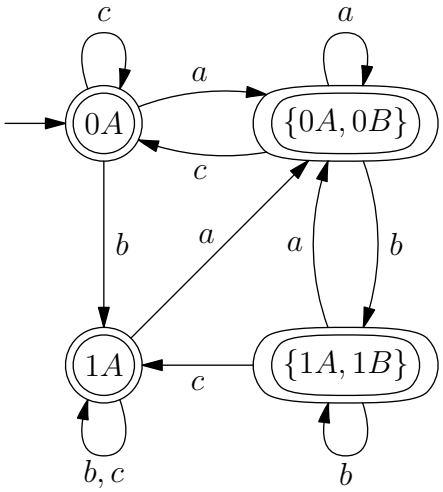
Typische Fehler:
Zunächst schien die Definition der Sprache Schwierigkeiten zu bereiten. So haben viele in ihrer Gewinnstrategie für Bob eine Fallunterscheidung nach den Fällen $u = v$ oder $u = w$ vorgenommen und hierbei die Strategie für Bob (Wahl eines Wortes $z \in L$) mit dem Kriterium für die Mitgliedschaft von Wörtern in der Sprache vermischt. Noch problematischer waren die Versuche, Bob das Wort $z = u^n \$ v^n \$ w^n$ oder ähnliche Konstrukte wählen zu lassen, ohne anzugeben, was u, v, w eigentlich sind. Wegen $u, v, w \notin \{a, b\}$ ist schon $z \notin L$ und damit die Strategie für Bob zum Scheitern verurteilt. Selbst wenn wir annehmen, daß $u, v, w \in \{a, b\}$ gemeint war, lassen sich leicht Gegenbeispiele finden.
Eher klassisch sind die Verständnis- bzw. Anwendungsfehler des Pumpinglemmas: Weder führt gleichzeitiges Argumentieren über alle Wörter $z \in L$ zum Erfolg (leichte Gegenbeispiele), noch ist es korrekt, eine oder nur bestimmte Zerlegungen zu betrachten.
Oft wurde etwa ein Wort $z = a^n \$\$ a^n$ (o.ä.) gewählt und dabei übersehen, daß Alice einfach $z = uvwxy$ mit $vwx = a\$\$a$ und $w = \$\$$ wählen kann, um das Spiel zu gewinnen. Ähnliche Probleme treten auf, wenn keines der Teilwörter zwischen den \$ das leere Wort ist, weil sich dann immer eine passende Zerlegung finden läßt.
Übrigens: Das Pumpinglemma gilt immer, und nicht nur für bestimmte Sprachen oder für bestimmte Wörter.

Aufgabe 4 (10 Punkte)

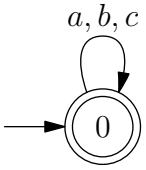
Als synchronisiertes Produkt erhalten wir mit dem Verfahren aus der Vorlesung:



Um den minimalen DFA zu bilden, wenden wir nun die Potenzmengenkonstruktion an und erhalten den folgenden Automaten.



Da alle Zustände auch Endzustände sind (und der Automat deterministisch ist, also insbesondere nicht blockiert), erkennt er die universelle Sprache. Der minimale DFA ist also:



Alternativ kann man auch dem NFA direkt ansehen, daß er nicht nie blockiert und nur Endzustände besitzt.

Typische Fehler: Sofern die Definition des synchronisierten Produkts bekannt war, wurden bei dieser Aufgabe wenig Fehler gemacht.