Lösungsvorschlag zur Vorlesung Formale Sprachen, Automaten und Prozesse

Aufgabe T23

Es ist sofort zu sehen, dass die Sprache $L(G)$ das leere Wort ε nicht enthalten kann. Wir können also direkt damit beginnen G in CNF zu bringen. Dazu ersetzen wir alle Terminalsymbole durch Nichtterminale.

$$\begin{aligned}
&S \to R_a A R_b \mid R_a R_b\\
&A \to S \mid AA\\
&R_a \to a\\
&R_b \to b
\end{aligned}$$

Anschließend führen wir Übergangs–Nichtterminale ein, um Ketten von mehr als zwei Nichtterminalen aufzulösen. Hier gibt es nur eine solche Kette.

$$\begin{aligned}
&S \to R_a C_{AR_b} \mid R_a R_b\\
&A \to S \mid AA\\
&C_{AR_b} \to A R_b\\
&R_a \to a\\
&R_b \to b
\end{aligned}$$

Wir sehen, dass wir fast Chomsky–Normalform erreicht haben. Lediglich die Kettenregel $A \to S$ muss noch eliminiert werden. Wir gehen vor, wie auf den Folien beschrieben.

$$\begin{aligned}
&S \to R_a C_{AR_b} \mid R_a R_b\\
&A \to R_a C_{AR_b} \mid R_a R_b \mid AA\\
&C_{AR_b} \to A R_b\\
&R_a \to a\\
&R_b \to b
\end{aligned}$$

Diese Grammatik ist in CNF.

Wir widmen uns nun der Greibach–Normalform. Alle Nichtterminale in der Grammatik sind erreichbar und produktiv. Wir können also mit dem Algorithmus beginnen. Dazu benötigen wir eine Totalordnung $<$ auf der Menge der Nichtterminalsymbole

$$N = \{S, A, R_a, R_b, C_{AR_b}\},$$

d.h. eine Ordnung, bei der je zwei unterschiedliche Elemente $X \neq Y \in N$ entweder $X < Y$ oder $Y < X$ erfüllen. Hier wählen wir folgende Totalordnung auf N:

$$S < A < C_{AR_b} < R_a < R_b$$

Der Algorithmus aus der Vorlesung bringt zuerst alle Regeln der Form $X \to Y\alpha$ mit $\alpha \in N^*$ und $X \geq Y$ in eine Form $X \to Z\beta$, $\beta \in N^*$ mit $X < Z$ $(\star)$, $X, Y, Z \in N$. Betrachten wir unsere Grammatik sehen wir, dass die Regeln für S bereits die gewünschte Form haben, die Regeln für A aber nicht. Wir lösen die linksrekursive Regel AA wie in der Vorlesung beschrieben auf. Dazu führen wir ein neues Nichtterminal–Symbol Z_A ein. Wir schreiben dieses ans Ende

$$\begin{array}{l}
S \to R_a C_{AR_b} \mid R_a R_b \\
A \to R_a C_{AR_b} \mid R_a R_b \mid R_a C_{AR_b} Z_A \mid R_a R_b Z_A \\
C_{AR_b} \to A R_b \\
R_a \to a \\
R_b \to b \\
\hline
Z_A \to A \mid A Z_A
\end{array}$$

Da $A < C_{AR_b}$ gilt, muss auch die Regel $C_{AR_b} \to AR_b$ angepasst werden.

$$\begin{array}{l}
S \to R_a C_{AR_b} \mid R_a R_b \\
A \to R_a C_{AR_b} \mid R_a R_b \mid R_a C_{AR_b} Z_A \mid R_a R_b Z_A \\
C_{AR_b} \to R_a C_{AR_b} R_b \mid R_a R_b R_b \mid R_a C_{AR_b} Z_A R_b \mid R_a R_b Z_A R_b \\
R_a \to a \\
R_b \to b \\
\hline
Z_A \to A \mid A Z_A
\end{array}$$

Nun erfüllen alle Regeln die gewünschte Eigenschaft ($\star$). Die Regeln für die beiden größten Nichtterminale R_a und R_b entsprechen schon der GNF. Nun ersetzen wir in den Regeln von C_{AR_b} jeweils R_a und R_b durch ihre rechten Seiten. Danach wiederholen wir diesen Schritt in den Regeln von A und S.

$$\begin{array}{l}
S \to a C_{AR_b} \mid a R_b \\
A \to a C_{AR_b} \mid a R_b \mid a C_{AR_b} Z_A \mid a R_b Z_A \\
C_{AR_b} \to a C_{AR_b} R_b \mid a R_b R_b \mid a C_{AR_b} Z_A R_b \mid a R_b Z_A R_b \\
R_a \to a \\
R_b \to b \\
\hline
Z_A \to A \mid A Z_A
\end{array}$$

Die einzigen Regeln, die noch nicht der GNF entsprichen sind die für das neu eingeführte Nichtterminal. Aber hier können wir jetzt das führende Nichtterminal A in den rechten Seiten durch die rechten Seite der Regeln von A ersetzen.

$$\begin{array}{l}
S \to a C_{AR_b} \mid a R_b \\
A \to a C_{AR_b} \mid a R_b \mid a C_{AR_b} Z_A \mid a R_b Z_A \\
C_{AR_b} \to a C_{AR_b} R_b \mid a R_b R_b \mid a C_{AR_b} Z_A R_b \mid a R_b Z_A R_b \\
R_a \to a \\
R_b \to b \\
\hline
Z_A \to a C_{AR_b} \mid a R_b \mid a C_{AR_b} Z_A \mid a R_b Z_A \mid a C_{AR_b} Z_A \mid a R_b Z_A \mid a C_{AR_b} Z_A Z_A \mid a R_b Z_A Z_A
\end{array}$$

Wir können nun doppelte Regeln löschen.

$$\begin{array}{l}
S \to a C_{AR_b} \mid a R_b \\
A \to a C_{AR_b} \mid a R_b \mid a C_{AR_b} Z_A \mid a R_b Z_A \\
C_{AR_b} \to a C_{AR_b} R_b \mid a R_b R_b \mid a C_{AR_b} Z_A R_b \mid a R_b Z_A R_b \\
R_a \to a \\
R_b \to b \\
\hline
Z_A \to a C_{AR_b} \mid a R_b \mid a C_{AR_b} Z_A \mid a R_b Z_A \mid a C_{AR_b} Z_A Z_A \mid a R_b Z_A Z_A
\end{array}$$

Abschließend erhalten wir eine zur Grammatik G äquivalente Grammatik in GNF:

$S \to aC_{AR_b} \mid aR_b$
$A \to aC_{AR_b} \mid aR_b \mid aC_{AR_b}Z_A \mid aR_bZ_A$
$C_{AR_b} \to aC_{AR_b}R_b \mid aR_bR_b \mid aC_{AR_b}Z_AR_b \mid aR_bZ_AR_b$
$R_a \to a$
$R_b \to b$
$Z_A \to aC_{AR_b} \mid aR_b \mid aC_{AR_b}Z_A \mid aR_bZ_A \mid aC_{AR_b}Z_AZ_A \mid aR_bZ_AZ_A$

Aufgabe T24
Angenommen L sei regulär. Dann existiert ein n, so dass jedes Wort $z \in L$ mit $|z| \geq n$ in $z = uvwxy$ zerlegt werden kann mit

1. $|vwx| \leq n$,
2. $|vx| \geq 1$ und
3. $uv^iwx^iy \in L$ für alle $i \in \mathbf{N}_0$.

Sei $z = a^n\$a^n\$a^n$. Offensichtlich gilt $z \in L$. Falls vx mindestens ein \$ enthält, so ist $uv^0wx^0y = uwy$ offensichtlich nicht in L enthalten, da dieses Wort noch höchstens ein \$ enthält.
Wir können also davon ausgehen, daß vx kein \$ enthält. Für jede solche Zerlegung die (1) und (2) erfüllt, gilt aber daß vwx höchstens zwei der getrennten a-Blöcke überlappt, da $|vwx| \leq n$. Damit ist aber $uv^2wx^2y \notin L$, da mindestens einer der drei a-Blöcke genau die Länge n hat, während mindestens einer der beiden anderen Blöcke mindestens die Länge $n+1$ hat.

Aufgabe T25
Wir nutzen das Pumping-Lemma um zu zeigen, dass L nicht kontextfrei ist. Sei n beliebig, z ein Präfix von π und sei $uvwxy$ eine beliebige Zerlegung von z mit $|vwx| \leq n$ und $|vx| > 0$. Falls L kontextfrei wäre, dann wäre uv^iwx^iy ebenfalls ein Präfix von π für jedes $i \in \mathbf{N}$.
O.b.d.A sei $v \neq \varepsilon$ und v enthalte nicht das Symbol „.“, ansonsten ergibt sich sofort ein Widerspruch, wenn wir v pumpen. Dann muss aber die Dezimaldarstellung von π periodisch sein: Da ein Präfix eines Präfixes der Dezimaldarstellung immer noch ein Präfix ist, ist $uv^i \in L$ für jedes $i \in \mathbf{N}$. Also kommt erst u und danach wiederholt sich v immer und immer wieder. Damit wäre die Dezimaldarstellung von π aber periodisch. Dies kann jedoch nicht sein, da π keine rationale Zahl ist. Also ist L nicht kontextfrei.

Aufgabe H19 (5+5 Punkte)
Zuerst bringen wir die Grammatik in Chomsky-Normalform und erhalten

$$S \to AX \mid a, \quad X \to BC, \quad A \to SR_a, \quad B \to SR_b, \quad C \to R_cS$$

$$R_a \to a, \quad R_b \to b, \quad R_c \to c$$

Nun kümmern wir uns um S und erhalten zunächst

$$S \to SR_aX \mid a, \quad X \to BC, \quad A \to SR_a, \quad B \to SR_b, \quad C \to cS$$

und dann

$$S \to aZ \mid a, \quad Z \to aX \mid aXZ, \quad X \to BC, \quad A \to SR_a, \quad B \to SR_b, \quad C \to cS.$$

Abschließend betrachten wir A, B und X. Für A und B ist die Auflösung einfach, da die rechten Seiten mit S beginnen, S aber schon konvertiert ist. Danach kann auch X umgewandelt werden, und insgesamt ergibt sich

$$S \to aZ \mid a, \quad Z \to aX \mid aXZ, \quad C \to cS, \quad A \to aZR_a \mid aR_a$$

$$B \to aZR_b \mid aR_b, \quad X \to aZR_bC \mid aR_bC$$

$$R_a \to a, \quad R_b \to b, \quad R_c \to c$$

Aufgabe H20 (3+3+3+3 Punkte)

1. Die Sprache L_1 wird durch die folgende Grammatik erkannt:

$$S \to aSc \mid B \mid \epsilon \qquad B \to bBc \mid \epsilon$$

2. Es lässt sich über das Pumping-Lemma zeigen, dass L_2 nicht kontextfrei ist. Für beliebige n, sei das Wort $z = a^n b^n c^{n^2}$. Es liegt offensichtlich in L_2. Es gibt nun viele verschiedene Möglichkeiten, z in $uvwxy$ zu zerlegen. Da aber $|vwx| \leq n$ gelten muss, kann vwx nicht alle drei Terminale (a, b und c) gleichzeitig enthalten. Man sieht leicht, daß ein Aufpumpen zwangsweise dazu führt, dass das entstehende Wort nicht mehr in L_2 liegt.

3. Auch bei L_3 lässt sich anhand des Pumping-Lemmas zeigen, dass diese Sprache nicht kontextfrei ist. Sei n beliebig und $z = a^n b^n a^n b^n \in L_3$. Sei $z = uvwxy$ eine entsprechende Zerlegung. Wir unterscheiden zwischen zwei Fällen:

   1. Fall: vwx ist komplett in der ersten Hälfte oder komplett in der zweiten Hälfte von z enthalten. Durch Pumpen von v und x ändert sich daher eine Hälfte $a^n b^n$ und die andere Hälfte bleibt gleich. Das so entstandene Wort ist somit nicht in L_3 enthalten.

   2. Fall: vwx liegt im Übergang der beiden Wörter, also ist es ein Unterwort von $b^n a^n$. Das heißt, beim pumpen muss sich das erste b^n und das zweite a^n. Da $|vwx| \leq n$, ändert sich jedoch weder das erste a^n noch das zweite b^n. Somit ist das gepumpte Wort nicht mehr in L_3, da es die Form $a^n b^{n+i} a^{n+i} b^n$ mit $i \geq 1$ hat.

4. Bei L_4 können wir schon wieder das Pumping-Lemma benutzen, zum Beispiel für das Wort $z = a^n ab^{n-1} a^n \in L_4$ mit $p = p^R = a^n$ und $q = ab^{n-1}$. Für eine Zerlegung $z = uvwxy$ mit $|vwx| \leq n$ gilt $vwx \in L(a^* b^* + b^* a^*)$. Somit können wir beim Pumpen zwischen drei Fällen unterscheiden:

   1. Fall: Die Länge von q ändert sich, p und p^R bleiben gleich.

   2. Fall: Die Länge von p ändert sich, p^R bleibt gleich.

   3. Fall: Die Länge von p^R ändert sich, p bleibt gleich.

   In jedem der drei Fälle wird das Wort so verändert, dass es nicht mehr in der Sprache ist. Folglich ist L_4 nicht kontextfrei.

Aufgabe H21 (Bonuspunkte)

Ein Beweis, daß L nicht kontextfrei ist, findet sich in R. J. Moll und S. M. Venkatesan, „Fibonacci Numbers Are Not Context-Free", *The Fibonacci Quarterly* 29.1 (1991):59-61. Ihr Beweis verwendet Ogdens Lemma, welches eine Verallgemeinerung des beliebten Pumping-Lemmas ist. Allerdings läßt sich der Beweis auch leicht mit dem Pumping-Lemma selbst führen, wenn man zum Beispiel L^R statt L betrachtet.

Für die Sprache L' taugt diese Beweismethode allerdings nicht und wir müssen uns etwas anderes überlegen. Zunächst zeigen wir, dass L' nicht regulär ist. Anschließend folgern wir daraus, dass L' auch nicht kontextfrei ist. Sei M eine Sprache. Definieren wir uns

$$d(M) = \lim_{n\to\infty} \frac{\bigl|\, \{\, w \in M \mid |w| \le n \,\} \,\bigr|}{n}.$$

Jetzt ist $d(M)$ der Grenzwert für die relative Anzahl an Wörtern aus dem Alphabet, die in M sind.

Nehmen wir jetzt einmal an L' wäre regulär. Wir werden einen Widerspruch schaffen, indem wir zum einen zeigen, dass $d(L')$ eine rationale Zahl ist, aber auch zeigen, dass $d(L')$ keine rationale Zahl ist.

Zunächst zeigen wir, dass $d(L')$ eine rationale Zahl ist. Wir definieren uns $L'_1 = L' \cap 10\$^*$ und $L'_2 = L' \cap 11\$^*$. Dann ist natürlich L' die disjunkte Vereinigung dieser Sprachen. Desweiteren wären auch L'_1 und L'_2 regulär. Wie sähe ein minimaler DFA für L'_1 aus? Dieser müßte zunächst 10 lesen und dann nur Transitionen für \$ besitzen, alle anderen Zeichen führten sofort zum Fangzustand. Wie sieht aber ein DFA aus, der nur \$-Transitionen besitzt? Es gibt immer nur einen Folgezustand, so daß die Zustände einen Pfad bilden müssen, der sich irgendwann wieder selbst trifft. Mit anderen Worten: Ein Pfad, der in einen Kreis mündet. Dieser Kreis hat eine bestimmte Länge und eine bestimmte Anzahl von Endzuständen. Deren Verhältnis ist eine rationale Zahl r_1. Wieviele Wörter enthält nun L'_1 bis zur Länge n? Natürlich $r_1 n + O(1)$. Daher gilt $d(L'_1) = r_1$. Analog gilt $d(L'_2) = r_2$ für eine rationale Zahl r_2. Da L' die disjunkte Vereinigung von L'_1 und L'_2 ist, gilt $d(L') = d(L'_1) + d(L'_2)$. Somit ist $d(L')$ ebenfalls eine rationale Zahl.

Nun zeigen wir, dass $d(L')$ keine rationale Zahl ist.

Es gilt, dass F_m in Binärdarstellung die Länge $\log(F_m) + O(1)$ hat und bis auf endlich viele Ausnahmen die Anzahl der Wörter gleicher Länge in L und L' gleich ist. Daher gilt auch

$$d(L') = \lim_{n\to\infty} \frac{\bigl|\, \{\, F_m \mid \log(F_m) \le n \,\} \,\bigr|}{n}.$$

Jetzt ist aber $F_m = \phi^m/\sqrt{5} + O(1)$ und daher gibt es $m + O(1)$ Fibonaccizahlen bis zur Länge $m \log(\phi)$, woraus $d(L') = 1/\log(\phi)$ folgt. Wir müssen jetzt nur noch zeigen, daß $\log(\phi)$ irrational ist und wir haben einen Widerspruch.

Es ist $\phi = (\sqrt{5}+1)/2$ und wenn $\log\phi + 1 = a/b$ mit $a, b \in \mathbf{N}$, dann wäre

$$2^a = (\sqrt{5}+1)^b = \sum_{k=0}^{b} \binom{b}{k} \sqrt{5}^b = s\sqrt{5} + t$$

für Zahlen $s, t \in \mathbf{N}$, $s, t > 0$, was natürlich unmöglich ist. Daher ist L' nicht regulär. Es bleibt noch zu zeigen, daß L' nicht kontextfrei ist.

Wir können $L' = 10U \cup 11U'$ schreiben mit $U, U' \subseteq \$^*$. Man kann sich leicht überlegen, daß genau dann L' kontextfrei ist, wenn U und U' kontextfrei sind. Als unäre Sprachen sind aber U und U' genau dann kontextfrei, wenn sie regulär sind, was genau dann der Fall ist, wenn auch L' regulär ist. (Unäre Sprachen haben ein einelementiges Alphabet.)

Die einzige Lücke in diesem Beweis besteht darin, daß in der Vorlesung nie bewiesen wurde, daß eine unäre Sprache genau dann kontextfrei ist, wenn sie regulär ist. Wir müssten dies hier also extra noch einmal beweisen. Eigentlich folgt dies aus dem Satz von Parikh, welcher aber ebenfalls in der Vorlesung nicht vorkam. Die Originalarbeit ist R. Parikh (1966), On Context-Free Languages, *Journal of the Association for Computing Machinery*, 13(4). Dort oder in Lehrbüchern über Formale Sprachen läßt sich der Beweis finden, welcher der letzte Schritt zur Lösung dieser Bonusbonusaufgabe ist.