



Aufgabe 1

Beweis durch Induktion über $n \in \mathbb{N}$

Induktionsanfang: $n = 0$: $\bigcup_{i=0}^0 A_i$. A_0 ist nach der Aufgabenstellung endlich $\Rightarrow A$ abzählbar.

Induktionsschritt: $n \rightarrow n + 1$: Zu zeigen: $\bigcup_{i=1}^{n+1} A_i$ ist abzählbar.

Induktionsvoraussetzung: $\bigcup_{i=1}^n A_i$ ist abzählbar.

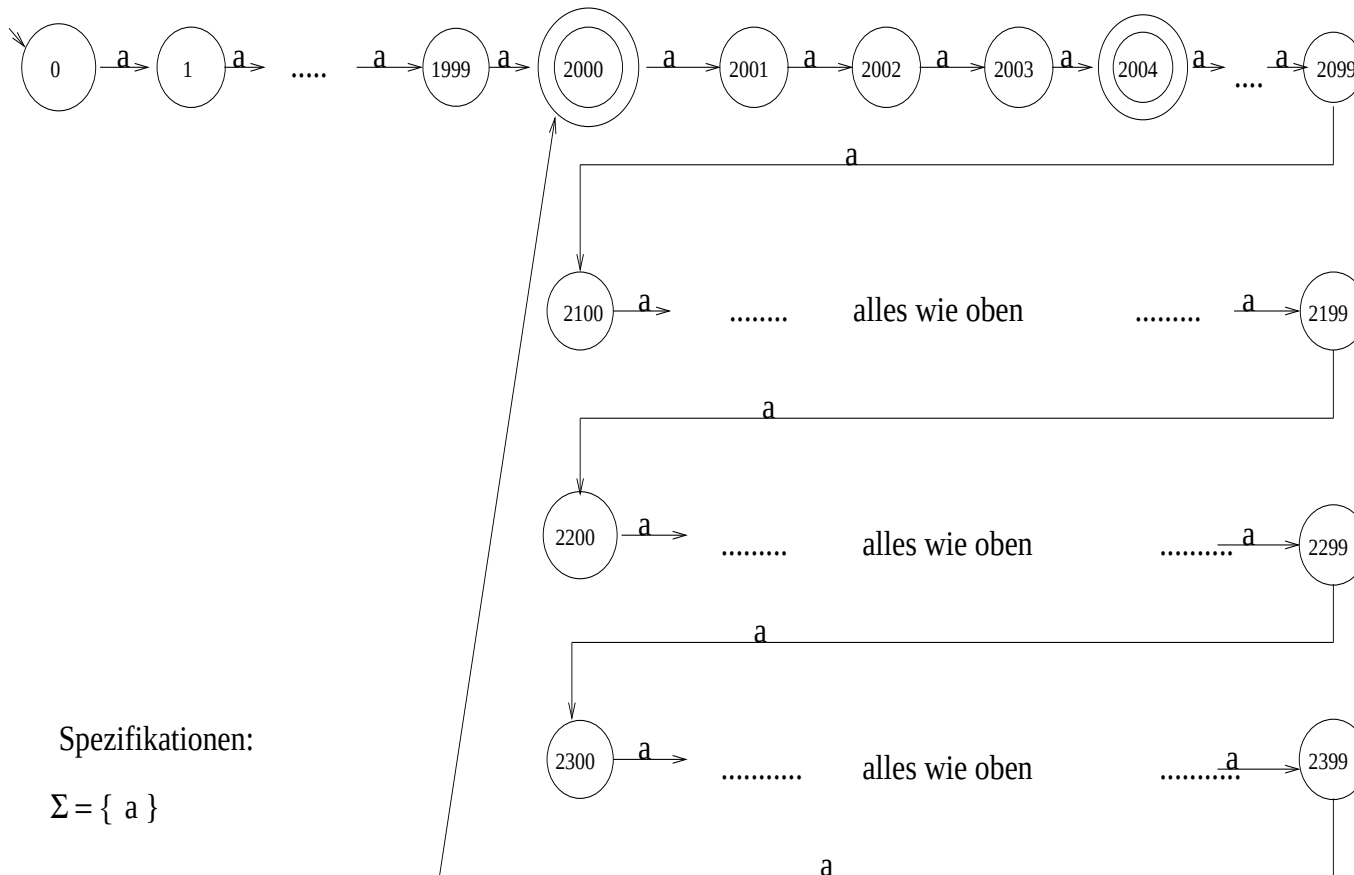
$$\bigcup_{i=1}^{n+1} A_i = \left(\bigcup_{i=1}^n A_i \right) \cup A_{n+1}$$

$\bigcup_{i=1}^n A_i$ ist nach Induktionsvoraussetzung abzählbar, A_{n+1} ist wegen der Endlichkeit (Aufgabenstellung) ebenfalls abzählbar. Da die Vereinigung zweier abzählbaren Menge abzählbar ist (dies wurde in der Vorlesung gezeigt), folgt daraus, daß $\bigcup_{i=1}^{n+1} A_i$ abzählbar ist.



Aufgabe 2

Skizze des Automaten:



Spezifikationen:

$$\Sigma = \{ a \}$$

$$Q = \{ i \mid 0 \leq i \leq 2399 \}$$

$$F = \{ i \mid (i \bmod 4 = 0) \wedge (i \bmod 100 \neq 0) \vee (i \bmod 4 = 0) \wedge (i \bmod 400 = 0) \}$$

$$s = 0$$

$$\Delta = \{ ((i, a, i+1) \text{ wenn } i < 2399), (i, a, 2000) \text{ wenn } i = 2399 \}$$

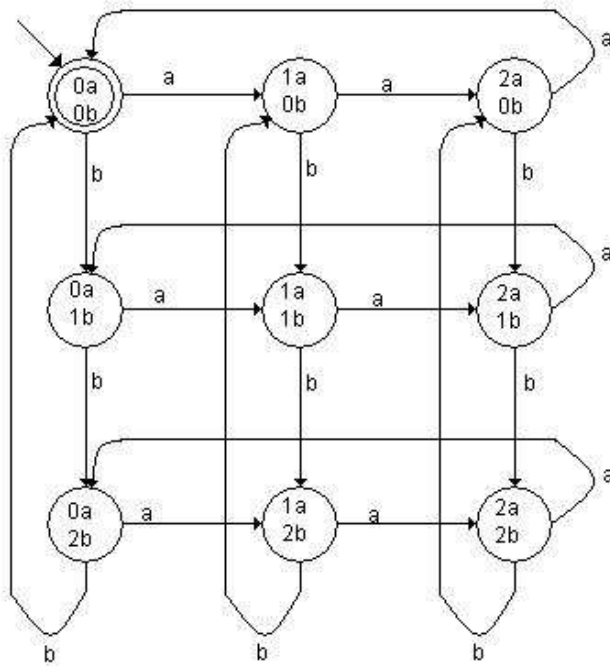


Aufgabe 3

- Die Menge der Fortsetzungssprachen von L ist die Menge aller Sprachen, die an ein Teilwort von L angehängt jeweils wieder ein Wort von L sind.

$$F_L = \{L'(anf) \in \{a,b\}^* \mid \text{Anzahl der a und b in Worten der Sprache } L' \text{ plus die Anzahl der a und b in anf ist durch 3 teilbar}\}$$

•





Aufgabe 4

1. Zunächst gilt es zu bemerken, dass C nicht der Endzustand ist. Wir haben also zwei Endzustände A und B . Wenn der Automat sich also einmal im Zustand C befindet wird er nicht terminieren, d.h. kein gültiges Wort erzeugen. Da A der Anfangszustand und zugleich Endzustand ist, gilt schonmal, dass das leere Wort ϵ , in der Sprache L_1 enthalten ist. Wir können in A verweilen, indem wir beliebig viele 0'en an das Wort anhängen. Wir können nun auch übergehen da wir uns in einem Endzustand befinden oder in B übergehen indem wir eine 1 anhängen. B ist ebenfalls ein Endzustand und wir können noch weitere 1'en anhängen. Es darf jedoch keine 0 mehr folgen, da wir sonst in C landen, und der Automat dann nicht mehr terminiert. Unsere Sprache besteht also aus den Worten, die mit einer beliebigen Anzahl 0'en beginnen und mit einer beliebigen Anzahl 1'en enden.

$$L_1 = \{uv \in \{0,1\}^* \mid u \in \{0\}^*, v \in \{1\}^*\}$$

2. Wenn man sich den zweiten Automaten, genauer anschaut, stellt man fest, dass es eigentlich nicht viele Möglichkeiten gibt in den Endzustand D zu gelangen. In der Tat ist dies nur möglich mit der Zeichenfolge 101. Alle Worte dieser Sprache enden also mit 101. Ganz so leicht ist aber dann doch nicht, denn

$$L_2 = \{w \in \{0,1\}^* \mid w \text{ endet mit ungerader Anzahl 10 Kombinationen gefolgt von einer 1}\}$$

Begründung: Es gilt, nach zwei 0'en ist man immer im Zustand A und nach zwei 1'en immer in Zustand B . Uns kann also egal sein was davor war. Betrachten wir nun das letzte dieser Paare, so können wir davon ausgehen, dass sich ab jetzt 1 und 0 nur noch abwechseln. Und es gilt, dass man sich nach einer ungeraden Anzahl von 1 0 Kombinationen immer in Zustand C befindet, und dann noch eine 1 benötigt um in Zustand D zu gelangen.

3. Auch der dritte Automat ist nicht gerade leicht. Aber man kann sich die Sache etwas leichter machen, indem man sich die Sprachen anschaut, die entstehen, wenn man jeweils nur einen Endzustand betrachtet. Wir betrachten also mal nur die Sprache, die durch Endzustand A definiert wird und nennen sie L_A . Die andere nennen wir entsprechend L_C . Im Prinzip besteht der Automat aus zwei Hälften, der linken Seite mit den Zuständen A und B und der rechten mit C und D . Weiter kann man erkennen, dass bei einer 1 die Seite gewechselt wird. Das heißt also, dass die Sprache L_A nur aus Worten besteht, die eine gerade (0 gehört zu den geraden Zahlen) Anzahl von 1'en enthält, und L_C entspricht eine ungerade Anzahl.

Betrachten wir zunächst L_C genauer (ist leichter). Wir landen in C immer nach einer ungeraden 1 und kehren dann immer nach zwei 0'en zu C zurück. Dies bedeutet

$$L_C = \{w \in \{0,1\}^* \mid \text{Anzahl der 1'en ist ungerade und } w \text{ endet mit gerader Anzahl 0'en}\}$$



Analog besteht L_A nur aus Worten die eine gerade Anzahl 1'en enthalten und mit einer ungeraden Anzahl 0'en enden. Mit einer Ausnahme, wenn nämlich w keine 1 enthält, dann besteht w aus einer geraden Anzahl 0'en, da wir in A beginnen und nicht in B .

$$L_A = \{w \in \{0,1\}^* \mid \text{Anzahl 1'en in } w \text{ ist gerade } > 0 \text{ und endet mit ungerader Anzahl 0'en oder } w \text{ besteht nur aus gerader Anzahl 0'en}\}$$

Aufgabe 5

a) Zu zeigen ist, dass die Sprache $L_1 = \{a^{2^n} \mid n \in \mathbb{N}\}$ nicht regulär ist. Zwei verschiedene Verfahren, um die Nichtregulärkeit einer Sprache zu zeigen, sind das Pumping-Lemma für reguläre Sprachen und die Myhill-Nerode Relation.

- **Beweis mit Pumping - Lemma**

Wir nehmen an, dass die Sprache L_1 regulär ist. Nach dem Pumping Lemma gibt es dann eine Zahl $n \geq 1$, so dass jedes Wort $w = a^{2^n} \in L_1$ mit $|w| > n$ zerlegt werden kann in $w = xyz$ mit $y \neq \epsilon$. Dann ist $y = a^k, k \geq 1$. Nach dem Lemma folgt dann, dass alle $w' = xy^i z$ für alle $i \in \mathbb{N}$, d.h. alle $w' = a^{2^n + k}$ in L_1 sind. Daraus folgt aber, dass $2^n + k = 2^m$ für alle $k \geq 1, m > n$ gelten soll. Das ist offenbar ein Widerspruch, weil für $k=1$ die linke Seite keine zweier Potenz ist.

$\Rightarrow$ die Sprache L_1 ist keine reguläre Sprache.

- **Beweis mit dem Satz von Myhill - Nerode**

Wir nehmen an, dass die Sprache L_1 regulär ist. Dann existiert ein DEA, der die Sprache akzeptiert und der minimal ist. Nach dem Satz von Myhill - Nerode existieren endlich viele Äquivalenzklassen bezüglich der Myhill - Nerode Relation. Das ist aber ein Widerspruch, weil für beliebige m und $n, m > n, a^{2^n}$ und a^{2^m} nicht in derselben Äquivalenzklasse sind (daraus folgt, dass es unendlich viele Äquivalenzklassen gibt). Hier gilt, dass der Abstand zwischen zwei aufeinanderfolgende 2-er Potenzen 2^m und $2^{m+1} - 2^m$ ist. Setzt man also a^{2^n} mit 2^n a's fort, bleibt man in derselben Sprache. Setzt man aber a^{2^m} mit 2^n a's fort, ist man nicht mehr in der Sprache, weil die nächste 2-er Potenz $2^{m+1} = 2^m + 2^m$ ist und $2^m + 2^m > 2^m + 2^n$ gilt.

$\Rightarrow$ es existieren unendlich viele Äquivalenzklassen.

$\Rightarrow$ die Sprache L_1 ist nicht regulär.

b) Zu zeigen ist, dass die Sprache $L_2 = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$ nicht regulär ist.

- **Beweis mit dem Satz von Myhill - Nerode**

Wir nehmen an, dass die Sprache L_2 regulär ist. Nach dem Satz von Myhill - Nerode existieren endlich viele Äquivalenzklassen bezüglich der Myhill - Nerode Relation.

Es gilt:



$[\epsilon] \neq [a]$, weil $\epsilon \underline{ab} \in L_2$, aber $a \underline{ab} \notin L_2$

$[a] \neq [aa]$, weil $a \underline{b} \in L_2$, aber $aa \underline{b} \notin L_2$

Im Allgemeinen gilt, dass $[a^i] \neq [a^j]$ für $i \neq j$.

$\Rightarrow$ es gibt unendlich viele Äquivalenzklassen. Das ist ein Widerspruch!

$\Rightarrow$ die Sprache L_2 ist nicht regulär.

- **Beweis mit Pumping - Lemma**

Wir nehmen an, dass die Sprache regulär ist. Sei n die nach dem Pumping-Lemma existierende Konstante. Offenbar ist $w = a^n b^n \in L_2$ und nach dem Pumping-Lemma existiert eine Zerlegung xyz von w mit $|xy| \leq n$ und $y \neq \epsilon$. Dann gilt, dass $y = 1^j$ für irgendein $j \geq 1$. Nach dem Pumping-Lemma muss auch $w' = a^{n+j} b^n$ in der Sprache sein.

Das ist offenbar ein Widerspruch!

$\Rightarrow$ die Sprache L_2 ist nicht regulär.