Theoretische Informatik
D. Flumini, L. Keller, O. Stern

Loesungen zum Übungsblatt 9

Abgabe: Kalenderwoche 50

Lösung 1.
Wäre das Komplement des allgemeinen Halteproblems semi-entscheidbar, dann wären sowohl H als auch $\overline{H}$ semi-entscheidbar. Wie in der Vorlesung gezeigt wurde, wäre dann H entscheidbar. Wir wissen aber, dass das allgemeine Halteproblem nicht entscheidbar ist. Folglich kann $\overline{H}$ nicht semi-entscheidbar sein.

Lösung 2.
Mögliche Begrundungen sind:

(a) Der Term $2n(5n + 3\log(n))$ lässt sich wie folgt umformen: $10n^2 + 6n\log(n)$
Zu prüfen sind nun folgende Teilprobleme:
$f_1 : 10n^2 \in \mathcal{O}(n^2)$
$f_2 : 6n\log(n) \in \mathcal{O}(n^2)$
f_1 stimmt, da $\underbrace{10 \cdot n^2}_{f_1(n)} \leq \underbrace{10}_{c} \cdot \underbrace{n^2}_{g(n)}$
f_2 stimmt ebenfalls, da $\underbrace{6n\log(n)}_{f_2(n)} \leq \underbrace{3}_{c} \cdot \underbrace{n^2}_{g(n)}$
Dadurch gibt es nun ein c, so dass $\underbrace{2n(5n + 3\log(n))}_{f(n)} \leq \underbrace{13}_{c} \cdot \underbrace{n^2}_{g(n)}$ gilt. Die Aussage ist somit wahr.

(b) Zu zeigen ist, dass ein c existiert, so dass $13 \cdot 2^{n+5} \leq c2^n$ gilt.
$13 \cdot 2^{n+5} = 13 \cdot 2^5 \cdot 2^n = 416 \cdot 2^n$
Dadurch gilt: $13 \cdot 2^{n+5} \leq 416 \cdot 2^n$

(c) Würde $n^2 \cdot 3^n \in \mathcal{O}(n^5)$ gelten, so gäbe es ein c so dass $n^2 \cdot 3^n \leq cn^5$ gilt.
$$\begin{aligned} & n^2 \cdot 3^n \leq cn^5 \\ \iff\ & 3^n \leq c(\tfrac{n^5}{n^2}) \\ \iff\ & 3^n \leq cn^3 \\ \iff\ & e^{n\cdot\log(3)} \leq ce^{3\cdot\log(n)} \end{aligned}$$
Durch die Exponentialdarstellung sehen wir nun, dass auf der linken Seite $n\cdot\log(3)$ vorkommt und auf der rechten Seite $3 \cdot \log(n)$. Da n schneller wächst als $\log(n)$ ist die Aussage falsch. Es gibt also kein c, welches die Gleichung erfüllt.

Lösung 3.
Zwischen den Funktionen gelten folgende Beziehungen:

$$\mathcal{O}(f_1(n)) \subset \mathcal{O}(f_5(n)) \subset \mathcal{O}(f_2(n)) = \mathcal{O}(f_4(n)) \subset \mathcal{O}(f_3(n))$$

Lösung 4. (a) Dies ist der Fall, wenn zwei Funktionen asymptotisch gleich sind. $f(n) \in \theta(g(n))$

(b) Die Aussage gilt. Solche Funktionen könnten beispielsweise $f(n) = 5 \cdot n\log(n) + 33$ und $g(n) = n\log(n)$ sein. Beide sind verschieden, haben aber dieselbe Zeitkomplexität.

(c) Falsch. Die $\mathcal{O}$-Notation gibt nur eine obere Schranke an. Es kann sein, dass der logarithmische Algorithmus einen hohen konstanten Aufwand hat $Alg_1 = n\log(n) + 9999999$ und der quadratische nicht $Alg_2 = 2 \cdot n^2$. In diesem Fall könnten Kollektionen mit einem Umfang von bis zu 3166 Datensätzen mit dem quadratischen Verfahren schneller sortiert werden.

Lösung 5. (a) Der Zeuge ist ein Bit x_i für jede Zahl a_i mit

$$x_i = \begin{cases} 1 & \text{falls } a_i \in \text{S} \\ 0 & \text{falls } a_i \notin \text{S} \end{cases}$$

für alle $1 \leq i \leq n$, also ein Wort $x_1x_2x_3...x_n$ über dem Alphabet $\{0, 1\}$. Der Verifizierer addiert alle a_i mit $x_i = 1$ und vergleicht diese Summen mit b.

(b) Eine mögliche Lösung für diese Aufgabe:
Die Eingabe für den Verifizierer ist ein Wort

$$a_1 1 a_2 1 ... 1 a_n 1 b \# x_1 x_2 ... x_n$$

mit a_i und b unär codiert für alle $1 \leq i \leq n$ und $x_i \in \{0, 1\}$. Die Zahl $a_i = 5$ ist zum Beispiel als 00000 codiert.

Informell arbeitet der Verifizierer (hier eine 2-Band-Turingmaschine) wie folgt: Der Verifizierer kopiert den Zeugen auf das 2. Band und löscht ihn auf dem 1. Band. Danach setzt er beide Lese-/Schreibköpfe an den Anfang der Bänder. Nun werden alle a_i mit $x_i = 1$ auf dem ersten Band addiert (wie im Beispiel der Vorlesung über Turingmaschinen) und alle anderen a_i werden vom 1. Band gelöscht. Am Schluss löscht der Verifizierer das 2. Band und bleibt auf dem 1. Band auf dem 1. Symbol von b stehen.

Sobald das 2. Band leer ist, kopiert der Verifizierer b auf das zweite Band und löscht es auf dem 1. Band. Am Schluss dieses Kopiervorgangs stehen beide Lese-/Schreibköpfe auf dem am weitesten rechts liegenden Symbol auf dem jeweiligen Band. Also kann der Verifizierer durch nach links laufen auf beiden Bändern die zuvor berechnete Summe mit b vergleichen.

Die Zeitkomplexität dieses Verifizierers ist $\mathcal{O}(m)$, wenn die Eingabe (d.h. der Zeuge und die Zahlen $a_1, a_2, ..., a_n$ und b) insgesamt die Länge m hat.