



Aufgabe 1

Es sei L eine kontextfreie Sprache.

a)

Möglichkeit 1:

Behauptung: $L_a = L \cap L^R$ ist nicht kontextfrei.

Beweis: Annahme: L_a ist kontextfrei. Dann wäre NKA-Sprache unter Schnitt abgeschlossen. (nach der Aufgabenstellung ist L kontextfrei $\Rightarrow^{Satz V} L^R$ kontextfrei) Also wäre $L \cap L^R = L_a$ kontextfrei.

Widerspruch zur Satz.Vorl. $\Rightarrow L_a$ ist nicht kontextfrei.

Möglichkeit 2:

Wir geben ein Gegenbeispiel an. Sei $L = \{a^c, b^l, a^l \mid c, l \in \mathbb{N}\}$

$\Rightarrow L^R = \{a^l, b^l, a^c \mid c, l \in \mathbb{N}\}$

$\Rightarrow L_a = L \cap L^R = \{a^l, b^l, a^l \mid l \in \mathbb{N}\}$. L_a ist keine kontextfreie Sprache.

Dies kann man genauso mit dem Pumpinglemma zeigen, wie wir in der Vorlesung gezeigt haben, dass $L_{abc} = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ nicht kontextfrei ist.

b)

Möglichkeit 1:

Behauptung: $L_b = L \cdot L^R$ ist eine kontextfreie Sprache.

Beweis: Nach der Aufgabenstellung ist L eine kontextfreie Sprache

$\Rightarrow^{Satz V} L^R$ ist kontextfrei $\Rightarrow^{Satz V} L \cdot L^R = L_b$ ist kontextfrei.

Möglichkeit 2:

Es sei $G = (V, T, S, P)$ eine Grammatik mit $L(G) = L$

Es sei $G^R = (V, T, S^R, P^R)$ eine Grammatik für L^R .

Definiere neue Grammatik $G^b = (V \cup \{S^b\}, T, S^b, P^b)$, wobei $P^b = P \cup P^R \cup \{S^b \rightarrow SS^R\}$.

Dann ist $L_b(G^b) = L \cdot L^R$. (Dies ist genauer zu begründen.)

c)

Behauptung: Sprache $L_c = \{ww^R \mid w \in L \text{ und } L \text{ ist kontextfrei}\}$ ist nicht kontextfrei.

Wir geben ein Gegenbeispiel an: Es sei $L = \{w = a^{2n}b^n \mid w \in L, n \in \mathbb{N}\}$.

Dann ist $L_c = \{a^{2n}b^{2n}a^{2n} \mid n \in \mathbb{N}\}$. Diese Sprache ist nicht kontextfrei.

Dies beweist man genauso (Pumpinglemma) wie wir in der Vorlesung gezeigt haben, dass $L_{abc} = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ nicht kontextfrei ist.

Aufgabe 2

Zu zeigen ist, dass die regulären Sprachen genau die rechtslinearen Sprachen sind.

Beobachtung: Es sei $G = (T, N, S, R)$ eine rechtslineare Grammatik. Wenn $S \Rightarrow_G^* \alpha$, dann ist $\alpha = xB$ oder $\alpha = x$ mit $x \in T^*, B \in N$.



1. Wir wollen zeigen, dass für jede rechtslineare Grammatik ein NEA existiert, der die gleiche Sprache akzeptiert. Zu jeder gegebenen Grammatik $G = (T, N, S, R)$ kann man einen NEA $M = (Q, \Sigma, s, F, \Delta)$ wie folgt konstruieren:

- $F = \{f\}$ Hierbei ist f ein zusätzliches Symbol.
- $Q := N \cup F$: Für jedes Nichtterminal gibt es einen Zustand in M .
- $\Sigma := T$: Jedes Terminal wird zu einem Symbol aus dem Alphabet.
- $s := S$: Startsymbol wird zu Startzustand.
- Für jede Regel der Form $A \rightarrow aB$ ist $(A, a, B) \in \Delta$.
- Für jede Regel der Form $A \rightarrow a \in R$ ist $(A, a, f) \in \Delta$. (Dies gilt auch für $a = \epsilon$.)

Es sei $w \in L(G)$. Dann existiert eine Ableitungsfolge, die w erzeugt. Dieser Ableitungsfolge entspricht in natürlicher Weise ein Pfad im Übergangsgraphen von M , der im Startzustand s beginnt und im Endzustand f endet. Beachte $A \rightarrow aB \in R \iff (A, a, B) \in \Delta$. Damit wird w von M akzeptiert. Diese Behauptung ist zu formal zu beweisen, etwa durch Induktion über die Länge der Ableitungsfolgen.

Der so konstruierte Automat akzeptiert die Sprache $L(G)$. Es gilt also: Jede rechtslineare Sprache ist eine reguläre Sprache.

2. Es sei L eine reguläre Sprache. Wir zeigen, es gibt eine rechtslineare Grammatik $G = (T, N, S, R)$, die L erzeugt. Ist $L = \{\epsilon\}$, dann ist die Behauptung klar. Es sei also im folgenden $L \neq \{\epsilon\}$. Dann existiert ein endlicher Automat $M = (Q, \Sigma, s, F, \Delta)$ ohne ϵ -Übergänge mit $L(M) = L$. Wir konstruieren G wie folgt:

- $N := Q$: Jeder Zustand wird zu einem Nichtterminalsymbol.
- $T := \Sigma$: Jedes Symbol aus dem Alphabet wird zu einem Terminal.
- $S := s$: Der Startzustand wird zum Startsymbol.
- $R = \{A \rightarrow aB \mid (A, a, B) \in \Delta\} \cup \{A \rightarrow a \mid (A, a, B) \in \Delta, B \in F\}$.

Die so definierte Grammatik ist rechtslinear, alle Regeln haben die angegebene Form. Wenn M ein Wort w akzeptiert, dann gibt es einen Pfad im Übergangsgraphen vom Startzustand s zu einem Endzustand. Diesem entspricht eine Folge von Ableitungen in der Grammatik, die w erzeugen. Dies ist genauer zu beweisen, etwa durch Induktion über die Länge des Pfades.

Diese Grammatik erzeugt $L(M)$. Also gilt: Jede reguläre Sprache ist eine rechtslineare Sprache.

Aufgabe 3

Teil a.)

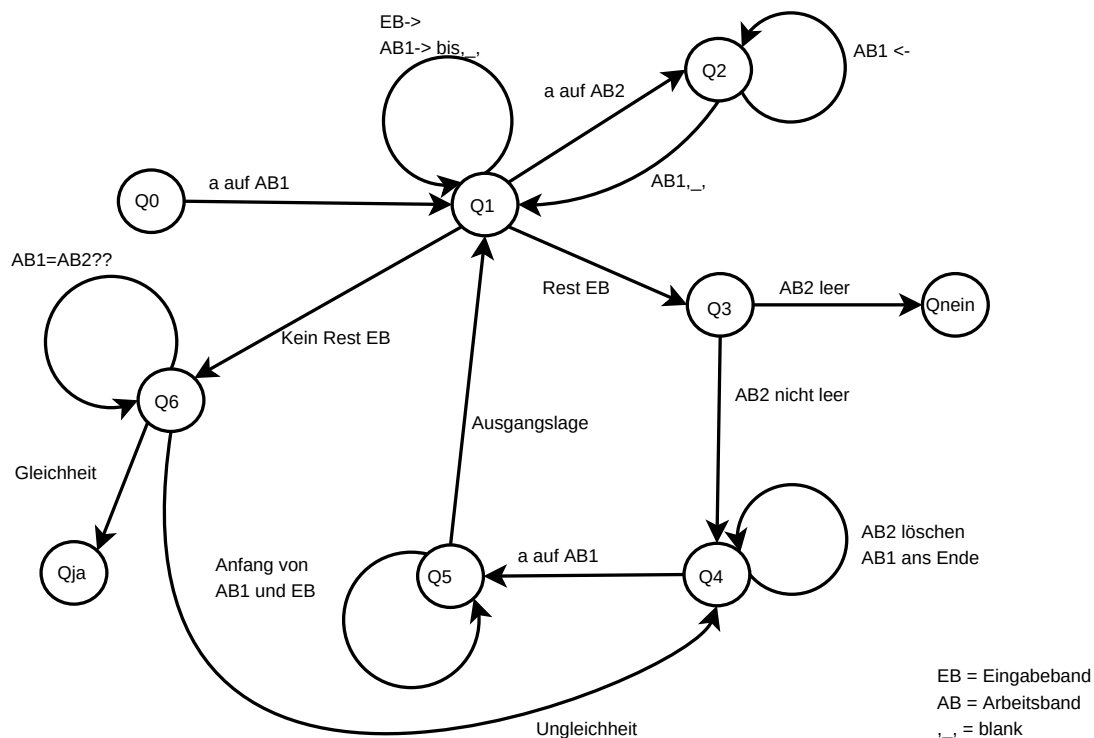


Prinzipielle Idee: Wir benutzen hier eine Turing-Maschine die neben dem Eingabeband noch zwei Arbeitsbänder nutzt. Um Herauszufinden ob x eine Quadratzahl ist, sucht man eine Zahl n für die gilt: $n * n = x$. In unserem Fall steht a^x auf dem Eingabeband und der Lesekopf steht am linken Ende des Eingabewortes. Auf das erste Arbeitsband wird dann a^n geschrieben. Auf das zweite Arbeitsband wird nun jeder n -te Buchstabe des Eingabewortes kopiert. Steht nach Abarbeiten des Eingabebandes auf den Arbeitsbändern dasselbe, so haben wir unser a^n gefunden und x ist eine Quadratzahl. Die Turing-Maschine geht dann in einen akzeptierenden Endzustand.

Was ist zu beachten? 1. Ist das letzte kopierte a nicht das letzte a der Eingabe, d. h. auf dem Eingabeband stehen noch m a 's mit $m < n$ so kann x nicht ganzzahlig durch n geteilt werden und somit kann x nicht n^2 sein. In diesem Fall werden die beiden Arbeitsbänder gelöscht und n inkrementiert.

2. Wird a^n auf Arbeitsband 1 länger als a^x , d. h. auf Arbeitsband 2 wird kein a kopiert, so wird endgültig nicht akzeptiert, da x keine Quadratzahl sein kann.

Skizze des Übergangsdiagramms:



Formale Definition der Turing Maschine: $M = (Q, \Sigma, \Gamma, s, F, \Delta)$ mit $Q = (Q_0, \dots, Q_6, q_{ja}, q_{nein})$, $\Sigma = (B, a)$. Hierbei steht B für ein Blank.

$\Gamma = (a)s = Q_0 F = q_{ja} \Delta \subset (Q \times \Sigma \times \Gamma^2) \times (Q \times K_{EB} \times (\Gamma_{AB1}, \Gamma_{AB2}) \times (K_{AB1}, K_{AB2}))$

Wir nehmen an, die Turing Maschine steht am Anfang am linken Ende der Eingabe.

Theoretische Informatik (WS04)

Prof. Dr. Raimund Seidel

Wei Ding, Dierk Johannes

Lösungsvorschlag zu Aufgabenblatt 7



$\Delta = (Q_0, a, B, B) \longrightarrow (Q_1, \downarrow, a, \epsilon, \downarrow, \downarrow)$
 $(Q_1, a, a, B) \longrightarrow (Q_1, \rightarrow, \epsilon, \epsilon, \rightarrow, \downarrow)$
 $(Q_1, a, B, B) \longrightarrow (Q_2, \leftarrow, \epsilon, a, \leftarrow, \rightarrow)$
 $(Q_2, a, a, B) \longrightarrow (Q_2, \downarrow, \epsilon, \epsilon, \leftarrow, \downarrow)$
 $(Q_2, a, B, B) \longrightarrow (Q_1, \downarrow, \epsilon, \epsilon, \rightarrow, \downarrow)$
 $(Q_1, B, a, B) \longrightarrow (Q_3, \leftarrow, \epsilon, \epsilon, \downarrow, \leftarrow)$
 $(Q_3, a, a, B) \longrightarrow (q_{nein})$
 $(Q_3, a, a, a) \longrightarrow (Q_4, \leftarrow, \epsilon, B, \downarrow, \leftarrow)$
 $(Q_4, a, a, a) \longrightarrow (Q_4, \leftarrow, \epsilon, B, \downarrow, \leftarrow)$
 $(Q_4, a, a, B) \longrightarrow (Q_4, \downarrow, \epsilon, \epsilon, \rightarrow, \downarrow)$
 $(Q_4, a, B, B) \longrightarrow (Q_5, \leftarrow, a, \epsilon, \leftarrow, \downarrow)$
 $(Q_5, B, a, B) \longrightarrow (Q_5, \downarrow, \epsilon, \epsilon, \leftarrow, \downarrow)$
 $(Q_5, B, B, B) \longrightarrow (Q_1, \rightarrow, \epsilon, \epsilon, \rightarrow, \downarrow)$
 $(Q_1, B, B, B) \longrightarrow (Q_6, \downarrow, \epsilon, \epsilon, \leftarrow, \leftarrow)$
 $(Q_6, B, a, a) \longrightarrow (Q_6, \downarrow, \epsilon, B, \leftarrow, \leftarrow)$
 $(Q_6, B, B, B) \longrightarrow (Q_{ja})$
 $(Q_6, B, a, B) \longrightarrow (Q_4, \leftarrow, \epsilon, \epsilon, \rightarrow, \downarrow)$
 $(Q_6, B, B, a) \longrightarrow (Q_4, \leftarrow, \epsilon, B, \rightarrow, \leftarrow)$

Teil b)

Es sei $L_p = \{a^n \mid n \in \mathbb{N} \text{ und } n \text{ ist eine Primzahl}\}$.

Idee:

Die Turing-Maschine soll nacheinander prüfen, ob n durch $2, 3, \dots, n-1$ teilbar ist. (Es genügt bis $\lfloor \sqrt{n} \rfloor$ zu testen). Falls n durch eins dieser i teilbar ist, geht die Maschine in den Zustand „nein“ über und hält, falls sie nach allen Tests keinen Teiler ausfindig gemacht hat, und beim Test, ob sie i nochmal incrementieren muss feststellt, daß sie bereits auf Teilbarkeit durch $n-1$ getestet hat, geht sie in den Zustand „ja“ über und hält.

Dazu benötigt sie folgende Teilmaschinen:

- INIT

Dieser Teilautomat schreibt $aa\#$ auf das Arbeitsband und lässt den Eingabekopf auf Anfang stehen. Die Übergänge sehen wie folgt aus:

$\delta(INIT, \epsilon, \epsilon) \rightarrow (INIT_1, N, a, R) \quad \delta(INIT_1, \epsilon, \epsilon) \rightarrow (INIT_2, N, a, R) \quad \delta(INIT_2, \epsilon, \epsilon) \rightarrow (INIT_3, N, \#, R)$ Wobei mit Erreichen des Zustandes „ $INIT_3$ “ der Automat fertig ist.



- RÜCKA

Dieser Teilautomat bewegt den Arbeitskopf zurück auf das erste a , der Eingabekopf bleibt stehen. Voraussetzung für diesen Teilautomat ist, daß der Arbeitskopf sich nicht ausserhalb des bereits beschriebenen Bandes befindet. Die Übergänge sehen wie folgt aus:

$\delta(RCKA, \varepsilon, a) \rightarrow (RCKA_1, N, a, L) \delta(RCKA, \varepsilon, \#) \rightarrow (RCKA_1, N, \#, L) \delta(RCKA, \varepsilon, \bar{b}) \rightarrow (RCKA_2, N, \bar{b}, R) \delta(RCKA_1, \varepsilon, a) \rightarrow (RCKA_1, N, a, L) \delta(RCKA_1, \varepsilon, \bar{b}) \rightarrow (RCKA_2, N, \bar{b}, R) \delta(RCKA_2, \varepsilon, a) \rightarrow (RCKA_3, N, a, N) \delta(RCKA_2, \varepsilon, \bar{b}) \rightarrow (RCKA_2, N, \bar{b}, R)$ Wobei mit Erreichen des Zustandes „ $RCKA_3$ “ der Automat fertig ist. Im weiteren werde ich einmal die Schreibweise RÜCKA(a) verwenden, die bedeuten soll, daß bei Erreichen des Zustandes „ $RCKA_3$ “ in Zustand a übergegangen werden soll.

- RÜCKE

Dieser Teilautomat bewegt den Eingabekopf zurück auf das erste a , der Arbeitskopf bleibt stehen; auch hier ist Voraussetzung, daß der Eingabekopf nicht ausserhalb des beschriebenen Bandes steht.

$\delta(RCKE, a, \varepsilon) \rightarrow (RCKA, L, \varepsilon, N) \delta(RCKE, \bar{b}, \varepsilon) \rightarrow (RCKE_1, R, \varepsilon, N)$ Wobei mit Erreichen des Zustandes „ $RCKE_1$ “ der Automat fertig ist.

- INC

Dieser Teilautomat incrementiert die Anzahl der a auf dem Arbeitsband und verschiebt das $\#$. Bei diesem Automat muss der Arbeitskopf sich anfänglich in den a befinden.

$\delta(INC, \varepsilon, a) \rightarrow (INC, N, a, R) \delta(INC, \varepsilon, \#) \rightarrow (INC_1, N, a, R) \delta(INC_1, \varepsilon, a) \rightarrow (INC_2, N, \#, N)$ Wobei mit Erreichen des Zustandes „ INC_2 “ der Automat fertig ist.

- TEST

Dieser Automat testet, ob die Anzahl der a auf dem Arbeitsband (im folgenden i genannt) bereits größer ist als die Hälfte der Anzahl der a auf dem Eingabeband. Hierzu müssen beide Köpfe sich auf dem jeweils ersten a befinden.

$\delta(TEST, a, a) \rightarrow (TEST_1, R, a, N) \delta(TEST, a, \#) \rightarrow (TEST_3, N, \#, L) \delta(TEST, \bar{b}, a) \rightarrow (TEST_2, L, a, N) \delta(TEST, \bar{b}, \#) \rightarrow (TEST_3, L, \#, L) \delta(TEST_1, a, a) \rightarrow (TEST, R, a, R) \delta(TEST_1, \bar{b}, a) \rightarrow (TEST_2, L, a, N)$ Wobei mit Erreichen des Zustandes „ $TEST_2$ “ der Test bestätigt wird, mit „ $TEST_3$ “ verworfen wird. Im weiteren werde ich die Schreibweise TEST(a, b) verwenden, wobei mit a die Sprungmarke gemeint ist, die bei $i > \lfloor \frac{n}{2} \rfloor$ verwendet wird, mit b die Sprungmarke für den gegenteiligen Fall.

- DIV

Dieser Teilautomat testet die Teilbarkeit von n (die Anzahl der a auf dem Eingabeband) durch i (die Anzahl der a auf dem Arbeitsband). Voraussetzung ist, daß beide Köpfe sich auf dem ersten a befinden.

$\delta(DIV, a, a) \rightarrow (DIV, R, a, R) \delta(DIV, a, \#) \rightarrow (RCKA(DIV), N, \#, L) \delta(DIV, \bar{b}, a) \rightarrow (DIV_1, L, a, N) \delta(DIV, \bar{b}, \#) \rightarrow (DIV_2, L, \#, L)$ Wobei das Erreichen des Zustandes „ DIV_1 “ n nicht durch i teilbar ist, bei Erreichen des Zustandes „ DIV_2 “ die Teilbarkeit vorliegt. DIV(a, b) meint: springe zu a , wenn n durch i teilbar, zu b sonst.

Die Hauptmaschine setzt sich nun wie folgt zusammen: 1. INIT



2. RÜCKA
3. DIV(12,4)
4. RÜCKA
5. RÜCKE
6. INC
7. RÜCKA
8. TEST(13,9)
9. RÜCKA
10. RÜCKE
11. gehe zu 3
12. n ist keine Primzahl, STOP
13. n ist Primzahl, STOP

Aufgabe 4

Es ist $\overline{L_p} = \{a^n \mid n \in \mathbb{N}, n \text{ ist keine Primzahl}\}$. Es existiert also eine Zahl $k \in \mathbb{N}$, die n teilt.

Idee man nehme die TM von Aufgabe 4b, ersetze aber INIT durch folgende Teilmaschine: $\delta(\text{INIT}, \varepsilon, \varepsilon) \rightarrow (\text{INIT}, N, a, R)$ $\delta(\text{INIT}, \varepsilon, \varepsilon) \rightarrow (\text{INIT}_1, N, a, R)$ $\delta(\text{INIT}_1, \varepsilon, \varepsilon) \rightarrow (\text{INIT}_2, N, \#, N)$

Wobei nun das Ende der Maschine durch Erreichen des Zustandes „ INIT_2 “ signalisiert wird. Die Idee dahinter ist, daß diese TM den Teiler errät und direkt mit diesem in Schritt 3 feststellt, daß n nicht prim ist .