Aufgabe 1

Die Sprache L über den Alphabet $\Sigma = \{0, 1\}$ enthält alle Wörter, bei denen beim Lesen von links nach rechts der Unterschied in der Zahl der 0en und 1en stets höchstens 3 ist. Also ist $w \in L$ genau dann, wenn für alle u, v mit $w = uv$ gilt $||u|_0 - |u|_1| \leq 3$. Erinnerung: $|w|_a$ bezeichnet die Zahl der a's im Wort w.

(a) Sei $A = (Q, \Sigma, \delta, q_0, E)$ ein deterministischer endlicher Automat für L. Es sei $w_1 = 111$, $w_2 = 11$, $w_3 = 1$,$w_4 = \epsilon$, $w_5 = 0$, $w_6 = 00$, $w_7 = 000$. Machen Sie sich klar, dass der Automat jedes dieser Wörter verarbeiten können muss. Folgern Sie, dass der Automat mindestens sieben Zustände haben muss. Schreiben Sie Ihr Argumentation schlüssig und vollständig auf.

> Ein deterministischen endlicher Automat hat keinen zusätzlichen Speicher zur Verfügung, in dem die Anzahl der bisher vorkommenden 0 und 1 gespeichert werden könnte. Ein DEA kann die von der Sprache benötigten Anzahl an 0 und 1 nur in Form von Zustanden speichern. Um die Anzahl von 3 Einsen bzw. 3 Nullen zu speichern, sind also 6 Zustände nötig. Da die Sprache auch das leere Wort erkennen soll, ist noch ein zusätzlicher Zustand für dieses leere Wort nötig.

(b) Begründen Sie, dass L regulär ist.

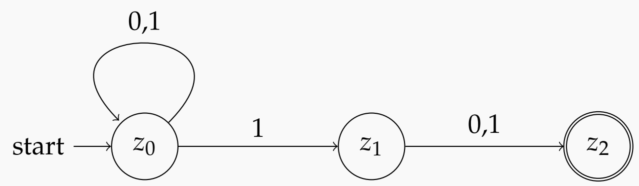
> L ist nicht regulär. Widerspruchsbeweis durch das Pumping Lemma.
>
> **j** = Pumping-Zahl
>
> **k** = zur Pumping-Zahl abhängige Zahl
>
> **i** = Zahl, die die Werte $\{0, 1, 2, 3 \ldots\}$ annimmt und mit der aufgepumpt wird.
>
> $\omega = 0^j 1^k = uvw$ mit $|j - k| \leq 3$ und $j, k \geq 0$
>
> $\omega = 0^j 1^{j+k} = uvw$ mit $|k| \leq 3$
>
> Wir teilen 0^j in uv auf.
>
> $u = 0^{j-i}$ $v = 0^i$ $w = 1^{j+k}$
>
> $\rightarrow$ man kann nicht unendlich aufpumpen. L ist nicht regulär.

(c) Jemand behauptet, diese Sprache sei nicht regulär und gibt folgenden „Beweis" dafür an: Wäre L regulär, so sei n eine entsprechende Pumping-Zahl. Nun ist $w = (01)^n \in L$. Zerlegt man nun $w = uxv$, wobei $u = 0$, $x = 1$, $v = (01)^{n-1}$, so ist zum Beispiel $ux^5v \notin L$, denn es ist $ux^5v =$ 01111101010101.... Legen Sie genau dar, an welcher Stelle dieser „Beweis" fehlerhaft ist.

> (i) Das Wort $(01)^n$ ist schlecht gewählt: Das für den Pumping-Lemma Beweis gewählte Wort $(01)^n$ spiegelt nicht die Eigenschaft der Sprache wieder, dass sich die Anzahl von Nullen und Einsen um maximal drei unterscheidet.

(ii) Außerdem wurde das $(01)^n$ Wort falsch zerlegt. Für die Pumping-Zahl $n = 3$ gibt es sehr wohl eine Zerlegung, die beim Aufpumpen regulär ist, also: $\omega = 010101$ ($u = 01$, $x = 01$ und $v = 01$). $ux^5v = 01010101010101 \in L$.

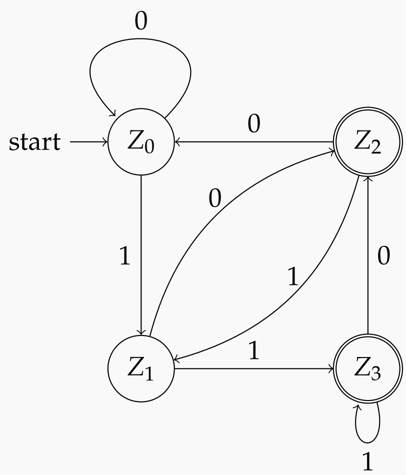
(d) In anderen Fällen können nichtdeterministische endliche Automaten echt kleiner sein als die besten deterministischen Automaten. Ein Beispiel ist die Sprache $L_2 = \Sigma^*1\Sigma$ aller Wörter, deren vorletztes Symbol 1 ist. Geben Sie einen nicht-deterministischen Automaten mit nur drei Zuständen an, L_2 erkennt.



(e) Führen Sie auf Ihrem Automaten die Potenzmengenkonstruktion und anschließend den Minimierungsalgorithmus durch. Wie viele Zustände muss ein deterministischer Automat für L_2 also mindestens haben?

Potenzmengenkonstruktion

Name	Zustandsmenge	Eingabe 0	Eingabe 1
Z_0	Z_0 $\{z_0\}$	Z_0 $\{z_0\}$	Z_1 $\{z_0, z_1\}$
Z_1	Z_1 $\{z_0, z_1\}$	Z_2 $\{z_0, z_2\}$	Z_3 $\{z_0, z_1, z_2\}$
Z_2	Z_2 $\{z_0, z_2\}$	Z_0 $\{z_0\}$	Z_1 $\{z_0, z_1\}$
Z_3	Z_3 $\{z_0, z_1, z_2\}$	Z_2 $\{z_0, z_2\}$	Z_3 $\{z_0, z_1, z_2\}$



Minimierungsalgorithmus

Z_0	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
Z_1	$*^2$	$\varnothing$	$\varnothing$	$\varnothing$
Z_2	$*^1$	$*^1$	$\varnothing$	$\varnothing$
Z_3	$*^1$	$*^1$	$*^2$	$\varnothing$
	Z_0	Z_1	Z_2	Z_3

$*^1$ Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.

$*^2$ Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.

$*^3$ In weiteren Iterationen markierte Zustände.

Übergangstabelle

Zustandspaar	**0**	**1**
(Z_0, Z_1)	$(Z_0, Z_2)\ *^2$	$(Z_1, Z_3)\ *^2$
(Z_2, Z_3)	(Z_0, Z_1)	$(Z_1, Z_3)\ *^2$

Wie aus der oben stehenden Tabelle abzulesen ist, gibt es keine äquivalenten Zustände. Der Automat kann nicht minimiert werden. Er ist bereits minimal.