



Aufgabe 1

a)

L_1 ist kontextfrei.

Wir schauen zunächst mal die Sprache L_1 genau an. Wenn wir einen Automaten für diese Sprache bauen möchten, müssen wir uns irgendwie die Anzahl der gelesenen a's merken können. Dafür bräuchten wir bei einem EA unendlich viele Zustände, bei einem NKA können wir dies mit dem Keller realisieren.

Wir vermuten also: Die gegebene Sprache ist eine NKA- Sprache, aber keine DEA- Sprache. Dies wollen wir nun beweisen.

Beweis, dass L_1 keine DEA-Sprache ist (mit Fortsetzungssprachen)

Betrachten wir Präfixe von der Form $a^{7n}b$. Wenn wir aus diesen ein Wort der Sprache L_1 erzeugen möchten, müssen wir das Wort b^{n-1} anhängen, und erhalten somit also das Wort $a^{7n}b^n$, was ein Wort der Sprache L_1 ist.

Es gilt also $\forall n \in \mathbb{N}: F(a^{7n}b) = \{b^{n-1}\}$

Da diese Fortsetzungssprachen für beliebige $m, n \in \mathbb{N}$ jeweils paarweise verschieden sind, ist die Teilmenge $\{F(a^{7n}b) | n \in \mathbb{N}\}$ der Menge aller Fortsetzungssprachen unendlich groß, und damit auch die Menge aller Fortsetzungssprachen. Daraus folgt nach dem Satz von Myhill & Nerode: L ist keine DEA-Sprache.

Wem das zu langweilig war, kann das ganze alternativ auch mit dem Pumping-Lemma beweisen:

Beweis, dass L_1 keine DEA-Sprache ist (mit Pumping-Lemma)

Sei $N \in \mathbb{N}$.

Weil es so schön war, betrachten wir wieder das Wort $x = a^{7N}b^N$.

Sei uvw eine beliebige Unterteilung von x mit $|uv| \leq N$ und $|v| \geq 1$. Dann gilt: $u = a^k, v = a^l$ und $w = a^mb^N$, mit $k, l, m \in \mathbb{N}$ und $k + l + m = 7N$ sowie $l \geq 1$.

Wähle ich nun $i = 0$, so gilt:

$$x' = a^k a^{l \cdot 0} a^m b^N = a^{k+m} b^N.$$

Es gilt: $k + m = 7N - l \leq 7N - 1$, also $\lfloor (k + m)/7 \rfloor < N$.

Damit gilt: $w' \notin L_1$. Damit ist die Pumping-Eigenschaft nicht erfüllt, und daraus folgt, dass L_1 keine DEA-Sprache ist.

Konstruktion eines NKA für L_1 :

Theoretische Informatik (WS05)

Prof. Dr. Raimund Seidel

Wei Ding

Lösungsvorschlag zu Aufgabenblatt 5



Informelle Beschreibung: Wir lesen zunächst die a's und schreiben nach jedem 7. gelesenen a ein B in den Keller. Sobald wir das erste b lesen, fangen wir an, die Bs aus dem Keller zu nehmen. Wir akzeptieren das Wort genau dann wenn wir ab jetzt nur noch b's lesen und der Keller beim Lesen des letzten Zeichens leer wird.

Formal: $M_1 = \{\Sigma, \Gamma, \epsilon, Q, a_1, F, \Delta\}$

$\Sigma = \{a, b\}^*$

$\Gamma = \{B, \epsilon\}$

$Q = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, b_1, f\}$

Übergangsregeln:

$\forall i \in \{1, \dots, 6\} : (a_i, U, a, U, a_{i+1}) \in \Delta$

$(a_7, U, a, BU, a_1) \in \Delta$

$\forall i \in \{1, \dots, 7\} : (a_i, BU, b, U, b_1) \in \Delta$

$(b_1, BU, b, U, b_1) \in \Delta$

$(b_1, \epsilon, \epsilon, \epsilon, f) \in \Delta$

b)

L_2 ist regulär, weil folgender Automat L_2 akzeptiert:

Hier Bild einfüegen

Idee: Man benötigt 7 Zustände, um $n \bmod 7$ zu speichern. Danach vergleicht man die Anzahl der b's mit der gespeicherten Zahl. Da für $m > 6$ kein Endzustand mehr erreicht werden kann, kommt man mit endlich vielen Zuständen aus.

c)

L_3 ist kontextfrei.

Auch hier können wir uns wieder mit einem NKA im Keller die Anzahl der gelesenen a's und b's merken, was bei einem DEA nicht funktionieren würde. Wir wollen also beweisen, dass L_3 eine NKA- aber keine DEA-Sprache ist.

Beweis, dass L_3 keine DEA-Sprache ist (mit Fortsetzungssprachen)

Betrachten wir Präfixe von der Form $a^m b^n c$. Durch Anhängen von c^{m+n-1} machen wir daraus ein Wort aus L_3 .

Wir stellen also fest: $\forall m, n \in \mathbb{N} : F(a^m b^n c) = \{c^{m+n-1}\}$

Auf diese Weise erhalten wir für jedes $m + n =: k \in \mathbb{N}$ eine Fortsetzungssprache. Da diese jeweils paarweise verschieden sind, haben wir also unendlich viele Fortsetzungssprachen, und



daraus folgt nach Myhill & Nerode: L_3 ist keine DEA-Sprache.

Beweis, dass L_3 keine DEA-Sprache ist (mit Pumping Lemma)

Sei $N \in \mathbb{N}$.

Wir betrachten das Wort $a^N b^m c^{N+m}$.

Sei uvw eine beliebige Unterteilung von x mit $|uv| \leq N$ und $|v| \geq 1$. Wir stellen fest: Wegen $|uv| \leq N$ gilt: v besteht nur aus a 's. Sei k die Anzahl der a 's in v .

Wählen wir $i = 0$, so gilt:

$$x' = uv^0w = a^{N-k} b^m c^{N+m}.$$

Da $k \geq 1$, gilt: $(N - k) + m \neq N + m$, und damit ist $w' \notin L_3$.

Damit ist die Pumping-Eigenschaft nicht erfüllt, und daraus folgt, dass L_3 keine DEA-Sprache ist.

Konstruktion eines NKA für L_3 :

Informelle Beschreibung: Wir lesen zunächst die a 's und schreiben jeweils ein Zeichen in den Keller. Sobald wir das erste b lesen, akzeptieren wir ab jetzt nur noch b 's und schreiben ebenfalls jeweils ein Zeichen in den Keller. Sobald wir das erste c lesen, akzeptieren wir ab jetzt nur noch c 's und nehmen jeweils ein Zeichen aus dem Keller. Wir akzeptieren das Wort genau dann wenn der Keller beim Lesen des letzten Zeichens leer wird.

Formal: $M_1 = \{\Sigma, \Gamma, \epsilon, Q, s, F, \Delta\}$

$$\Sigma = \{a, b, c\}^*$$

$$\Gamma = \{C, \epsilon\}$$

$$Q = \{s, 1, 2, 3, f\}$$

Übergangsregeln:

$$(1, U, a, CU, 1) \in \Delta$$

$$(1, U, b, CU, 2) \in \Delta$$

$$(1, CU, c, U, 3) \in \Delta$$

$$(2, U, b, CU, 2) \in \Delta$$

$$(2, CU, c, U, 3) \in \Delta$$

$$(3, CU, c, U, 3) \in \Delta$$

$$(3, \epsilon, \epsilon, \epsilon, f) \in \Delta$$

d)

L_4 ist nicht kontextfrei (und damit auch nicht regulär). Beweis mit Pumping Lemma:

Theoretische Informatik (WS05)

Prof. Dr. Raimund Seidel

Wei Ding

Lösungsvorschlag zu Aufgabenblatt 5



Sei $N \in \mathbb{N}$ frei vorgegeben.

Wähle $z = a^N b^N c^{N^2}$

Dann gilt für jede Zerlegung $x = uvwxy$:

Da irgendwelche Buchstaben auf jeden Fall aufgepumpt werden ($|vx| > 0$), müssen die c 's auch auf jeden Fall gepumpt werden und damit auch entweder die a 's oder die b 's. Offensichtlich können v und w nur maximal eine Sorte von Buchstaben enthalten. Aus $|vwx| \leq N$ folgt außerdem, dass v keine a 's enthalten kann. Es folgt:

$v = b^p$ und $x = c^q$ mit $1 \leq p < N$ und $1 \leq q < N$.

Sei nun $i = 2$:

$$\Rightarrow z^i = uv^iwx^iy = a^N b^{N+p} c^{N^2+q}$$

Damit $z^i \in L_4$ muss gelten:

$$(N^2 + q) = N \times (N + p)$$

$$\Leftrightarrow (N^2 + q) = (N^2 + N \times p)$$

$$\Leftrightarrow q = N \times p$$

$\Rightarrow$ Widerspruch zur Voraussetzung, dass $q < N$

Aufgabe 2

a)

Sei $M = (\Sigma, Q, \Gamma, s, \epsilon, F, \Delta)$ der NKA für L und $M' = (\Sigma', Q', \Gamma', s', \epsilon', F', \Delta')$ der NKA für L' .

In der Vorlesung wurde gezeigt, dass die Akzeptanz eines Wortes durch Verweilen des Automaten nach komplettem Einlesen des Eingabebandes in einem Endzustand und durch die komplette Entleerung des Kellers äquivalent sind. Von daher können wir davon ausgehen, dass wenn M und M' das Eingabeband abgearbeitet haben, sie sich in einem Endzustand befinden. Nun kann man relativ einfach einen NKA $M'' = (\Sigma'', Q'', \Gamma'', s'', \epsilon'', F'', \Delta'')$ konstruieren, der $L \bullet L'$ akzeptiert:

$$\Sigma'' = \Sigma \cup \Sigma'$$

$$Q'' = Q \cup Q' \cup \{fclear\}$$

$$\Gamma'' = \Gamma \cup \Gamma'$$

$$s'' = s$$

$$\epsilon'' = \epsilon$$

$$F'' = F'$$

$$\Delta'' = \Delta \cup \Delta'$$

$$\cup \{(f, A, \epsilon, A, fclear) | f \in F, A \in \Gamma\}$$

$$\cup \{(fclear, A, \epsilon, \epsilon, fclear) | A \in (\Gamma \setminus \{\epsilon\})\}$$

$$\cup \{(fclear, \epsilon, \epsilon, \epsilon, s') | f \in F\}$$



Auf deutsch: Man gibt dem Automaten die Möglichkeit aus jedem Endzustand in einen Zustand überzugehen, wo der Keller bis auf das Kellersymbol geleert wird. Von diesem Zustand kann der Automat dann mit leerem Keller weiter zum Startzustand des zweiten Automaten gehen, auf das dieser das Wort aus L' verarbeitet. M'' akzeptiert nun alle Konkatenierungen von Wörtern als L und L' .

b)

Sei wieder $M = (\Sigma, Q, \Gamma, s, \epsilon, F, \Delta)$ der NKA für L .

Nun ist der Automat $M' = (\Sigma, Q \cup \{fclear\}, \Gamma, s, \epsilon, F, \Delta')$ der NKA für L^* :

$$\Delta' = \Delta$$

$$\cup \{(f, A, \epsilon, A, fclear) | f \in F, A \in \Gamma\}$$

$$\cup \{(fclear, A, \epsilon, \epsilon, fclear) | A \in (\Gamma \setminus \{\epsilon\})\}$$

$$\cup \{(fclear, \epsilon, \epsilon, \epsilon, s) | f \in F\}$$

Man zieht diesmal also statt einer ϵ -Kante zum Startzustand eines weiteren Automaten M' eine ϵ -Kante zum Startzustand von M zurück. Da wir den Keller wieder vorher bis auf das Kellersymbol leermachen, kann der Automat wieder von neuem beginnen, und akzeptiert somit L^* .

Aufgabe 3

a)

L_5 ist nicht regulär. Beweis:

Betrachte $X = \{a, b\}^*$ und führe den Beweis über das Pumping Lemma:

Sei $N \in \mathbb{N}$ frei vorgegeben.

Wähle $z = a^N b^{2N} a^N \in L_5$

Dann gilt für jede Unterteilung von $z = uvw$:

v enthält nur a 's, aber mindestens eins. Daher gilt für jedes $i \geq 2$: $uv^i w \notin L_5$, da die Symmetrie verloren ist.

Andererseits ist L_5 aber kontextfrei: Es gibt einen endlichen Automaten M , der X akzeptiert. Dann kann man einen NKA bauen, der die gleichen Zustände hat, wie M . Dieser NKA liest das halbe Wort ein und legt die gelesenen Zeichen auf den Keller. Danach liest er vom Eingabeband und vom Keller parallel und läuft nur weiter, wenn beide Zeichen übereinstimmen. Zusätzlich müssen wir dafür sorgen, dass der Automat nicht mehr auf den Keller schreibt, sobald er einmal davon gelesen hat. Dann akzeptiert folgender NKA die Sprache L_5 :

$$\Sigma = \Sigma_M$$

$$\Gamma = \Sigma$$



$$Q = Q_M \times \{w, r\}$$

$$s = (s_M, w)$$

$$F = F_M \times \{r\}$$

$$\Delta = \{((q, w), k, a, ka, (p, w)) \mid (q, a, p) \in \Delta_M\} \cup \{((q, w), k, a, ka, (p, r)) \mid (q, a, p) \in \Delta_M\} \cup \{((q, r), a, a, \epsilon, (p, r)) \mid (q, a, p) \in \Delta_M\}$$

Γ

b)

L_6 ist regulär, da man einen NEA konstruieren kann, der L_6 akzeptiert.

Idee: Sei M der Automat, der X akzeptiert. O.B.d.A. nehmen wir an, dass M minimal ist (das ist kein notwendiges Kriterium, erleichtert aber das Verständnis) und nur einen Endzustand f besitzt (falls nicht: erzeuge Super-Endzustand f und lege ϵ -Kanten von den alten Endzuständen zu f).

Ein Wort w ist genau dann in L_6 , wenn es einen Pfad von s über w zu einem Zustand a gibt, und von dort einen Pfad über w^R zu f .

Konstruiere nun zunächst durch Umkehren der Übergangsregeln wie folgt einen Automaten M_R , der X^R akzeptiert:

$$\Sigma_R = \Sigma_M$$

$$Q_R = Q_M$$

$$s_R = f_M$$

$$F_R = \{s_M\}$$

$$\Delta_R = \{(p, a, q) \mid (q, a, p) \in \Delta_M\}$$

Wenn M_R nach dem Einlesen von w im Zustand q ist, dann gibt es in M offensichtlich einen Pfad von q über w^R nach f . Wenn es nun in M auch einen Pfad von s über w nach q gibt, dann liegt w in L_6 . Lasse nun beide Automaten parallel laufen. Akzeptiere w genau dann, wenn beide Automaten nach dem Lesen von w im gleichen Zustand sind. Ein entsprechender NEA M' sähe dann so aus:

$$\Sigma' = \Sigma_M$$

$$Q' = (Q_M \times Q_R)$$

$$s' = (s_M, s_R)$$

$$F' = \{(q, q)\}$$

$$\Delta' = \{((p, q), a, (p', q')) \mid (p, a, p') \in \Delta_M \wedge (q, a, q') \in \Delta_R\}$$

Aufgabe 4

Nein, die DKA-Sprachen sind unter Konkatenation und dem Kleene'schen Abschluss *nicht* abgeschlossen.

Beweisidee: Obwohl eine NKA-Sprache L eine nicht-DKA-Sprache sein kann, kann L die Vereinigung von mehreren DKA-Sprachen $L_1, L_2, \dots, L_k$ sein. Wenn jetzt jede von diesen



DKA-Sprachen mit einem eindeutigen Präfix $p_i, 1 \leq i \leq k$ konkateniert wird, dann ist ihre Vereinigung eine DKA-Sprache: ein DKA bestimmt eindeutig mittels des eindeutigen Präfixes zu welcher der Sprachen die Eingabe gehören könnte und simuliert den entsprechenden DKA. Allerdings kann die Eindeutigkeit der Präfixe durch Konkatenation mit einer anderen DKA-Sprache verletzt werden, so dass das Ergebnis dieser Konkatenation keine DKA-Sprache mehr ist.

Beweis: Die Sprachen $L_1 = \{a^i b^j c^k | i, j, k \in \mathbb{N}, i \neq j\}$ und $L_2 = \{a^i b^j c^k | i, j, k \in \mathbb{N}, j \neq k\}$ sind DKA-Sprachen. Ein DKA, der L_1 entscheidet, speichert die a 's auf dem Keller und beim Einlesen jedes b entfernt das oberste a . So stellt der DKA fest, ob die i a 's von genau i b 's gefolgt sind (und im positiven Fall lehnt ab). Schließlich wird überprüft, ob der Suffix aus lauter c 's besteht. Analog kann auch L_2 von einem DKA entschieden werden. Die Sprache $L_3 = \overline{L(a^* b^* c^*)}$ ist regulär, da sie das Komplement einer regulären Sprache ist, und, folglich eine DKA-Sprache.

Jetzt, die Vereinigung $L_1 \cup L_2 \cup L_3 = \overline{L_{abc}}$ ist bekanntlich aus der Vorlesung keine DKA-Sprache, aber die Sprache $L_4 = (\{\$ \} \cdot L_1) \cup (\{\$\$ \} \cdot L_2) \cup (\{\$\$\$ \} \cdot L_3)$ ist doch eine DKA-Sprache: wenn die Eingabe mit $\$\$\$$ beginnt, dann wird ein DKA für L_3 simuliert, sonst wenn die Eingabe mit $\$\$$ beginnt, dann wird ein DKA für L_2 simuliert, sonst wenn die Eingabe mit $\$$ beginnt, dann wird ein DKA für L_1 simuliert.

Jetzt, die Sprache $L_5 = \{\$, \$\$, \$\$\}$ ist regulär, und folglich eine DKA-Sprache. Wären die DKA-Sprachen unter Konkatenation abgeschlossen, dann sollte auch $L_6 = L_5 \cdot L_4$ DKA-Sprache sein.

Sei $L_7 = \{\$\$\$\$ \} \cdot \{a, b, c\}^*$. L_7 ist regulär. Da die DKA-Sprachen unter Schnitt mit regulären Sprachen abgeschlossen sind, sollte dann auch $L_8 = L_7 \cap L_6 = \{\$\$\$\$ \} \cdot (L_1 \cup L_2 \cup L_3) = \{\$\$\$\$ \} \cdot \overline{L_{abc}}$ eine DKA-Sprache sein.

Ist L_8 eine DKA-Sprache, dann existiert ein DKA M_{L_8} , der sie entscheidet. Beim Einlesen der ersten drei Symbole ($\$\$\$$) vom Startzustand q_s aus, modifiziert M_{L_8} den Keller und übergeht in einen neuen Zustand q_n , immer auf die selbe Weise, da M_{L_8} deterministisch ist. Betrachten wir jetzt den DKA M'_{L_8} , der mit kleinen Veränderungen aus M_{L_8} konstruiert werden kann: M'_{L_8} verfügt über einen neuen Startzustand q'_s und beim ersten Schritt liest die Eingabe nicht ein (d.h. der Kopf bleibt auf dem ersten Symbol), modifiziert den Keller auf die selbe Weise wie M_{L_8} beim Einlesen der ersten $\$\$\$$, und führt einen ϵ -Übergang in den Zustand q_n . Dann benimmt sich identisch wie M_{L_8} . M'_{L_8} ist deterministisch wie M_{L_8} und erkennt die Sprache $L_1 \cup L_2 \cup L_3 = \overline{L_{abc}}$ die bekanntlich keine DKA-Sprache ist: Widerspruch. Daraus folgt, dass die DKA-Sprachen unter Konkatenation *nicht* abgeschlossen sind.

Man kann leicht sehen, dass die Sprache $L_9 = \{\$ \} \cup L_4$ eine DKA-Sprache ist, wie L_4 . Da L_9 das Wort $\$$ enthält, ist der Kleene'sche Abschluss $(L_9)^*$ eine Obermenge von L_8 . Wäre jetzt L_9^* eine DKA-Sprache, dann so wäre auch $L_9^* \cap L_7 = L_8$, die, wie bereits bewiesen, keine DKA-Sprache ist. Daraus folgt, dass die DKA-Sprachen unter dem Kleene'schen Abschluss *nicht* abgeschlossen sind.