Informatik ᗡ: – Blatt 7

Rasmus Diederichsen

19. Juli 2014

Aufgabe 7.1

a)

Es ist $L = \{a^k b^l c^m \mid k, l, m \in \mathbb{N}\} \cup \{d^j a^k b^k c^k \mid j, k \in \mathbb{N}, j > 0\} = L_1 \cup L_2$. Offensichtlich ist L_2 nicht kontextfrei, da es sich in einer kontextfreien Grammatik nicht sicherstellen lässt, dass mehr als zwei verschieden Symbole gleich häufig vorkommen müssen. Dies wäre unproblematisch, falls $L_2 \subseteq L_1$, denn eine reguläre Sprache (wie L_1) kann ja ohne Weiteres eine Sprache höherer Klasse als Teilmenge beinhalten. Dies ist hier nicht der Fall, $L_1 \cap L_2 = \emptyset$. Um L darzustellen, ist also eine kontextfreie Grammatik nicht ausreichend.

b)

Es wird hier angenommen, dass es sich in der Sprachspezifikation um ein nicht-exklusives 'Oder' handelt.

Fall 1: Für $j = 0, k = l = m$ wähle v, x entweder so, dass sie jeweils nur ein Zeichen (beliebig oft wiederholt) enthalten oder leer sind (nicht beide zugleich). Man kann dann beliebig aufpumpen. Zwar verliert man die Struktur $a^k b^k c^k$, jedoch ist $a^l b^m c^n$ ja immernoch $\in L$.

Fall 2: Falls $j = 0, k, l, m, \geq 0$, ist die Sprache offensichtlich kontextfrei (sie ist sogar regulär), da es keine weiteren Beschränkungen für k, l, m gibt. Das Pumping Lemma hilft hier also schonmal nicht weiter.

Fall 3: Falls $j \neq 0, k, l, m \geq 0$, ist die Sprache ebenfalls regulär und damit kontextfrei.

Fall 4: Falls $j \neq 0, k = l = m$ haben wir ein Wort $z = d^j a^k b^k c^k$ mit $j + 3k \geq n$. In diesem Fall wähle

- $u = \varepsilon$
- $v = d$
- $w = \varepsilon$

- $x = \varepsilon$
- $y = a^k b^k c^k$

Die Bedingungen $|vx| \geq 1$ und $|vwx| \leq n$ sind erfüllt. Offensichtlich ist $uv^i wx^i y = \varepsilon d^i \varepsilon \varepsilon^i a^k b^k c^k \in L$. Diese Zerlegung lässt sich immer finden, unabhäbgig von n. Das Pumping Lemma ist demnach hier unnütz.

Aufgabe 7.2

Die Sprache L ist nicht kontextfrei, was mir mittels des Pumping Lemmas zeigen. Sei n die Wortmindestlänge und $z_n = a^n b^{2^n} \in L$. Es gilt

$$|z_n| = n + 2^n \text{ und } |z_{n+1}| = n + 1 + 2^{n+1}$$

Es gibt folgende Möglichkeiten, eine Zerlegung $uvwxy$ von z_n zu finden:

Fall 1: $vwx = a^k, k \geq 1$ enthält nur as. Hierbei könten durch Pumpen beliebig viele as erzeugt werden, daher z.B. $uv^2wx^2y \notin L$.

Fall 2: vwx enthält as und bs.

(a) Falls $w = \varepsilon$, so ist $vwx = a^l b^k, n \geq l + k \geq 1$. Es gilt für $uv^2wx^2y = a^{n-l} a^{2l} b^{2k} b^{2^n - k} = a^{n+l} b^{2^n + k}$ dass $|uv^2wx^2y| = n + l + 2^n + k$. Dieses Wort muss mindestens die Länge von z_{n+1} haben, um in L enthalten sein zu können. Die Gleichung

$$\begin{aligned} n + l + 2^n + k &= n + 1 + 2^{n+1} \\ l + k &= 1 + 2^{n+1} - 2^n \\ &= 1 + 2^n(2-1) \\ &= 1 + 2^n \end{aligned}$$

ist jedoch nicht erfüllbar, da $l + k \leq n$ nach Vorraussetzung.

(b) Falls $w \neq \varepsilon$, ändert dies nichts, da $l + k$ dadurch lediglich kleiner würde und die obige Gleichung weiterhin unerfüllbar ist.

Fall 3: vwx enthält nur bs. Mit der Begründung von Punkt 1 gilt $uv^iwx^iy \notin L$

Aufgabe 7.3

Ausgehend von

$$\begin{array}{lcl} S & \rightarrow & E \mid ABC \\ A & \rightarrow & bdC \mid E \mid S \\ B & \rightarrow & acDC \mid C \\ C & \rightarrow & ab \mid Ja \mid IdA \\ D & \rightarrow & deF \mid DeF \mid dEF \mid DEF \\ E & \rightarrow & abba \mid aBBa \mid AbbA \\ F & \rightarrow & JcJ \mid GJ \mid bG \\ G & \rightarrow & F \mid IG \\ H & \rightarrow & SA \mid Hcc \\ I & \rightarrow & c \mid cIc \mid dF \end{array}$$

Markieren wir

Schritt 1

$$
\begin{array}{lcl}
S & \to & E \mid ABC \\
A & \to & bdC \mid E \mid S \\
B & \to & acDC \mid C \\
C & \to & ab \mid Ja \mid IdA \\
D & \to & deF \mid DeF \mid dEF \mid DEF \\
E & \to & abba \mid aBBa \mid AbbA \\
F & \to & JcJ \mid GJ \mid bG \\
G & \to & F \mid IG \\
H & \to & SA \mid Hcc \\
I & \to & c \mid cIc \mid dF
\end{array}
$$

Schritt 2

$$
\begin{array}{lcl}
S & \to & E \mid ABC \\
A & \to & bdC \mid E \mid S \\
B & \to & acDC \mid C \\
C & \to & ab \mid Ja \mid IdA \\
D & \to & deF \mid DeF \mid dEF \mid DEF \\
E & \to & abba \mid aBBa \mid AbbA \\
F & \to & JcJ \mid GJ \mid bG \\
G & \to & F \mid IG \\
H & \to & SA \mid Hcc \\
I & \to & c \mid cIc \mid dF
\end{array}
$$

Schritt 3

$$
\begin{array}{lcl}
S & \to & E \mid ABC \\
A & \to & bdC \mid E \mid S \\
B & \to & acDC \mid C \\
C & \to & ab \mid Ja \mid IdA \\
D & \to & deF \mid DeF \mid dEF \mid DEF \\
E & \to & abba \mid aBBa \mid AbbA \\
F & \to & JcJ \mid GJ \mid bG \\
G & \to & F \mid IG \\
H & \to & SA \mid Hcc \\
I & \to & c \mid cIc \mid dF
\end{array}
$$

Die reduzierte Grammatik ist also

Schritt 4

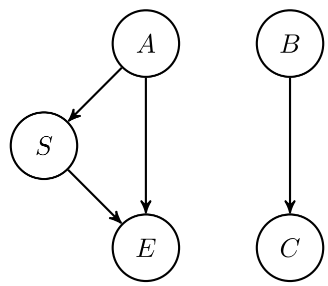
$$
\begin{array}{lcl}
S & \to & E \mid ABC \\
A & \to & bdC \mid E \mid S \\
B & \to & acDC \mid C \\
C & \to & ab \mid Ja \mid IdA \\
D & \to & deF \mid DeF \mid dEF \mid DEF \\
E & \to & abba \mid aBBa \mid AbbA \\
F & \to & JcJ \mid GJ \mid bG \\
G & \to & F \mid IG \\
H & \to & SA \mid Hcc \\
I & \to & c \mid cIc \mid dF
\end{array}
$$

Wir sehen also, dass die Sprache nicht leer ist. Markiert man bis zum Ende durch, erhält man

$$
\begin{array}{lcl}
S & \to & E \mid ABC \\
A & \to & bdC \mid E \mid S \\
B & \to & acDC \mid C \\
C & \to & ab \mid Ja \mid IdA \\
D & \to & deF \mid DeF \mid dEF \mid DEF \\
E & \to & abba \mid aBBa \mid AbbA \\
F & \to & JcJ \mid GJ \mid bG \\
G & \to & F \mid IG \\
H & \to & SA \mid Hcc \\
I & \to & c \mid cIc \mid dF
\end{array}
$$

$$
\begin{aligned}
S &\rightarrow E \mid ABC \\
A &\rightarrow E \mid S \\
B &\rightarrow C \\
C &\rightarrow ab \mid IdA \\
E &\rightarrow abba \mid aBBa \mid AbbA \\
I &\rightarrow c \mid cIc
\end{aligned}
$$

Die Regeln der Form $V \rightarrow V$ sind $\{S \rightarrow E, A \rightarrow E, A \rightarrow S, B \rightarrow C\}$. Hieraus entsteht der Graph



Der Graph ist bereits kreisfrei. Es müssen also noch die Senken eliminiert werden.

Elimination von Senke C

Man führt die neuen Regeln

$$B \rightarrow ab \mid IdA$$

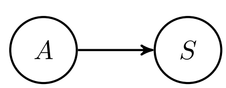
ein. Entfernt wird die Regel $B \rightarrow C$.

Elimination von Senke E

Man führt die neuen Regeln

$$
\begin{aligned}
S &\rightarrow abba \mid aBBa \mid AbbA \\
A &\rightarrow abba \mid aBBa \mid AbbA
\end{aligned}
$$

ein. Entfernt werden die Regeln $A \rightarrow E$ und $S \rightarrow E$. Der reduzierte Graph ist



Eliminiere Senke S

Man führt die Regeln

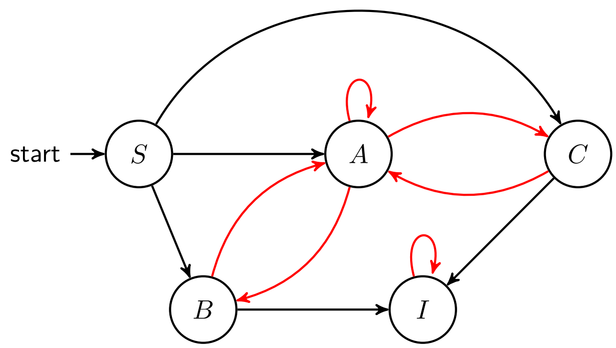
$$A \rightarrow abba \mid aBBa \mid AbbA \mid ABC$$

ein, von denen Teile in A bereits enthalten sind. Wir entfernen $A \to S$. Die finale Grammatik ist

$$\begin{array}{ccl} S & \to & abba \mid aBBa \mid AbbA \mid ABC \\ A & \to & abba \mid aBBa \mid AbbA \mid ABC \\ B & \to & ab \mid IdA \\ C & \to & ab \mid IdA \\ E & \to & abba \mid aBBa \mid AbbA \\ I & \to & c \mid cIc \end{array}$$

wobei die Variable E nicht mehr erreicht wird aufgrund der Subsitutionen mit ihrer rechten Seite in den anderen Regeln.

Der Graph bestehend aus den Variablen ist



Dieser Graph hat Kreise, also ist die Sprache nicht endlich.

Aufgabe 7.4

Das “Problem” mit der zweiten Definition ist, dass A nur eine Variable sein kann, der Rest der Satzform muss auf u und w aufgeteilt werden, die auf der rechten Seite unverändert bestehen bleiben müssen. Egal ob man A mit A oder B belegt, die jeweils andere Variable wird nach anwenden der Produktion rechts oder links stehen bleiben.

Um diese Art von Regeln darstellen zu können, führen wir ein:

$$\begin{array}{ccc} X & \to & AB \\ Y & \to & BA \end{array}$$

ein, die beide der zweiten Definition Genüge tun. Zusätzlich definiert man die Regel

$$X \to Y$$

sowie für jede Regel

$$uVv \to u \dots AB \dots v$$

eine Regel

$$uVv \to u \dots X \dots v$$

Alternativ könnte man Regeln der folgenden Form einführen (C sei eine neue Variable):

$$\begin{array}{rcl} AB & \to & AC \\ AC & \to & BC \\ BC & \to & BA \end{array}$$

Aufgabe 7.5

		b	c	a	c	c	a	
			c				c	
	c	b	c	a	a	c	b	c
			c		a	c	a	b
b	c	b	c	b	c	b	c	a

Aufgabe 7.6