Grundzüge der Theoretischen Informatik, WS 21/22: Musterlösung zum 13. Übungsblatt

Julian Dörfler

Dieses Blatt hat 26 reguläre Punkte, jedoch sind hiervon wie gewohnt nur 16 Punkte für die Zulassung relevant, der Rest sind Bonuspunkte.

Aufgabe A13.1 (Quiz) *(4 Punkte)*
Welche der folgenden Aussagen sind richtig, welche falsch. Begründen Sie Ihre Antworten jeweils kurz.

(a) Jede NP-schwere Sprache ist entscheidbar.

(b) Wenn $A \leq_{\mathrm{P}} B$ und A in NP ist, dann ist auch B in NP.

(c) Wenn $A \leq_{\mathrm{P}} B$ und B in NP ist, dann ist auch A in NP.

(d) $\mathsf{NSpace}(O(n^2)) \subsetneq \mathsf{DSpace}(O(n^{42}))$

Lösung A13.1 (Quiz)

(a) Falsch. Das Halteproblem ist nicht entscheidbar, jedoch ist dieses NP-schwer: Es existiert aber eine Polynomialzeit-Reduktion von SAT auf das Halteproblem: Bilde die SAT-Instanz φ auf das folgende Programm ab:

Probiere alle Belegungen der Variablen in φ aus. Falls davon eine erfüllend ist, terminiere. Ansonsten divergiere.

(b) Falsch. Wir hatten schon gezeigt, dass unentscheidbare NP-schwere Sprachen existieren. Sei B solch eine Sprache und $A \in \mathsf{NP}$ beliebig. Dann gilt $A \leq_{\mathrm{P}} B$, aber $B \notin \mathsf{NP}$, da alle Sprachen in NP entscheidbar sind.

(c) Wahr. Sei f die Reduktionsfunktion $A \leq_{\mathrm{P}} B$, berechnet durch das Programm P. Da $B \in \mathsf{NP}$ ist, existiert eine NTM N, die B in nicht-deterministischer Polynomialzeit entscheidet. Wir können nun A in nicht-deterministischer Polynomialzeit entscheiden, indem wir zuerst P simulieren und dann B ausführen. Es gilt also $A \in \mathsf{NP}$.

(d) Wahr. Aus dem Satz von Savitch folgt $\mathsf{NSpace}(n^2) \subseteq \mathsf{DSpace}(O(n^4))$. Der Platzhierachiesatz gibt uns nun $\mathsf{DSpace}(O(n^4)) \subsetneq \mathsf{DSpace}(O(n^{42}))$ wodurch die Aussage folgt.

Aufgabe A13.2 (Kontextfreie Grammatiken) *(4 Punkte)*
Wann immer Sie im folgenden eine kontextfreie Grammatik angeben sollen, so dürfen zusätzlich zur Definition aus dem Skript beliebige Produktionen der Form $V \to \varepsilon$ verwendet werden.

(a) Geben Sie eine kontextfreie Grammatik für die Sprache

$$L_1 = \{x \in \{0,1\}^\star \mid x \text{ enthält den Substring } 01\}$$

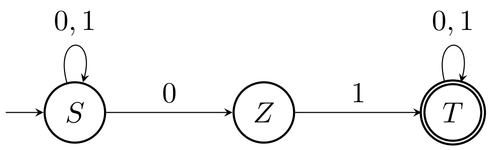
an und begründen Sie die Korrektheit Ihrer Konstruktion kurz. *Hinweis:* Ihre Grammatik sollte nicht wesentlich mehr als 3 Variablen verwenden.

(b) Geben Sie eine kontextfreie Grammatik für die Sprache aller korrekten Klammerausdrücke[1] an und begründen Sie die Korrektheit Ihrer Konstruktion kurz. *Hinweis:* Ihre Grammatik sollte nicht wesentlich mehr als eine Variable verwenden.

(c) Welche der beiden Sprachen lassen sich auch durch rechtslineare Grammatiken erkennen? Beweisen Sie Ihre Antworten.

Lösung A13.2 (Kontextfreie Grammatiken)

(a) Die Sprache L_1 wird erkannt von folgendem NEA:



Daraus konstruieren wir die Grammatik $G_1 = (\{S, Z, T\}, \{0, 1\}, P_1, S)$, die L_1 akzeptiert, wobei P_1 gegeben ist durch:

$$\begin{aligned} S &\to 0S \mid 1S \mid 0Z \\ Z &\to 1T \\ T &\to \varepsilon \mid 0T \mid 1T \end{aligned}$$

(b) Eine alternative Beschreibung für korrekte Klammerausdrücke ist die folgende:

(i) ε ist offensichtlich ein korrekter Klammerausdruck.

(ii) Wenn K ein korrekter Klammerausdruck ist, so ist (K) ebenfalls ein korrekter Klammerausdruck.

(iii) Wenn K_1 und K_2 korrekte Klammerausdrücke sind, so ist K_1K_2 ebenfalls ein korrekter Klammerausdruck.

(iv) Alles andere sind keine korrekten Klammerausdrücke.

Dies erlaubt uns direkt die Konstruktion einer Grammatik $G_K = (\{S\}, \{(,)\}, P_K, S)$, die korrekte Klammerausdrücke erkennt, wobei P_k wie folgt gegeben ist:

$$S \to \varepsilon \mid (S) \mid SS$$

[1]Siehe Aufgabe A9.2 des 9. Übungsblattes für eine Definition

(c) Wir wissen, dass eine rechtslineare Grammatik für eine Sprache genau dann existiert, wenn diese Sprache regulär ist.

Die Sprache L_1 ist regulär, wie durch obigen NEA gezeigt, somit existiert eine rechtslineare Grammatik, die L_1 erkennt. Die kontextfreie Grammatik die wir oben für L_1 konstruiert hatten ist sogar schon fast rechtslinear, es müssen nur noch die ε-Produktionen entfernt werden.

Die Sprache aller Palindrome hingegen erlaubt keine rechtslinearen Grammatiken, da sie wie in Aufgabe A9.2 bewiesen nicht regulär ist.

Aufgabe A13.3 (Korrektheit von Grammatiken) *(4 Punkte)*
Zeigen Sie, dass die Grammatik $G = (\{S\}, \{0,1\}, P, S)$ die Sprache aller Palindrome erkennt, wobei P gegeben ist durch $S \to \varepsilon \mid 0 \mid 1 \mid 0S0 \mid 1S1$. Gehen Sie hierzu wie folgt vor:

(a) Zeigen Sie, dass alle durch diese Grammatik produzierten Wörter, die noch Variablen enthalten, genau die Wörter der Form xSx^{rev} sind für beliebige Strings $x \in \{0,1\}^\star$.

(b) Folgern Sie daraus, dass G die Sprache L_2 aller Palindrome erkennt, gegeben durch

$$L_2 = \{x \in \{0,1\}^\star \mid x = x^{\text{rev}}\}\,.$$

Lösung A13.3 (Korrektheit von Grammatiken)

(a) Wir zeigen die Aussage per natürlicher Induktion über $n = |x|$ bzw. die Länge der Ableitung von xSx^{rev}.

$n = 0$ Das einzige Wort der Form xSx^{rev} ist hier S, welches ohne Anwendung von Produktionen aus S abgeleitet werden kann.

$n \to n+1$ Aus xSx^{rev} können wir durch Anwendung der Produktionen nur zwei Wörter produzieren, die noch Variablen enthalten. $x0S0x^{\text{rev}}$ und $x1S1x^{\text{rev}}$. Im Fall $x0S0x^{\text{rev}}$ gilt $(x0)^{\text{rev}} = 0x^{\text{rev}}$ und somit hat das produzierte Wort die korrekte Form. Im Fall $x1S1x^{\text{rev}}$ gilt $(x1)^{\text{rev}} = 1x^{\text{rev}}$ und somit hat das produzierte Wort ebenfalls die korrekte Form. Weiterhin ist $\{y0 \mid y \in \{0,1\}^{|x|}\} \cup \{y1 \mid y \in \{0,1\}^{|x|}\} = \{y \in \{0,1\}^{|x|+1}\}$, somit sind also auch alle Worte der Form $x'Sx'^{\text{rev}}$ produzierbar für $|x'| = |x| + 1$.

(b) Wir betrachten zuerst, welche Worte wir durch Anwendung einer einzelnen Regel aus xSx^{rev} erhalten können, die nur noch Terminale enthalten.

xx^{rev}*:* Dies ist offensichtlich ein Palindrom, da $(xx^{\text{rev}})^{\text{rev}} = (x^{\text{rev}})^{\text{rev}}x^{\text{rev}} = xx^{\text{rev}}$.

$x0x^{\text{rev}}$*:* Dies ist ebenfalls ein Palindrom, da $(x0x^{\text{rev}})^{\text{rev}} = (x^{\text{rev}})^{\text{rev}}0^{\text{rev}}x^{\text{rev}} = x0x^{\text{rev}}$.

$x1x^{\text{rev}}$*:* Dies ist ebenfalls ein Palindrom, direkt analog zu $x0x^{\text{rev}}$.

Weiterhin lässt sich jedes Palindrom x von G produzieren. Dabei unterscheiden wir zwischen 2 Fällen:

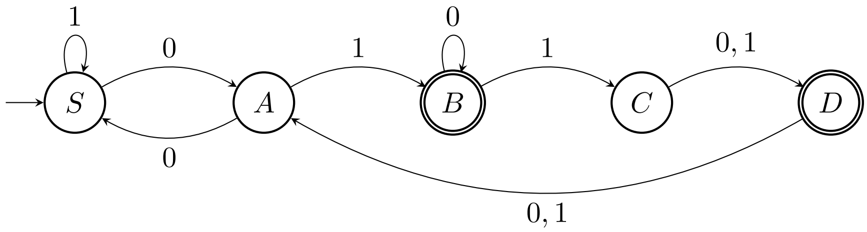
$|x|$ *ist gerade:* Dann lässt sich x zerlegen in $x = x_1x_2$ mit $|x_1| = |x_2|$. Nun gilt aber $x_2 = x_1^{\text{rev}}$, da $x = x^{\text{rev}}$. Nun lässt sich $x_1Sx_1^{\text{rev}}$ nach der vorherigen Teilaufgabe von G produzieren, also lässt sich $x = x_1x_1^{\text{rev}}$ ebenfalls produzieren.

$|x|$ *ist ungerade:* Dann lässt sich x zerlegen in $x = x_10x_2$ (analog für $x = x_11x_2$) mit $|x_1| = |x_2|$. Nun gilt aber $x_2 = x_1^{\text{rev}}$, da $x = x^{\text{rev}}$. Nun lässt sich $x_1Sx_1^{\text{rev}}$ nach der vorherigen Teilaufgabe von G produzieren, also lässt sich $x = x_10x_1^{\text{rev}}$ ebenfalls produzieren.

Aufgabe A13.4 (Rechtslineare Grammatiken) *(6 Punkte)*

(a) Wir vervollständigen den Beweis aus der Vorlesung: Zeigen Sie, dass wenn eine Sprache L regulär ist, diese auch von einer rechtslinearen Grammatik erkannt wird.

(b) Wenden Sie diesen Beweis nun praktisch an, indem Sie zu folgendem DEA eine äquivalente rechtslineare Grammatik angeben:



Lösung A13.4 (Rechtslineare Grammatiken)

(a) Sei L eine reguläre Sprache. Dann gibt es einen DEA $M = (Q, \Sigma, \delta, q_0, Q_{\text{acc}})$, der L erkennt. Falls in M eingehende Transitionen zu q_0 existieren und $\varepsilon \in L(M)$, dann kopieren wir q_0 und alle ausgehenden Kanten von q_0 und machen diesen neuen Zustand den Startzustand. Danach machen wir mit diesem neuen DEA weiter.

Wir konstruieren die rechtslineare Grammatik $G = (Q, \Sigma, P, q_0)$, wobei P die folgenden Produktionen enthält:

$$q \to \sigma q' \text{ für alle } q, q' \in Q \text{ und } \sigma \in \Sigma \text{ mit } \delta(q, \sigma) = q'$$
$$q \to \sigma \text{ für alle } q \in Q,\ q' \in Q_{\text{acc}} \text{ und } \sigma \in \Sigma \text{ mit } \delta(q, \sigma) = q'$$

Falls $q_0 \in Q_{\text{acc}}$, fügen wir die Produktion $q_0 \to \varepsilon$ hinzu.

Wir zeigen nun $L(G) = L(M)$. Es gilt für jedes Wort $w = w_1w_2 \dots w_n \in \Sigma^\star$:

$$\begin{aligned}
w \in L(G) \Leftrightarrow &\text{es gibt Variablen } q_1, \dots, q_{n-1} \in Q \text{ mit} \\
&\quad q_0 \Rightarrow_G w_1q_1 \Rightarrow_G w_1w_2q_2 \Rightarrow_G \dots \Rightarrow_G w_1w_2 \dots w_{n-1}q_{n-1} \Rightarrow_G w_1w_2 \dots w_{n-1}w_n \\
\Leftrightarrow &\text{es gibt Zustände } q_1, \dots, q_{n-1} \in Q \text{ und } q_n \in Q_{\text{acc}} \text{ mit} \\
&\quad \delta(q_0, w_1) = q_1, \delta(q_1, w_2) = q_2, \dots, \delta(q_{n-2}, w_{n-1}) = q_{n-1}, \delta(q_{n-1}, w_n) = q_n \\
\Leftrightarrow &w \in L(M)
\end{aligned}$$

(b) Wir wenden das beschriebene Verfahren an und erhalten die Grammatik $G = (\{S, A, B, C, D\}, \{0, 1\}, P, S)$ wobei P gegeben ist durch die folgenden Produktionen:

$$\begin{aligned}
S &\to 1S \mid 0A \\
A &\to 0S \mid 1B \mid 1 \\
B &\to 0B \mid 0 \mid 1C \\
C &\to 0D \mid 0 \mid 1D \mid 1 \\
D &\to 0A \mid 1A
\end{aligned}$$

Aufgabe A13.5 (Lineare Grammatiken) *(4 Punkte)*
Zur Erinnerung: Eine Grammatik (V, Σ, P, S) ist rechtslinear, wenn sie Typ-2 ist und die rechte Seite aller Produktionen die Form $\Sigma V \cup \Sigma$ hat, mit Ausnahme einer möglichen $S \to \varepsilon$ Produktion.
Weiterhin ist eine Grammatik linkslinear, wenn sie Typ-2 ist und die rechte Seite aller Produktionen die Form $V\Sigma \cup \Sigma$ hat, mit Ausnahme einer möglichen $S \to \varepsilon$ Produktion.
Beweisen Sie, dass links-und rechtslineare Grammatiken die selben Sprachen erkennen.

Lösung A13.5 (Lineare Grammatiken) Wir hatten schon gezeigt, dass die rechtslinearen Grammatiken genau die regulären Sprachen beschreiben.

Es reicht also zu zeigen, dass die linkslinearen Grammatiken ebenfalls genau die regulären Sprachen beschreiben.

Sei $G = (V, \Sigma, P, S)$ eine linkslineare Grammatik, die eine Sprache L erkennt. Nun ergibt sich durch Reversieren der rechten Seiten einer jeden Produktion eine rechtslineare Grammatik G', die eine reguläre Sprache L' erkennt. Wir betrachten nun eine Ableitung eines Wortes $x \in L$ in G. Durch Reversieren aller Zwischenworte in der Ableitung sehen wir, dass $S \Rightarrow^*_{G'} x^{\text{rev}}$. Umgekehrt impliziert $S \Rightarrow^*_{G'} y$ ebenfalls mit Reversieren aller Wörter, dass $S \Rightarrow^*_G y^{\text{rev}}$. Es gilt also $L' = L^{\text{rev}}$, und daher ist L' ebenfalls regulär.

Falls L auf der anderen Seite eine reguläre Sprache ist, so können wir eine rechtslineare Grammatik G_{rev} angeben, die L^{rev} erkennt. Mit dem selben Argument können wir diese Grammatik nun zu einer linkslinearer Grammatik reversieren, die L erkennt.

Aufgabe A13.6 (Chomsky-Normalform) *(4 Punkte)*
Konstruieren Sie zur Grammatik $G = (\{S, A, B, C, D\}, \{a, b\}, P, S)$ eine äquivalente

Grammatik in Chomsky-Normalform, wobei P gegeben ist durch:

$$\begin{aligned}
S &\to ABCD \\
A &\to B \mid a \\
B &\to \varepsilon \mid Ab \mid AB \\
C &\to bb \\
D &\to C
\end{aligned}$$

Gehen Sie hierzu wie folgt vor:

(a) Entfernen Sie alle ε-Produktionen.

(b) Entfernen Sie alle Kettenproduktionen.

(c) Konstruieren Sie die Grammatik in Chomsky-Normalform.

Geben Sie nach jedem dieser Schritte Ihre Grammatik an.

Lösung A13.6 (Chomsky-Normalform)
Wir geben nach jedem Schritt die Produktionen an (siehe Syntactic Sugar 28.11 im Skript):

(a) Die Variablen A und B sind nullierbar.

$$\begin{aligned}
S &\to ABCD \mid ACD \mid BCD \mid CD \\
A &\to B \mid a \\
B &\to Ab \mid b \mid AB \mid A \mid B \\
C &\to bb \\
D &\to C
\end{aligned}$$

(b) Wir entfernen zuerst zyklische Kettenproduktionen (A und B sind äquivalent):

$$\begin{aligned}
S &\to AACD \mid ACD \mid CD \\
A &\to a \mid Ab \mid b \mid AA \\
C &\to bb \\
D &\to C
\end{aligned}$$

Nun ersetzen wir die restlichen Kettenproduktionen (in diesem Fall nur $D \to C$):

$$\begin{aligned}
S &\to AACD \mid ACD \mid CD \\
A &\to a \mid Ab \mid b \mid AA \\
C &\to bb \\
D &\to bb
\end{aligned}$$

(c) Wir fügen zuerst die Hilfsvariable T_b ein:

$$\begin{aligned} S &\to AACD \mid ACD \mid CD \\ A &\to a \mid AT_b \mid b \mid AA \\ C &\to T_bT_b \\ D &\to T_bT_b \\ T_b &\to b \end{aligned}$$

Bleibt also nur noch übrig die Produktionen von S aus mit mehr als 2 Variablen auf der rechten Seite zu ersetzen:

$$\begin{aligned} S &\to AS_{ACD} \mid AS_{CD} \mid CD \\ S_{ACD} &\to AS_{CD} \\ S_{CD} &\to CD \\ A &\to a \mid AT_b \mid b \mid AA \\ C &\to T_bT_b \\ D &\to T_bT_b \\ T_b &\to b \end{aligned}$$

Aufgabe A13.7 (Unäre Sprachen ($\star$)) *(1+3 Bonuspunkte)*
Eine *unäre* Sprache L ist eine Teilmenge von $\{1\}^*$. Zeigen Sie:

(a) Es gibt unäre Sprachen die nicht entscheidbar sind.

(b) Wenn L unär und NP-schwer ist, dann gilt $\mathsf{P} = \mathsf{NP}$.
Hinweis: Zeigen und nutzen Sie die Existenz einer polynomiell zeitbeschränkten Reduktionsfunktion f von SAT nach L, so dass $f(F) = f(F') \Leftrightarrow |f(F)| = |f(F')|$ und $f(F) = f(F') \Rightarrow (F \in \mathsf{SAT} \Leftrightarrow F' \in \mathsf{SAT})$.

Lösung A13.7 (Unäre Sprachen ($\star$))

(a) Die Sprache $H' := \{1^i \mid i \in H_0\}$ ist nicht entscheidbar, da $H_0 \leq H'$.

(b) Falls $L = \{1\}^*$ ist L offensichtlich nicht NP-schwer. Sei f die Reduktionsfunktion von SAT nach L, wobei $L \neq \{1\}^*$ unär ist. Wir modifizieren f wie folgt: Wenn $f(x) \notin \{1\}^*$, dann setzen wir stattdessen $f(x) := 1^u$, so dass $1^u \notin L$ und beobachten, dass f dann weiterhin eine Reduktionsfunktion von SAT nach L ist. Nun definieren wir eine Äquivalenzrelation auf SAT:

$$F \sim F' :\Leftrightarrow f(F) = f(F') \left(\Leftrightarrow |f(F)| = |f(F')| \text{ da } \mathsf{im} f \subseteq \{1\}^*\right) \qquad (\star)$$

Da f eine polynomialzeit Reduktion ist, gibt es Konstanten c und d, so dass $\forall F: |f(F)| \leq c|F|^d$. Ferner gilt: Wenn $F \sim F'$, dann sind F und F' entweder beide erfüllbar oder beide unerfüllbar. Wir geben einen Algorithmus für SAT an:
Gegeben F

(1) Initialisiere ein Array A der Größe $c|F|^d$.

(2) Wenn F nur eine Variable hat, dann teste ob F erfüllbar ist. Wenn ja, dann setze $A[|f(F)|] = 1$, ansonsten setze $A[|f(F)|] = 0$.

(3) Ansonsten wähle eine Variable x_i aus und berechne F' und F'', so dass x_i in F' auf wahr gesetzt wird und in F'' auf falsch gesetzt wird. Hierzu wird die Formel danach vereinfacht, also falls eine Klausel schon erfüllt ist, wird diese Klausel entfernt und falls eine Klausel leer werden würde, wird die gesamte Formel direkt unerfüllbar.

(4) Falls $A[|f(F')|]$ oder $A[|f(F'')]$ undefiniert sind, dann fahre rekursiv für die undefinierten Einträge fort.

(5) Setze $A[|f(F)|] = A[|f(F')|] \vee A[|f(F'')|]$.

Zuletzt geben wir $A[|f(F)|]$ aus.
Die Korrektheit des Algorithmus folgt aus $(\star)$ - es genügt mit den Äquivalenzklassen von $F \sim F' \Leftrightarrow |f(F)| = |f(F')|$ zu rechnen. Ferner ist der Algorithmus polynomiell zeitbeschränkt, da A nur polynomielle Größe hat, und wir jeden Eintrag von A nur einmal berechnen.
Es folgt, dass $\mathsf{SAT} \in \mathsf{P}$, also $\mathsf{P} = \mathsf{NP}$.