Lehrstuhl für Informatik 1
Prof. Dr. Gerhard Woeginger
Jan Böker, Tim Hartmann

WS 2018/19

Übung zur Vorlesung
Berechenbarkeit und Komplexität

Lösung Blatt 3

Hausaufgabe 3.1 **(2 + 1 + 2 Punkte)**

In dieser Aufgabe wird folgendes RAM-Programm betrachtet:

```
 1: CLOAD 2
 2: STORE 2
 3: LOAD 1
 4: IF c(0)>0 THEN GOTO 6
 5: END
 6: CSUB 1
 7: STORE 1
 8: LOAD 2
 9: MULT 2
10: STORE 2
11: GOTO 3
```

(a) **Wenn die RAM als Eingabe im Register $c(1)$ eine Zahl $m \geq 0$ erhält, welcher Wert steht dann bei Termination im Register $c(2)$? Begründen Sie Ihre Antwort.**

Die RAM initialisiert Register $c(2)$ mit dem Wert 2. Dann durchläuft sie eine Schleife: Jeder Schleifendurchlauf dekrementiert zuerst $c(1)$ (durch die Befehle LOAD 1; CSUB 1; STORE 1) und quadriert dann $c(2)$ (durch die Befehle LOAD 2; MULT 2; STORE 2). Die Schleife bricht nach m Schritten ab, wenn $c(1)$ den Wert 0 erreicht. Der Effekt der Schleife ist also, dass die Zahl 2 im Register $c(2)$ genau m-mal quadriert wird. Am Ende steht daher die Zahl 2^{2^m} im Register $c(2)$.

(b) **Analysieren Sie die asymptotische (Worst-Case-)Laufzeit der RAM im uniformen Kostenmaß. Nehmen Sie dazu an, dass die Eingabe im Register $c(1)$ eine Binärzahl mit n Bits ist.**

Wenn die Eingabe m eine Binärzahl mit n Bits ist, gilt $2^{n-1} \leq m < 2^n$ und somit $m \in \Theta(2^n)$. Die Schleife wird daher $\Theta(2^n)$ mal durchlaufen. Da jeder Schleifendurchlauf aus konstant vielen Schritten besteht, ist die Gesamtlaufzeit $\Theta(2^n)$.

(c) **Analysieren Sie die asymptotische (Worst-Case-)Laufzeit der RAM im logarithmischen Kostenmaß. Nehmen Sie dazu an, dass die Eingabe im Register $c(1)$ eine Binärzahl mit n Bits ist.**

Wenn die Eingabe m eine Binärzahl mit n Bits ist, gilt $2^{n-1} \leq m < 2^n$, und im Worst-Case gilt $m = 2^n - 1$. Die Schleife wird dann $2^n - 1$ mal durchlaufen, und die Laufzeit beträgt (bei Vernachlässigung von Konstanten und insbesondere der durch die Werte in Register $c(1)$ verursachten Kosten)

$$\sum_{i=1}^{2^n-1} \log(2^{2^i}) = \sum_{i=1}^{2^n-1} 2^i = 2^{2^n} - 2,$$

wodurch man eine Worst-Case-Laufzeit von $\Theta(2^{2^n})$ erhält.

Hinweis: Im Best-Case $m = 2^{n-1}$ beträgt die Laufzeit $\Theta(2^{2^{n-1}})$, was nicht gleich $\Theta(2^{2^n})$ ist.

Hausaufgabe 3.2 **(5 Punkte)**

In dieser Aufgabe wird eine Erweiterung U' der universellen Turingmaschine U aus der Vorlesung „Turing-Maschinen II“ betrachtet, wozu eine sinnvolle Erweiterung der Gödelnummern auf k-Band-Turingmaschinen mit beliebigen Bandalphabeten angenommen wird. U' arbeitet wie folgt auf der Eingabe $\langle M \rangle w$, wobei M eine k-Band-TM und $w \in \{0,1\}^*$ ein Wort ist:

(1) Konstruiere aus M die Kodierung einer äquivalenten 1-Band-TM M', wie es in der Vorlesung skizziert wurde.

(2) Konstruiere aus M' die Kodierung einer äquivalenten 1-Band-TM M'', die nur das Bandalphabet $\{0,1,B\}$ verwendet, wie in Tutoriumsaufgabe 3.2.

(3) Arbeite wie U auf $\langle M'' \rangle w$.

Damit ist U' eine universelle 1-Band-TM, die eine beliebige k-Band-TM mit einem beliebigen Bandalphabet simulieren kann.

Analysieren Sie den Zeit- und Speicherverlust von U' bei der Simulation von M. Analysieren Sie dazu insbesondere den Speicherverlust von U. Die Kodierungslänge von M soll dabei als Konstante angesehen werden, und es darf angenommen werden, dass Zeit- und Speicherverlust der Konstruktion in Schritt 2 konstant sind.

Es bezeichne $t(n)$ die Laufzeit und $s(n)$ den Speicherbedarf von M.

(1) Der Konstruktionsvorgang von M' und M'' benötigt nur konstante Zeit und konstanten Speicherplatz, da die Kodierungslänge von M als Konstante angesehen wird.

(2) Nach Vorlesung hat M' Laufzeit $\mathcal{O}(t(n)^2)$ und Speicherbedarf $\mathcal{O}(s(n))$: M' hat für jedes Band von M zwei Spuren, eine für den Bandinhalt und eine für die Kopfposition. Für jeden Schritt von M muss M' über das Band laufen und die Kopfpositionen aktualisieren, wodurch sich der quadratische Zeitverlust ergibt.

(3) Nach dem ersten Punkt und dem Hinweis hat auch M'' Laufzeit $\mathcal{O}(t(n)^2)$ und Speicherbedarf $\mathcal{O}(s(n))$.

(4) Da U nur konstanten Zeitverlust hat, hat die Simulation Laufzeit $\mathcal{O}(t(n)^2)$. Somit muss nur noch der Speicherverlust von U analysiert werden:

- In der Initialisierung kopiert U die Gödelnummer auf Spur 2 und die Kodierung des ersten Zustandes auf Spur 3, was konstant viel Speicherplatz benötigt. Danach wird die Kodierung von Spur 1 entfernt.
- In einem Simulationsschritt wird die Zustandsübergangsfunktion mit Hilfe der Kodierung auf Spur 2 ausgewertet. Anschließend werden die Inschrift auf Spur 1 und der kodierte Zustand auf Spur 3 aktualisiert. Zudem werden die (konstant großen) Inhalte auf Spur 2 und Spur 3 mit dem Kopf bewegt. Auf Spur 2 und 3 wird also weiterhin konstant viel Speicherplatz benötigt, während auf Spur 1 genau der Speicherplatz von M'' benötigt wird.

Folglich hat U auch nur konstanten Speicherverlust.

Insgesamt hat U' Laufzeit $\mathcal{O}(t(n)^2)$ und Speicherbedarf $\mathcal{O}(s(n))$.

Anmerkung: Aufgrund des Hinweises wird ignoriert, dass die Alphabetreduktion einen Zeitverlust hat, der quadratisch in der Länge der Eingabe und der Ausgabe ist. (Dieser Verlust lässt sich mit einer leicht abweichenden Definition einer Turingmaschine vermeiden.) Wenn man dies nicht tut, erhält man die Laufzeit $\mathcal{O}(t(n)^4)$.

Hausaufgabe 3.3 **(5 Punkte)**

Zeigen Sie, dass die Menge $\mathbb{N}^* = \bigcup_{n\in\mathbb{N}} \mathbb{N}^n$ der endlichen Wörter über den natürlichen Zahlen abzählbar ist.

Hinweis: Die Menge $\mathbb{N}^*$ ist die Menge von endlichen Tupeln über den natürlichen Zahlen. Zum Beispiel sind $(1,1,1),(11,1),(1,11),(111)$ verschiedene Elemente von $\mathbb{N}^*$.

Lösung 1: Für $w = a_1 \dots a_n \in \mathbb{N}^*$ mit $a_i \in \mathbb{N}$ definiere $\mathrm{ind}(w) = \max\{a_1, \dots, a_n, n\} \in \mathbb{N}$. Dann ist die Menge $M_i = \{w \in \mathbb{N}^* \mid \mathrm{ind}(w) = i\}$ endlich für alle $i \in \mathbb{N}$. Weiter sei $k_i = \sum_{j<i} |M_i|$ und $M_i = \{w_{i,0}, \dots, w_{i,|M_i|-1}\}$ eine beliebige Durchnummerierung von M_i. Definiere $c : \mathbb{N}^* \to \mathbb{N}$ mit $c(w_{i,j}) = k_i + j$ für $i \geq 1$ und $c(\epsilon) = 0$. Dann ist c bijektiv und somit ist $\mathbb{N}^*$ abzählbar.

Lösung 2: Betrachte $\Sigma = \{0, 1, \#\}$ und $c : \Sigma^* \to \mathbb{N}^*$ mit

$$c(w) = \begin{cases} a_1 \dots a_n & \text{falls } w = \mathrm{bin}(a_1)\# \dots \# \mathrm{bin}(a_n) \\ 0 & \text{sonst} \end{cases} . \quad (1)$$

Dann ist c surjektiv. Da Σ^* abzählbar ist existiert eine Bijektion $c' : \mathbb{N} \to \Sigma^*$. Durch Verknüpfung von c' und c erhalten wir somit eine surjektive Abbildung von $\mathbb{N}$ nach $\mathbb{N}^*$. Also ist $\mathbb{N}^*$ abzählbar.