13. Aufgabe (Akzeptanz durch Kellerautomat)

Gegeben sei ein Kellerautomat K mit Startzustand z_0 und akzeptierendem Zustand z_a und den Transitionen:

$(z_0, –, \$, z_1, S)$,
$(z_1, –, S, z_1, 0S0)$, $(z_1, –, S, z_1, 1S1)$, $(z_1, –, S, z_1, \varepsilon)$
$(z_1, 0, 0, z_1, \varepsilon)$, $(z_1, 1, 1, z_1, \varepsilon)$, $(z_1, –, \$, z_a, \varepsilon)$

Welche Wörter über $\Sigma = \{0, 1\}$ werden mindestens (also ggf. auch andere) von K akzeptiert:

		JA	NEIN
(i)	alle Palindrome?	◯	◯
(ii)	alle Palindrome gerader Länge?	◯	◯
(iii)	alle Doppelwörter ww, $w \in \Sigma^*$?	◯	◯
(iv)	alle Binärzahlen mit Werten der Art $2^{2n}-1$, $n = 1, 2, 3, \ldots$	◯	◯
(v)	alle Binärzahlen mit Werten der Art $2^{2n}+1$, $n = 1, 2, 3, \ldots$	◯	◯
(vi)	alle Binärzahlen mit Werten der Art $2^{2n+1}+1$, $n = 1, 2, 3, \ldots$	◯	◯

Lösung

(i) NEIN: 1 z.B. wird nicht akzeptiert.

(ii) JA: Der Kellerautomat „spielt dafür im Keller eine Grammatik durch“, ähnlich wie beim konstruktiven Beweis, dass kontextfreie Sprachen durch Kellerautomaten akzeptiert werden. Sie ist mit den Regeln $S \rightarrow 0S0 \mid 1S1 \mid \varepsilon$ zwar nicht kontextfrei, erzeugt aber offensichtlich die {0, 1}-Palindrome gerader Länge.

(iii) NEIN: 1010 wird z.B. nicht akzeptiert.

(iv) JA: Alle diese sind Palindrome gerader Länge: $2^{2\cdot1}-1 \approx 11$, $2^{2\cdot2}-1 \approx 1111$ usw.

(v) NEIN: 101 mit Wert 2^2+1 wird z.B. nicht akzeptiert, hat ungerade Länge.

(vi) JA: Alle diese sind Palindrome gerader Länge: $2^{2\cdot1+1}+1 \approx 1001$, $2^{2\cdot2+1}+1 \approx 100001$ usw.

14. Aufgabe (Berechnung durch Turing-Maschine)

Gegeben sei eine Turingmaschine *M* mit Bandalphabet Σ = {0,1}, Startzustand z_a, Endzustand z_e und dem Programm:

$\delta(z_a, 0)$	=	$(z_0, 0, R)$
$\delta(z_a, 1)$	=	$(z_1, 1, R)$
$\delta(z_0, 0)$	=	$(z_0, 0, R)$
$\delta(z_0, 1)$	=	$(z_0, 1, R)$
$\delta(z_0, B)$	=	$(z_L, 0, L)$
$\delta(z_1, 0)$	=	$(z_1, 0, R)$
$\delta(z_1, 1)$	=	$(z_1, 1, R)$
$\delta(z_1, B)$	=	$(z_L, 1, L)$
$\delta(z_L, 0)$	=	$(z_L, 0, L)$
$\delta(z_L, 1)$	=	$(z_L, 1, L)$
$\delta(z_L, B)$	=	(z_e, B, R)

Welchen Output produziert die Turingmaschine aus einem Input *w*, der ein nichtleeres Wort aus Nullen und Einsen ist?

Lösungsbeispiel

Das Ergebnis ist das Eingabewort, an das aber das erste Zeichen dieses Wortes angehängt wurde.

Durch z_0 oder z_1 merkt sich die TM, ob vorne 0 oder 1 steht. In beiden Zuständen geht die TM nach rechts und hängt hinten 0 bzw. 1 an. In z_L wird der Lese-Schreib-Kopf mit L-Bewegungen (und dem obligatorischen Herunterfallen vom String mit Sprung zurück) auf das erste Zeichen des Ausgabewortes gesetzt.

15. Aufgabe (Turing-Reduzierbarkeit)

Es seien die Sprachen L_1, L_2 über dem Alphabet Σ = {0,1 } wie folgt gegeben:

$$L_1 = \{0^n \mid n \geq 1\},$$
$$L_2 = \{10^n \mid n \geq 1\}.$$

Zeigen Sie, dass das Wortproblem der Sprache L_1 auf das Wortproblem der Sprache L_2 Turing-reduzierbar ist.

Lösungsbeispiele

Der Algorithmus setzt als Ausgabe eine 1 vor die Eingabe.
Die Turing-Maschine mit Startzustand z_0, Endzustand z_e und mit

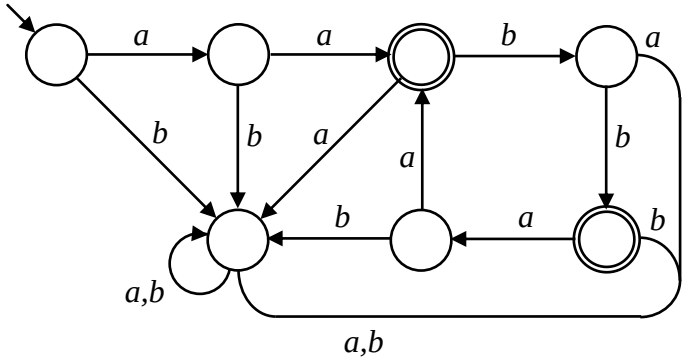
δ(z0,0) = (z0,0,L)
δ(z0,1) = (z0,1,L)
δ(z0,B) = (ze,1,N)

berechnet die Funktion $f: \begin{cases} \Sigma^* \to \Sigma^* \\ w \mapsto 1\,w \end{cases}$, sodass genau dann $f(w) \in L_2$ gilt, wenn $w \in L_1$ gilt.

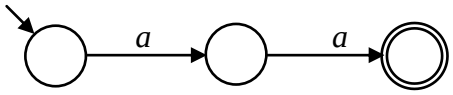
16. Aufgabe (Diverse Sprachbeschreibungen)

In der Folge werden zehn Sprachen, L_1 bis L_{10}, über $\Sigma = \{a, b\}$ beschrieben, die untereinander teilweise übereinstimmen. Dies geschieht nach mehreren Methoden:

- natürlich-sprachlich
  - L_1: Jedes Wort enthält *aa*.
  - L_2: Jedes Wort beginnt mit *aa*. Auf *aa* folgt stets zunächst *bb* oder nichts mehr. Auf *bb* folgt stets zunächst *aa* oder nichts mehr.
- als regulärer Ausdruck (+ wie | bedeuten „oder“)
  - L_3: $(a+b)^* aa (a+b)^* (aa)^*$
  - L_4: *aa* (*bbaa*)*(ε+*bb*)
- als Automatensprache L_5 des folgenden endlichen Automaten:



- als Sprache L_6, die der folgende endliche nichtdeterministische Automat akzeptiert:



- als Sprache L_7 einer regulären Grammatik mit Startsymbol S:

  S → aR | bS R → a | aE | bS E → a | b | aE | bE

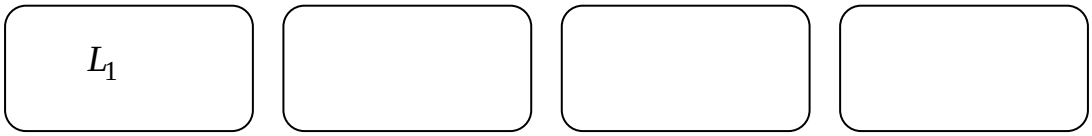
- als Sprache L_8 aller Wörter , für die eine Turing-Maschine das Ergebnis 1 berechnet, und zwar mit Anfangszustand z_0 und Endzustand z_e und folgender Zustandsüberführungsfunktion δ:

$$\delta(z_0,a)=(z_1,a,R), \quad \delta(z_0,b)=(z_0,b,R),$$
$$\delta(z_1,a)=(z_2,a,R), \quad \delta(z_1,b)=(z_0,b,R),$$
$$\delta(z_2,a)=(z_2,a,R), \quad \delta(z_2,b)=(z_2,b,R), \quad \delta(z_2,B)=(z_3,B,L),$$
$$\delta(z_3,a)=\delta(z_3,b)=(z_3,B,L), \quad \delta(z_3,B)=(z_e,1,N)$$

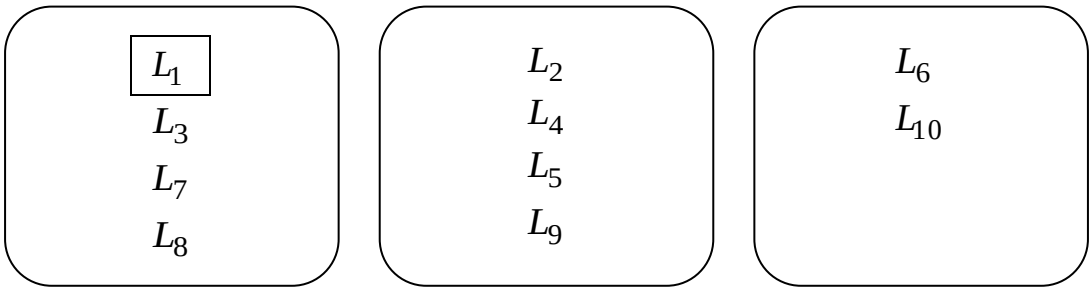
- als Sprache L_9 bzw. L_{10}, die ein Kellerautomat bei Endzustand z_e akzeptiert, mit Anfangszustand z_0 und folgender Transitionenmenge (links für L_9, rechts für L_{10}):

Δ_9 für L_9:	Δ_{10} für L_{10}:
$(z_0,a,\$,z_0,a)$, $(z_0,a,a,z_1,\varepsilon)$,	$(z_0,a,\$,z_0,1)$, $(z_0,a,1,z_0,2)$,
$(z_1,b,\$,z_1,b)$, $(z_1,b,b,z_2,\varepsilon)$,	$(z_0,-,2,z_e,\varepsilon)$
$(z_2,a,\$,z_2,a)$, $(z_2,a,a,z_1,\varepsilon)$,	
$(z_1,-,\$,z_e,\varepsilon)$, $(z_2,-,\$,z_e,\varepsilon)$	

Tragen Sie die Sprachen („L_n“) so in die folgenden Kästen ein, dass gleiche Sprachen im gleichen Kasten und unterschiedliche Sprachen in verschiedenen Kästen stehen. Es kann sein, dass nicht alle Kästen benutzt werden müssen.



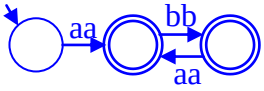
Lösung



… aa …

oder (a+b)* aa (a+b)*

aa(!)-bb-aa-bb-…

oder 

{ aa }

… wären anschauliche kurze (meist) formlose Kennzeichnungen, zum rascheren Einordnen.

Anmerkungen:

L3: Man sieht das garantierte aa mitten im Wort.
Das (aa)* am Ende ist wirkungslos und nur ein fieses Ablenkungsmanöver ☹, denn (a+b)*(aa)* = (a+b)* !

L4: *aa* (*bbaa*)* liefert nur L2-Wörter, aber nicht die mit bb endenden. Das repariert (ε+*bb*).

L5: Der Zustand links unten ist eine Falle/Mülltonne: Einmal drin wird nie mehr akzeptiert, und das geschieht genau dann, wenn von aabbaabb…. abgewichen wird.

L6: = {aa}, von L1 und L2 verschieden.

L7: R folgt auf a, E folgt auf aa. Alles (per E wie egal) darf kommen, sobald zwei a‘s nacheinander vorkamen, und erst nach zwei aufeinander folgenden a’s kann ein Wort enden, also L1.
Oder man zeichnet die Grammatik nach Rezept als Automat und enthält dann leicht erkennbar einen für L1.

L8: In z0 bzw. z1 bzw. z2 wurden gerade 0 bzw. 1 bzw. 2 a’s gelesen. In z0 bzw. z1 folgt auf b ein „neuer Versuch“ in z0.
In z2 läuft der Kopf ans Wortende und löscht in z3 rückwärts alles; dann wird 1 geschrieben, d.h. das Wort mit aa drin akzeptiert – L1!

L9: Das anfängliche a wird in z0 eingekellert, das nächste a leert den Keller und → z1.
Das erste b wird in z1 eingekellert, das nächste b leert den Keller und → z2.
z2 arbeitet praktisch wie z0.
Aufhören darf man bei leerem Keller (in z1, z2) aber nicht gleich am Anfang (in z0), also nach aa oder bb.

L10: Das erste a führt zu 1 im Keller, das zweite zu 2, und dann ist Schluss: {aa}