

Übung zur Vorlesung BERECHENBARKEIT UND KOMPLEXITÄT

Lösung Blatt 13

Tutoriumsaufgabe 13.1

- (a) Welche Teilmengen-Beziehungen sind über die Komplexitätsklassen P , NP , $coNP$, $PSPACE$ und $EXPTIME$ bekannt?

$P \subseteq NP \subseteq PSPACE \subseteq EXPTIME$ und

$P \subseteq coNP \subseteq PSPACE$.

Alles bis auf $coNP \subseteq PSPACE$ wurde in der Vorlesung behandelt.

In der Vorlesung haben wir gesehen, dass $NP \subseteq PSPACE$ (ein $PSPACE$ -Algorithmus kann alle Zertifikate mit durch ein Polynom beschränkter Länge durchprobieren).

Es folgt, dass $coNP \subseteq coPSPACE$, dann für $A \in coNP$ folgt, dass $\bar{A} \in NP \subseteq PSPACE$, und somit $A \in coPSPACE$.

Da wir das Akzeptanzverhalten einer TM invertieren können, folgt $coPSPACE = PSPACE$, und damit $coNP \subseteq PSPACE$.

(Alternativ kann man fest stellen, dass man $coNP$ -Zertifikate analog wie NP -Zertifikate durchprobieren kann.)

- (b) Welche der obigen Komplexitätsklassen sind eine Teilmenge der LOOP-berechenbaren Probleme?

Alle, denn jedes Problem aus $EXPTIME$ ist Loop-Berechenbar.

Sei $A \in EXPTIME$. Dann gibt es ein Polynom q und eine TM M , die $PSPACE$ in $2^{q(n)}$ vielen Schritten. In der Vorlesung haben wir gesehen, wie WHILE-Programme eine TM simulieren können. Die einzige verwendete While-Schleife, die nicht direkt als Loop-Schleife, dargestellt werden kann, ist die, welche einen Schritt der TM simuliert solange der Endzustand nicht erreicht wird. Es reicht den Inhalt dieser While-Schleife maximal $2^{q(n)}$ auszuführen.

Wir skizzieren, dass $2^{q(n)}$ Loop-berechenbar ist. Wir können all Potenzen, die im fixen Polynom $q(n)$ vorkommen sind Loop-berechenbar. Da zudem alle Konstanten vom fixen Polynom $q(n)$, die Addition und Multiplikation Loop-berechnen. Daher ist $q(n)$ Loop-berechenbar. Folgendes Loop-Programm berechnet 2^{x_1} gegeben x_1 .

$x_2 := x_2 + 1;$

$x_2 := x_2 + 1;$

```

 $x_3 := x_1;$ 
 $x_1 := 1;$ 
LOOP  $x_3$  DO
     $x_1 := x_1 \cdot x_2$ 
ENDLOOP

```

Wir können daher A mit folgendem LOOP-Programm berechnen. Berechne $2^{q(n)}$. Dann simuliere die TM M analog zur Simulation mit einem While-Programm. Ersetze dabei die äußere Schleife durch eine LOOP-Schleife, die ihren Inhalt $2^{q(n)}$ mal ausführt. (Ist in einem Simulationsschritt schon der Endzustand erreicht, so mache nichts im Schleifen-Körper.)

Tutoriumsaufgabe 13.2

- (a) Für ein Entscheidungsproblem A gebe es einen Algorithmus mit Laufzeitschranke $\mathcal{O}(n \cdot \log(n))$. Zudem gebe es eine polynomielle Reduktion $B \leq_p A$, die Laufzeit $\mathcal{O}(m^5)$ benötigt. Dabei bezeichne n und m jeweils die Eingabelängen der Probleme A bzw. B . Welche Laufzeit kann man daraus für einen Algorithmus für B folgern.

Wir können B wie folgt entscheiden. Reduziere die Eingabe b der Länge m auf eine Eingabe a für das Problem A . Da es eine Reduktion mit Laufzeit $\mathcal{O}(m^5)$ gibt, können wir die Länge von a abschätzen durch $\mathcal{O}(m^5)$. Dann entscheiden wir a mit dem Algorithmus für A mit Laufzeit $\mathcal{O}(n \cdot \log n)$ mit $n = \mathcal{O}(m^5)$ Eingabelänge von Algorithmus a . Also können wir B entscheiden in Laufzeit $\mathcal{O}(m^5 \cdot \log(m^5)) = \mathcal{O}(m^5 \cdot \log m)$.

- (b) Sei A ein beliebiges Entscheidungsproblem in P .

Geben Sie eine polynomielle Reduktion $A \leq_p \text{CLIQUE}$.

Sei \mathcal{A} ein Algorithmus der A in polynomieller Zeit löst.

Konstruktion: Entscheide die Instanz von A mit Algorithmus \mathcal{A} . Falls \mathcal{A} akzeptiert, konstruiere eine Clique-Instanz mit dem Graphen ohne Knoten und Zielgröße $k = 0$. Falls \mathcal{A} verwirft, konstruiere eine Clique-Instanz mit dem Graphen ohne Knoten und Zielgröße $k = 1$.

Da \mathcal{A} polynomielle Laufzeit, hat auch die Konstruktion polynomielle Laufzeit.

Tutoriumsaufgabe 13.3

Sei \mathbb{M} die Menge der Gödelnummern. Welche der folgenden Sprachen sind entscheidbar? Welche der folgenden Sprachen sind rekursiv aufzählbar?

- (a) \mathbb{M}

Es gibt eine TM, die entscheidet ob die Eingabe eine Gödelnummer ist (so wie wir es bei vielen Syntax-Überprüfungen in alten Aufgaben verwendet haben). Also ist \mathbb{M} entscheidbar und rekursiv aufzählbar.

- (b) $L_b = \{\langle M \rangle \mid M \text{ entscheidet } \mathbb{M}\}$.

L_b ist unentscheidbar nach Satz von Rice. Sei $\mathcal{S} := \{f_M \mid f_M(w) = 1, \text{ falls } w \in \mathbb{M}, f_M(w) = 0, \text{ falls } w \notin \mathbb{M}\} \subseteq \mathcal{R}$. Dann ist

$$\begin{aligned} L(\mathcal{S}) &= \{\langle M \rangle \mid M \text{ berechnet eine Funktion aus } \mathcal{S}\} \\ &= \{\langle M \rangle \mid L(M) = \mathbb{M}\} = L_b. \end{aligned}$$

Es gilt $\mathcal{S} \neq \emptyset$, denn für die TM M aus a), die \mathbb{M} entscheidet, gilt $f_M \in \mathcal{S}$.

Es gilt $\mathcal{S} \neq \mathcal{R}$, denn für die TM M , die sofort verwirft, gilt $f_M \notin \mathcal{S}$.

Nach Satz von Rice folgt, dass $L(\mathcal{S}) = L_b$ unentscheidbar ist.

L_b ist nicht rekursiv aufzählbar, da H_{all} nicht rekursiv aufzählbar und wir im folgendem zeigen, dass $H_{all} \leq L_b$.

Gegeben M , sei M^* eine TM, die wie folgt arbeitet. TM M^* kopiert Eingabe w auf 2-tes Band. Dann simuliere M auf Eingabe w (auf Band 1). Dann simuliere M_a (eine TM, die a) entscheidet) auf Eingabe w (auf Band 2), und übernehme die Ausgabe.

Konstruktion f ; Falls Eingabe w keine Gödelnummer ist, gib $\langle M' \rangle$ aus wobei M' eine TM ist, die immer verwirft. Sonst, falls w die Form $\langle M \rangle$ für eine TM M hat, konstruiere $\langle M^* \rangle$.

Die Konstruktion, insbesondere von $\langle M^* \rangle$ gegeben $\langle M \rangle$ ist berechenbar.

Korrektheit:

(\Rightarrow) Falls $w \in H_{all}$, dann hat w die Form $\langle M \rangle$ für eine TM M . Dann hält M auf jeder Eingabe. Dann terminiert die Simulation von M auf w . Dann verhält sich .

(\Leftarrow) Sei $w \notin H_{all}$. Falls w nicht die Form $\langle M \rangle$ hat, ist $f(w) = \langle M' \rangle \notin L_b$. Sonst hat w die Form $\langle M \rangle$ für eine TM M , und es gibt eine Eingabe x so dass M auf x nicht hält. Dann terminiert der erste Simulationsschritt von M^* nicht bei Eingabe x . Daher terminiert M^* bei Eingabe x nicht, und $f(w) = \langle M^* \rangle \notin L_b$.

(c) $L_c = \{\langle M \rangle \mid M \text{ entscheidet } L_b\}$.

Nach b) gilt, dass es keine TM M gibt mit $L(M) = L_b$. Daher ist $L_c = \emptyset$ und ist entscheidbar mit einer TM, die sofort verwirft. Daher ist L_c auch rekursiv aufzählbar.