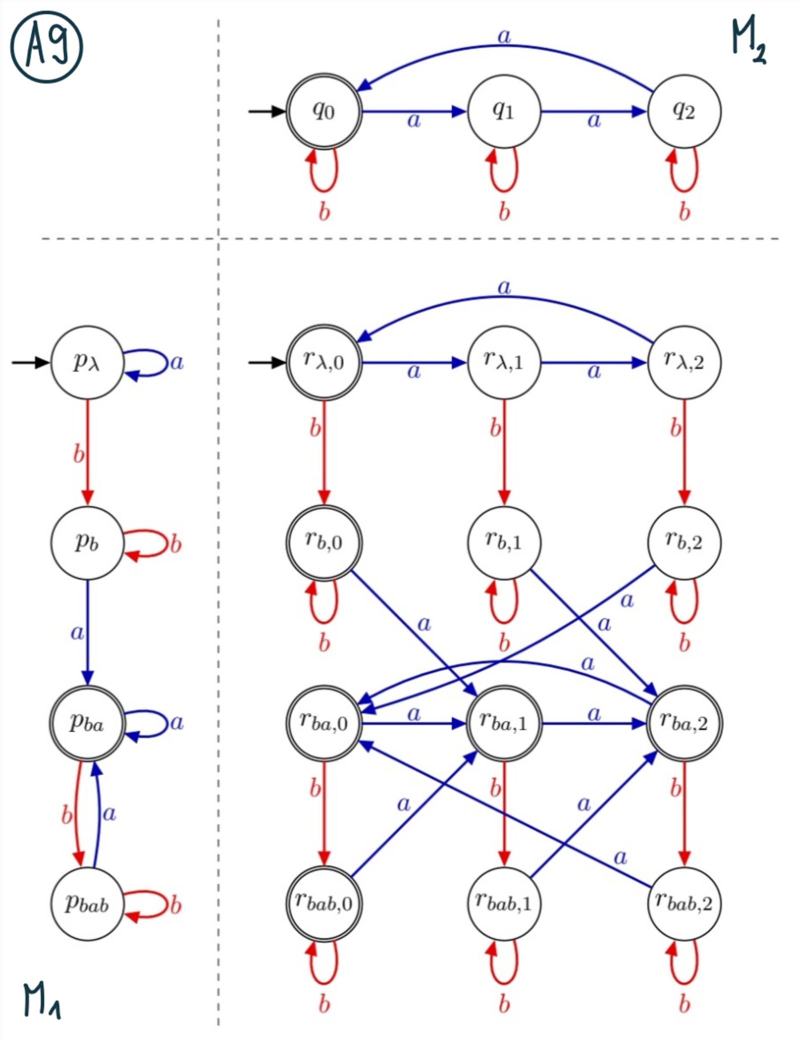
Bmk zu Serie 3: Ich empfehle euch diese Methode für die Konstruktion von Produktautomaten zu übernehmen:

$L(M_1) := \{w \in \{a,b\}^* \mid |w|_a \equiv 0 \mod 3\}$

$L(M_2) = \{w \in \{a,b\}^* \mid w \text{ endet auf } a \text{ und } w \text{ hat Teilwort } ab\}$



3.4 Beweise der Nichtexistenz

Bmk: Das Ziel in diesem Kapitel ist es, für bestimmte Sprachen zu beweisen, dass kein EA existiert, der diese Sprachen erkennt (d.h. $L \notin \mathcal{L}_{EA}$).

Weil wir dafür nicht einfach alle möglichen EAs ausprobieren können, schauen wir uns Eigenschaften an, die ein EA oder eine reguläre Sprache erfüllen muss und zeigen meist per Widerspruchsbeweis, dass dies für die geg. Sprache L nicht der Fall ist.

Wir lernen dafür 3 verschiedene Methoden kennen.

1) Lemma 3.3

Lem 3.3: Sei $A = (Q, \Sigma, \delta_A, q_0, F)$ ein EA.

Seien $x, y \in \Sigma^*$ mit $x \neq y$ s.d. $x, y \in Kl[p]$ für ein $p \in Q$, d.h.

$$(q_0, x) \vdash_A^* (p, \lambda) \quad \text{und} \quad (q_0, y) \vdash_A^* (p, \lambda)$$

Dann gilt: $\forall z \in \Sigma^* \ \exists r \in Q$ s.d. $xz, yz \in Kl[r]$.

Insbesondere gilt: $xz \in L(A) \iff yz \in L(A)$

Aufg 1) Sei $L = \{ x1y \in \{0,1\}^* \mid |x| = |y| \}$.

Zeige mit Lemma 3.3: $L \notin \mathcal{L}_{EA}$.

Lösung: 1) Annahme per Widerspruch: $\exists A = (Q, \Sigma, \delta_A, q_0, F)$ EA mit $L(A) = L$ $\quad (L \in \mathcal{L}_{EA})$

- Wir betrachten die $|Q|+1$ Wörter $w_k := 0^k 1$ für $0 \leq k \leq |Q|$. $\quad Q|:$
- Weil dies mehr Wörter sind als A Zustände besitzt, gilt für ein $0 \leq i < j \leq |$

  $$\hat{\delta}_A(q_0, 0^i 1) = \hat{\delta}_A(q_0, 0^j 1)$$
- Aus Lemma 3.3 folgt: $\forall z \in \{0,1\}^*: \quad \hat{\delta}_A(q_0, 0^i 1 z) = \hat{\delta}_A(q_0, 0^j 1 z)$.
- Sei $z := 0^i$.
- Dann erhalten wir einen Widerspruch, da $0^i 1 0^i \in L$ aber $0^j 1 0^i \notin L$. $\lightning$
- Somit ist L keine reguläre Sprache. $\square$

Rezept 1:
1) Annahme per Widerspruch: $\exists A = (Q, \Sigma, \delta_A, q_0, F)$ EA mit $L(A) = L$
2) Gut gewählte $|Q|+1$ Wörter w_k, $0 \leq k \leq |Q|$ finden
3) Weil dies mehr Wörter sind als A Zustände besitzt, gilt für ein $0 \leq i < j \leq |Q|$

   $$\hat{\delta}_A(q_0, w_i) = \hat{\delta}_A(q_0, w_j)$$
4) Lemma 3.3 anwenden
5) Gut gewähltes $z \in \Sigma^*$ finden s.d. $w_i z \in L \iff w_j z \notin L$
6) $L \notin \mathcal{L}_{EA}$ aus Widerspruch folgern

2) Pumping-Lemma

Lem 3.4 (Pumping-Lemma für reguläre Sprachen)

Sei L eine reguläre Sprache.

nicht nur in L

Dann $\exists n_0 \in \mathbb{N}$ konst. s.d. sich jedes Wort $w \in \Sigma^*$ mit $|w| \geq n_0$ in 3 Teile $x, y, z \in \Sigma^*$ zerlegen lässt, d.h. $w = yxz$, wobei

(i) $|yx| \leq n_0$,

(ii) $|x| \geq 1$ und

(iii) entweder $\{xy^k z \mid k \in \mathbb{N}\} \subseteq L$ oder $\{xy^k z \mid k \in \mathbb{N}\} \cap L = \emptyset$.

Aufg 2) Sei $L := \{ w \in \{a,b\}^* \mid w = vabv^R$ für $v \in \{a,b\}^*$ und v^R Umkehrung von $v \}$

Zeige mit Pumping-Lemma: $L \notin \mathcal{L}_{EA}$.

Lösung: 1) Annahme per Widerspruch: Sei L regulär.

2b) Aus dem Pumping-Lemma folgt: $\exists n_0 \in \mathbb{N}$ konst. s.d. $\forall w \in \{a,b\}^*$ mit $|w| \geq n_0$: $w = yxz$ wobei (i), (ii), (iii) erfüllt sind (Bedingungen aufschreiben!)

2a) Wir betrachten das Wort $v = a^{n_0}$ bzw. $w = a^{n_0} ab\, a^{n_0}$

2b) Aus (i) und (ii) folgt: $x = a^k$ mit $1 \leq k \leq n_0$,

$y = a^l$ mit $l + k \leq n_0$,

$z = a^{n_0 - l - k} ab\, a^{n_0}$

3) Seien $i = 1$ und $j = 2$.

Dann gilt: $w_1 = yx^1 z = a^l a^k a^{n_0 - l - k} ab\, a^{n_0} = a^{n_0} ab\, a^{n_0} \in L$

$w_2 = yx^2 z = a^l a^{2k} a^{n_0 - l - k} ab\, a^{n_0} = a^{n_0 + k} ab\, a^{n_0} \notin L$

4) Dies ist ein Widerspruch zu (iii). ↯ $\Rightarrow$ Somit gilt $L \notin \mathcal{L}_{EA}$ □

Rezept 2: 1) Annahme per Widerspruch: $L \in \mathcal{L}_{EA}$.

2a) Gut gewähltes Wort $w \in \Sigma^*$ finden, das länger als n_0 für ein bel. n_0 ist. (n_0 bleibt eine Variable.)

2b) Pumping-Lemma anwenden: $w = xyz$ mit $|yx| \leq n_0$ und $|x| \geq 1$

(Wir können nicht auswählen, wie x und y genau definiert sind.)

3) Ein $i, j \in \mathbb{N}$ finden mit: $yx^i z \in L \;\wedge\; yx^j z \notin L$

4) Begründen, dass dies ein Widerspruch zu (iii) im Pumping-Lemma ist $\Rightarrow L \notin \mathcal{L}_{EA}$

3) Kolmogorov-Methode

Satz 3.1: Sei $L \subseteq (\Sigma_{bool})^*$ eine reguläre Sprache.

Sei $L_x := \{ y \in (\Sigma_{bool})^* \mid xy \in L \}$ für jedes $x \in (\Sigma_{bool})^*$.

Dann $\exists c$ konst, s.d. $\forall x, y \in (\Sigma_{bool})^*$:

$$K(y) \leq \lceil \log_2(n+1) \rceil + c$$

falls y das n-te Wort der Sprache L_x ist.

Aufg 3) Sei $L := \{ ww \mid w \in \{0,1\}^* \}$

Zeige mit Methode der Kolmogorov-Komplexität: $L \notin \mathcal{L}_{EA}$

Lösung: 1) Annahme per Widerspruch: $L \in \mathcal{L}_{EA}$

2) Wir betrachten die Folge $(x_k)_{k \in \mathbb{N}}$ mit $x_k := (01)^k 11$.

Wir betrachten nun die Menge $L_{x_k} = \{ y \in \{0,1\}^* \mid x_k y \in L \} = \{ y \in \{0,1\}^* \mid (01)^k 11 y \in L \}$

Das kanonisch erste Wort in L_{x_k} ist x_k selbst.

Somit sind für verschiedene k auch die kanonisch ersten Wörter verschieden.

3) Mit Satz 3.1 gilt also für jedes x_k : $K(x_k) \leq \lceil \log(2) \rceil + c = c'$ für eine Konstante c'.

4) Es gibt unendl. viele Wörter $(x_k)_{k \in \mathbb{N}}$ aber nur endl. viele versch. Programme der Länge $\leq c'$. ↯

Dies ist ein Widerspruch und somit gilt $L \notin \mathcal{L}_{EA}$. □

Rezept 3: 1) Annahme per Widerspruch: $L \in \mathcal{L}_{EA}$

2) Eine gut gewählte unendl. Folge $(x_k)_{k \in \mathbb{N}}$ von Präfixen finden und $L_{x_k} = \{ y \in \Sigma^* \mid x_k y \in L \}$ betrachten

$(x_k)_{k \in \mathbb{N}}$ soll so gewählt werden, dass die kanonisch ersten Wörter in versch. $L_{x_k}, L_{x_{k'}}$ auch versch. sind.

3) Satz 3.1 anwenden: $\forall k \in \mathbb{N}$ gilt, dass das erste (kann variieren) Wort in L_{x_k} höchstens Kolm.-Komp. $\lceil \log_2(2) \rceil + c = c'$ konst. hat.

4) Widerspruch erläutern: Wir haben unendl. viele versch. Wörter, deren Kolm.-Komp. höchstens die Konstante c' ist. ↯ $\Rightarrow$ Somit gilt $L \notin \mathcal{L}_{EA}$

Bmk: Die Kolmogorov-Methode und das Pumping-Lemma können nicht immer benutzt werden um Nichtregularität einer Sprache zu zeigen.

D.h. Es gibt nicht reguläre Sprachen, für die Satz 3.1 bzw. Lemma 3.4 trotzdem gelten.