

Theoretische Informatik A

Lösungsvorschläge zur Klausur vom 24. September 2016

Aufgabe 1 (10 Punkte)

An den Korrektor: Das Wichtigste ist, dass die Idee der Reduktion richtig ist. Also dass die Reduktionsfunktion f in polynomieller Zeit berechenbar ist und $x \in \text{HC} \Leftrightarrow f(x) \in \text{DOUBLEHC}$. Ob die Kodierung richtig angegeben wurde, ist eher zweitrangig.

Es sei $x \in \{0, 1\}^*$, und sei $n := \lfloor \sqrt{\lg(x)} \rfloor$. Wir schreiben

$$x = \bar{x}_1 \cdots \bar{x}_n \bar{x}_r$$

wobei $\lg(\bar{x}_i) = n$ für $i = 1, \dots, n$ gelte. Der Term $(\bar{x}_r$ bezeichnet also den für die Repräsentation des Graphen irrelevanten Rest von x .) Sei $\bar{0}$ eine Abkürzung für das Wort der Länge n aus $\{0\}^*$. Dann definieren wir $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ durch

$$f(x) := \bar{x}_1 \bar{0} \cdots \bar{x}_n \bar{0} \bar{x}_1 \cdots \bar{0} \bar{x}_n$$

für alle $x \in \{0, 1\}^*$. Offenbar notiert $f(x)$ einen gerichteten Graphen mit $2n$ Knoten, sofern x einen gerichteten Graphen mit n Knoten notiert. Ferner bilden dann die neu hinzugekommenen Knoten $\{n+1, \dots, 2n\}$ eine Kopie des ursprünglichen Graphen, und weitere Kanten treten nicht auf. Deshalb gilt für alle $x \in \Sigma^*$:

$$x \in \text{HC} \Leftrightarrow f(x) \in \text{DOUBLE-HC}.$$

Da $\lfloor \sqrt{\lg(x)} \rfloor$ in Linearzeit berechnet werden kann und $\lg(f(x)) \leq 4 \cdot \lg(x)$ ist, gilt $f \in \text{FZEIT}(n)$. Es folgt $\text{HC} \leq_{\text{pol}} \text{DOUBLE-HC}$. Wegen der Transitivität von \leq_{pol} ist damit die NP-Schwere von DOUBLE-HC nachgewiesen.

Aufgabe 2 (10 Punkte)

An den Korrektor: Hier ist ebenfalls die Kodierung und auch die Korrektheit der einzelnen Schritte der Turingmaschine zweitrangig. Wichtig ist, dass man

sieht dass das Konzept zu zeigen, dass ein Problem in NP ist, verstanden wurde. Also korrekte Angabe des Zeugen und polynomieller Algorithmus zum verifizieren.

Das Wichtigste ist, die richtige Hilfeingabe (Zertifikat/Zeuge) zu benennen. Hier ist das die zwei Knoten Mengen V_1 und V_2 und Permutationen darauf, die die zwei Kreise beschreiben. Die Kontrollturingmaschine M überprüft zuerst ob die Eingabe und die Hilfeingabe sinnvoll sind. Dann überprüft sie, ob gilt dass $V = V_1 \cup V_2$ und ob V_1 und V_2 disjunkt voneinander sind. Dann muss noch überprüft werden ob die Permutationen gültige Kreise darstellen (ob die Kanten vorhanden sind): Falls alle Test mit ja ausfallen, akzeptiert M .

Hier nochmal formaler: Zunächst erläutern wir, wie M die Eingabe auf Band 0 verstehen soll: Wir haben zwei Blöcke durch „1“ getrennt. Jeder dieser Blöcke soll einen der disjunkten Kreise darstellen. Innerhalb so eines Blocks sollen durch „1“ getrennte Blöcke von Nullen durch ihre Länge die Nummern der Knoten verschlüsseln, die in der gegebenen Reihenfolge möglicherweise einen Kreis darstellen. – M arbeite dann die folgende Befehlsfolge ab:

1. Berechne mit Hilfe der Arbeitsbänder 2 und 3 sowie des Eingabebandes die ganzzahlige Wurzel der Länge der Eingabe: $n := \lfloor \sqrt{\lg(x)} \rfloor$.
 { Z. B. unter Verwendung der Identität $\sum_{i=1}^k (2i - 1) = k^2$. }
2. Kopiere den Inhalt von Band 0 auf Band 5 und Band 6 bis einschließlich des n -ten Blocks von Nullen, falls *nicht*
 - Band 0 mit „1“ beginnt,
 - dort häufiger als einmal zwei Einsen aufeinander folgen,
 - nie zwei Einsen aufeinanderfolgen,
 - vorher das Bandende von Band 0 erreicht ist *oder*
 - mindestens ein solcher Block aus mehr als n Nullen besteht;

andernfalls gehe zu HALT.

3. Teste, ob auf Band 5 für jedes $i \in \{1, \dots, n\}$ genau ein Block der Länge i vorkommt;

{ Dies kann man mit Hilfe eines Zählers auf Band 4 durchführen, indem man der Reihe nach für jedes i nach Blöcken der Länge i sucht. }
bei negativem Testausgang gehe zu HALT.

4. Bestimme die Stelle auf Band 5 an der „11“ vorkommt. Wir nehmen an, dass vor „11“ k -viele „0“-er Blöcke waren.
5. FOR $i = 1, \dots, k - 1$ DO
 - BEGIN
 - { Sei $\lg(b_i)$ die Länge des von links i -ten Blocks $b_i \in \{0\}^*$ auf Band 6. }
 - Berechne $m_i := (\lg(b_i) - 1) \cdot n + \lg(b_{i+1})$;
 - Teste, ob das m_i -te Symbol von x gleich „1“ ist; falls nicht, gehe zu HALT
 - END
6. Berechne $m_k := (\lg(b_k) - 1) \cdot n + \lg(b_1)$;
- Teste, ob das m_k -te Symbol von x gleich „1“ ist; falls nicht, gehe zu HALT
7. FOR $i = k + 1, \dots, n - 1$ DO
 - BEGIN
 - { Sei $\lg(b_i)$ die Länge des von links i -ten Blocks $b_i \in \{0\}^*$ auf Band 6. }
 - Berechne $m_i := (\lg(b_i) - 1) \cdot n + \lg(b_{i+1})$;
 - Teste, ob das m_i -te Symbol von x gleich „1“ ist; falls nicht, gehe zu HALT
 - END
8. Berechne $m_n := (\lg(b_n) - 1) \cdot n + \lg(b_{k+1})$;
- Teste, ob das m_n -te Symbol von x gleich „1“ ist; falls nicht, gehe zu HALT
9. HALT+.

Eine genaue Inspektion dieser Beschreibung ergibt, dass tatsächlich $L_M = \text{DOUBLEHC}$ gilt. Darüber hinaus arbeitet M in polynomialer Zeit, wie man leicht sieht.

Aufgabe 3 (8 Punkte, je 2 Punkte pro Teilaufgabe)

1. Diese Aussage ist richtig, denn nach Lemma 4.3.3 ist die Klasse NP abgeschlossen unter \leq_{pol} .
2. Diese Aussage ist falsch. Als Gegenbeispiel wähle man $A := \Sigma^*$ und eine Menge B , die nicht aus NP ist. Wegen $B \neq \emptyset$ gilt $\Sigma^* \leq_{pol} B$; als Reduktionsfunktion wähle man eine konstante Funktion mit Wert aus B .

Die Existenz einer Menge $B \notin \text{NP}$ folgt auch aus

$$\text{NP} \subseteq \bigcup_{c \in \mathbb{N}} \text{ZEIT}(2^{n^c}) \subseteq \text{ZEIT}(2^{2^n}) \subsetneq \text{ZEIT}(n \cdot 2^{2^n})$$

Weitere Beispiele für Wortmengen, die nicht aus NP sind, sind die nicht entscheidbaren Wortmengen.

3. Diese Aussage ist falsch. Als Gegenbeispiel wählen wir $A := \emptyset \in \text{P} \subseteq \text{NP}$ und die NP-vollständige Sprache $B := \text{HC}$ (Hamiltonkreisproblem). Wegen $B \neq \emptyset$ gilt $A \leq_{pol} B$, als Reduktionsfunktion wähle man eine konstante Funktion mit Wert aus $\Sigma^* \setminus B$. Aber $A = \emptyset$ ist nicht NP-schwer.
4. Diese Aussage ist richtig. Für alle Sprachen $L \in \text{NP}$ gilt $L \leq_{pol} A$, weil A NP-hart ist. \leq_{pol} ist eine transitive Relation, insbesondere impliziert $L \leq_{pol} A \leq_{pol} B$ damit $L \leq_{pol} B$. Somit ist B auch NP-schwer.

Aufgabe 4 (10 Punkte)

Die Sprache L ist regulär, deshalb gibt es einen endlichen (determinierten) Automaten $A = (\Sigma, Q, q_0, F, \delta)$ der die Sprache L erkennt. Wir wollen nun A so umbauen, dass genau die Wörter erkannt werden die ein Präfix eines Wortes aus L darstellen. Sei nun $F' = \{w \in Q \mid \text{es gibt } w \in \Sigma^* \text{ mit } \delta^*(q_0, w) \in F\}$ die Menge aller Zustände, die auf einem Weg von A zu einem akzeptierenden Zustand in F liegen. Der Rest des Automaten verhält sich genau gleich wie A . Es gilt also, dass $A' = (\Sigma, Q, q_0, F', \delta)$ Die Sprache $\text{PRAEFIX}(L)$ erkennt.

Aufgabe 5 (14 Punkte)

(a) (6 Punkte)

Die Grammatik kann durch folgende Regelmenge angegeben werden:

$$\begin{aligned} S &\rightarrow Tbb \\ T &\rightarrow aTb \mid \varepsilon. \end{aligned}$$

(b) (8 Punkte)

Wir benutzen das Pumping Lemma um zu zeigen, dass L nicht regulär ist. Wir nehmen an, L sei regulär. Dann gibt es eine Zahl n mit den im Pumping-Lemma genannten Eigenschaften. Es gilt $a^n b^{n+2} \in L$. Wir wählen $t := \varepsilon$, $z := a^n$, $\bar{t} := b^{n+2}$. Dann kann z in u, v, w zerlegt werden mit $z = uvw$, $v \neq \varepsilon$ und $\forall i \geq 0. tuv^i w \bar{t} \in L$ (insbesondere $tuv^0 w \bar{t} = tw \bar{t} \in L$). Es gibt k mit $v = a^k$ ($0 < k \leq n$) und es gilt $t w \bar{t} = a^{n-k} b^{n+2} \notin L$ (da die Anzahl der a s und die Anzahl der b s sich um mehr als 2 unterscheidet), also Widerspruch. Damit ist L nicht regulär.

Aufgabe 6 (6 Punkte)

(a) (3 Punkte)

Abbildung 1 zeigt eine Ableitungsbaum für das angegebene Wort.

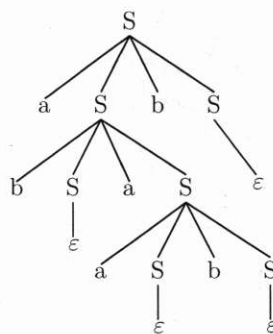


Abbildung 1: Ein Ableitungsbaum für das Wort abaabb

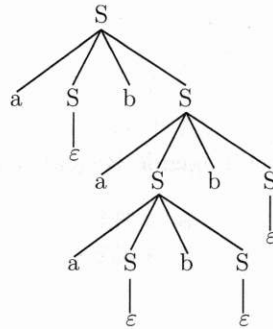


Abbildung 2: Ein weiterer Ableitungsbaum für das Wort abaabb

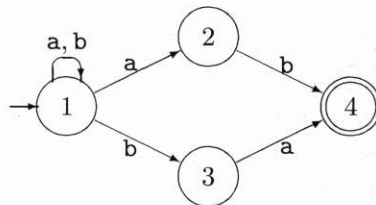
(b) (3 Punkte)

Die Grammatik ist nicht eindeutig, Abbildung 2 zeigt einen weiteren Ableitungsbaum für das angegebene Wort.

Aufgabe 7 (12 Punkte)

1. (6 Punkte)

Der nichtdeterminierte Automat $A = (\Sigma, \{1, 2, 3, 4\}, 1, \{4\}, \delta)$ mit dem folgenden Übergangsgraphen erkennt L :



2. (6 Punkte)

Das im Kurs angegebene Verfahren liefert:

P	a	b	$W \setminus V$
{1}	{1, 2}	{1, 3}	{1, 2}, {1, 3}
{1, 2}	{1, 2}	{1, 3, 4}	{1, 3}, {1, 3, 4}
{1, 3}	{1, 2, 4}	{1, 3}	{1, 3, 4}, {1, 2, 4}
{1, 3, 4}	{1, 2, 4}	{1, 3}	{1, 2, 4}
{1, 2, 4}	{1, 2}	{1, 3, 4}	-

Somit ist $\{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 2, 4\}, \{1, 3, 4\}\}$ die Zustandsmenge des vereinfachten Potenzautomaten B zu A , der Anfangszustand ist $\{1\}$ und die Endzustandsmenge ist $\{\{1, 2, 4\}, \{1, 3, 4\}\}$. Graphisch läßt sich B wie folgt darstellen:

