Berechenbarkeit und Komplexität WS 00/01
bei Prof. Dr. W. Thomas

Lösung zur Scheinklausur

erstellt von Sandip Sar-Dessai

Aufgabe 1 (6 Punkte) Geben Sie detailliert und mit Erläuterung eine Turingmaschine an, welche die Funktion $f : \Sigma^\star_{bool} \to \Sigma^\star_{bool}$ mit $f(w) = w$ (falls w eine 1 enthält) und $f(w) = \varepsilon$ (sonst) berechnet.

Lösung Idee: M durchsucht das Eingabewort, falls sie eine 1 findet, geht sie an den Anfang und stoppt, ansonsten bleibt sie auf dem abschließenden Blank stehen (Ausgabe ε):

$$\Gamma = \{0, 1, \sqcup\}, Q = \{q_0, q_1, q_S\}, M = (Q, \{0,1\}, \Gamma, q_0, q_s, \delta)$$

wobei δ entsprechend folgender Turingzeilen gegeben sei:

q_0	0	q_0	0	R	Noch keine 1 gefunden
q_0	1	q_1	1	L	1 gefunden: an den Anfang (mittels q_1)
q_0	$\sqcup$	q_s	$\sqcup$	N	Wort zuende, keine 1 gefunden: Ausgabe ε
q_1	0/1	q_1	0/1	L	Rücklauf
q_1	$\sqcup$	q_s	$\sqcup$	R	

Aufgabe 2 (6 Punkte) Zeigen Sie: Wenn eine Wortmenge $W \subseteq \Sigma^\star_{bool}$ aufzählbar ist, so ist sie der Definitionsbereich einer berechenbaren Funktion $f : \Sigma^\star_{bool} - \to \Sigma^\star_{bool}$. (Für den Algorithmus A, der f berechnet, muss also gelten: $w \in W$ gdw. A terminiert bei Eingabe w.)

Lösung Sei W aufzählbar, A der Algorithmus der W aufzählt. Gesucht ist eine Funktion $f : \Sigma^\star_{bool} - \to \Sigma^\star_{bool}$ mit $Def(f) = W$, also $f(x) = \perp$ für $x \notin W$. Der folgende Algorithmus berechnet ein f mit $Def(f) = W$:

Bei Eingabe w lasse zunächst A laufen und überprüfe die Ausgaben. Falls $w \in W$, wird A irgendwann w ausgeben, stoppe dann A und gebe einen Funktionswert aus, bspw. 1 (f berechnet dann $f : w \mapsto 1$ für $w \in Def(f)$). Falls $w \notin W$ terminiert der Algorithmus nicht, also $f(w) = \perp$.

Aufgabe 3 (6 Punkte) Seien $f_1, f_2 : N \to N$ LOOP-berechenbar. Zeigen Sie, dass auch $f : N \to N$ mit $f(n) = f_2(f_1(n))$ LOOP-berechenbar ist.

Lösung Seien f_1 und f_2 berechenbar durch die LOOP-Programme P_1, P_2. Dann läßt sich f durch folgendes LOOP-Programm berechnen (sei $m = maxind(P_1)$):

P_1;
`X2:=0;X3:=0;...;Xm:=0;` Alles bis auf Ausgabe löschen
P_2;

Auf die Eingabe n wird zunächst P_1 ausgeführt (mit Ausgabe $f_1(n)$), darauf dann P_2:

$$\begin{aligned}
[[P_2]](n, 0, ...) &= [[P_2]]([[X2 := 0; ...; Xm := 0]]([[P_1]](n, 0, ...)) \\
&= [[P_2]]([[X2 := 0; ...; Xm := 0]](f_1(n), k_1, ..., k_m, 0, ...) \\
&= [[P_2]](f_1(n), 0, ...) = (f_2(f_1(n)), l_1, ..., 0, ...)
\end{aligned}$$

Aufgabe 4 (6 Punkte) Geben Sie eine geeignete Übersetzung an, um WHILE-Programme in REPEAT-Programme zu überführen: Definieren Sie induktiv über den Aufbau der WHILE-Programme P das jeweilige REPEAT-Programm P'. Führen Sie den Induktionsschritt nur für die Verkettung von Programmen und für das while-Statement aus. Erinnerung: Das repeat-Statement lautet `repeat P until Xi = 0` und das while-Statement lautet `while Xi > 0 do P end`.

Lösung Sei P ein WHILE-Programm, zu zeigen: dann ex. REPEAT-Programm P' mit $[[P']](k_1, ...) = [[P]](k_1, ...)$.

Induktionsanfang: Sei $P = Xi := 0$ oder $P = Xi := Xj$ oder $P = Xi := Xj + 1$ oder $P = Xi := Xj - 1$ oder $P = Xi := Xj\, op\, Xk$ (mit $op \in \{+, -, *, mod, div\}$. Dann ist $P' := P$ ein äquivalentes REPEAT-Programm mit äquivalenter semantischer Fkt.

Induktionsschluß:

- [...]
- Sei $P = P_1; P_2$. Nach IV gibt es zu P_1 und P_2 äquivalente REPEAT-Programme P_1' und P_2'. Dann ist $P' = P_1'; P_2'$ ein zu P äquivalentes REPEAT-Programm.
- Sei $P = while\, Xi > 0\, do\, P_1\, end$. Nach IV gibt es zu P_1 äquivalentes REPEAT-Programm P_1'. Dann ist $P' := if\, Xi > 0\, then\, repeat\, P_1'\, until\, Xi = 0\, end$ ein zu P äquivalentes REPEAT-Programm.

Aufgabe 5 (6 Punkte) Wir betrachten das "Wertproblem" $\underline{WeP}$ für Turingmaschinen:

Gegeben: Turingmaschine M über Σ_{bool} und $u, v \in \Sigma_{bool}^{\star}$
Frage: Liefert M für die Eingabe u die Ausgabe v?

Zeigen Sie: $\underline{WP} \leq \underline{WeP}$. Zur Erinnerung: Beim Wortproblem $\underline{WP}$ ist eine Turingmaschine M über Σ_{bool} und $w \in \Sigma_{bool}^{\star}$ gegeben. Die Frage ist, ob M auf der Eingabe w stoppt.

Lösung Zu zeigen: $\underline{WP} \leq \underline{WeP}$. Also: gesucht f total und berechenbar mit $f : E_{WP} \to E_{WeP}$ und $\forall x \in E_{WP} : x \in WP \Leftrightarrow f(x) \in WeP$.

Konstruktion von f: $f(M, w)$ sei $(M', w, 1)$, wobei M' zunächst arbeitet wie M, dann eine 1 auf das Band schreibt, ein Blank, dann einen Schritt zurückgeht und auf der 1 terminiert.

f ist berechenbar. Weiterhin:

- Sei $(M, w) \in WP$, also $M : w \to stop$. Dann stoppt auch M' auf w mit Ausgabe 1, $M' : w \to 1$, also $f(M) \in WeP$
- Sei $(M, w) \notin WP$, also $M : w \to \infty$. Dann stoppt M' nicht, macht also keine Ausgabe, somit $f(M) \notin WeP$

Aufgabe 6 (6 Punkte) Zeigen Sie dass das folgende Problem entscheidbar ist.

Gegeben: Turingmaschine M über Σ_{bool}, die auf einer Eingabe der Länge n höchstens n^2 viele Felder des Bandes besucht, und ein $w \in \Sigma_{bool}^{\star}$.

Frage: Stoppt M auf w?

Hinweis: Es sei $M = (Q, \Sigma, \Gamma, q_0, q_s, \delta)$. Bestimmen Sie zunächst die Höchstanzahl der möglichen Konfigurationen in Abhängigkeit von $|Q|$, $|\Gamma|$ und $n = |w|$.

Lösung Betrachte die Anzahl der Möglichkeiten für Konfigurationen von M:

- M kann in jedem Zustand ein Zeichen lesen, schreiben, den Zustand und die Position wechseln, insgesamt $|Q|^2 \cdot |\Gamma|^2 \cdot 3$ Kombinationen.
- Jede dieser Kombinationen bezieht sich auf eine der möglichen Bandinschriften der $|w|^2$ Felder, also auf eine von möglichen $|\Gamma|^{|w|^2}$ Inschriften.

Damit gibt es (abzüglich des einen Links- und Rechtsverbots)

$$|Q|^2 \cdot |\Gamma|^{2+|w|^2} \cdot 3 - 2$$

Zustände. Der Entscheidungsalgorithmus arbeite wie folgt: bei zunächst leerer Liste l starte er M auf w schrittweise. Nach jedem Schritt prüfe er, ob der aktuelle Zustand in l ist, falls ja, terminiere er mit 0 (M ist in einer Schleife, $M : w \to \infty$), ansonsten füge er den Zustand an l an. Terminiert M, terminiere er mit 1. Da es endlich viele Zustände gibt, muß irgendwann (sofern M nicht terminiert) ein Zustand erneut auftreten.

Aufgabe 7 (7 Punkte) Skizzieren Sie in ca. 10 Zeilen die Arbeitsweise einer Turingmaschine M, die die Funktion $f : \Sigma^\star_{bool} \to \Sigma^\star_{bool}$ mit $f(w) = w^R$ berechnet (dabei ist w^R das Wort w umgedreht). Begründen Sie anhand dieser Turingmaschine M, dass f in $TIME(n^2)$ ist.

Lösung Eine Tm. M, die bei Eingabe w das Wort w^R erzeugt arbeitet etwa wie folgt:

Für jedes Zeichen des Wortes (bis sie auf ein Blank trifft) führt sie durch:

1. ersetze a durch $\underline{a}$
2. das erste Blank vor dem Wort (befindet sich jetzt vor dem Quellwort)
3. das zweite Blank vor dem Wort (befindet sich vor dem geschriebenen Spiegelwort)
4. füge das Zeichen dort ein
5. laufe zu $\underline{a}$ zurück, ersetze es durch a und gehe auf nächstes Zeichen

Hinterher muß sie nur noch an den Anfang von w^R gehen.

Zur Komplexität: (1) in einem Schritt, (2) und (3) in max. $2n + 2$ Schritten, (4) wieder in einem Schritt und (5) in max. $2n+4$ Schritten. Insgesamt ist also (1)-(5) in $O(n)$. Dies wird für n Zeichen wiederholt, hinzu kommt der Rücklauf in $2n+2$ Schritten, also ergibt sich eine Gesamtkomplexität von $n \cdot O(n) + 2n + 2 = O(n^2)$, also ist f in $TIME(n^2)$.

Aufgabe 8 (7 Punkte) Das Problem Vertex Cover (VC) ist wie folgt definiert:

Gegeben: Ungerichteter Graph $G = (V, E)$, $k \geq 1$ (in Binärdarstellung)

Frage: Existiert $U \subseteq V$ mit k Knoten, so dass jede Kante von G wenigstens einen Endpunkt in U hat?

1. Skizzieren Sie in 5-10 Zeilen die Arbeitsweise einer nichtdeterministischen polynomzeitbeschränkten Turingmaschine, die VC entscheidet. (Auf dem Band sei eine Beschreibung von G in Form der Liste der Knotennamen und der Liste der Kanten vorgegeben.)
2. Man kann die NP-Vollständigkeit von VC zeigen, indem man $3SAT \leq_p VC$ nachweist. Formulieren Sie genau, was $3SAT \leq_p VC$ bedeutet (nur die Definition ist gefragt).

Lösung

1. Die NTM arbeitet wie folgt:

   (a) Sie zählt die Knoten; ist k größer als diese Zahl, terminiert sie mit 0.

   (b) Sie rät eine Menge von k Knoten aus der Menge der verfügbaren (Liste entsprechend verkleinern).

   (c) Sie geht zum Anfang der Kantenliste.

   (d) Sie prüft bei jeder Kante, ob entweder Anfangs- oder Endknoten noch in der Liste ist, falls ja weiter, ansonsten terminiere mit Ausgabe 0. Sind alle Kanten geprüft, terminiere mit 1.

   Zur Komplexität: (1) in $O(n)$, (2) in $O(1)$, (3) in$O(n)$, (4) in $O(n^2)$. Insgesamt polynomielle Laufzeit der NTM, also $\underline{VC} \in NP$.

2. $\underline{3SAT} \leq_p \underline{VC}$ bedeutet, daß $3SAT$ polynomiell reduzierbar auf $\underline{VC}$ ist, also das eine totale berechenbare Fkt. $f : E_{3SAT} \to E_{VC}$ existiert, so daß:

   (a) diese durch eine polynomzeitbeschränkte DTM berechnet werden kann

   (b) gewährleistet ist, daß für alle $x \in E_{3SAT}$ gilt: $x \in 3SAT \Leftrightarrow f(x) \in VC$
