Goethe-Universität Frankfurt am Main
Institut für Informatik
Theorie komplexer Systeme
Prof. Dr. Nicole Schweikardt

4. November 2009

Diskrete Modellierung

Wintersemester 2009/2010

Übungsblatt 3

Abgabe: *bis 11. November 2009, 8.15 Uhr (vor der Vorlesung oder in Raum RM 11-15/113)*

Aufgabe 1: **(20 Punkte)**

Beweisen Sie die folgenden Aussagen durch vollständige Induktion nach n.

(a) Für alle $n \in \mathbb{N}$ mit $n \geq 5$ gilt: $2^n > n(n+1)$.

(b) Für alle $n \in \mathbb{N}$ mit $n \geq 2$ gilt: $\prod_{i=1}^{n} i^{\,i} \leq n^{n(n-1)}$

Aufgabe 2: **(25 Punkte)**

Betrachten Sie die beiden folgenden Algorithmen zur Sortierung der Komponenten eines Tupels $t = (a_1, \ldots, a_{2^k})$ der Länge 2^k, d.h. mit 2^k Komponenten, wobei $k \in \mathbb{N}$.

Algo 1 (bei Eingabe eines Tupels $t = (a_1, \ldots, a_{2^k})$):

(1) Falls $k = 0$ ist, dann gib t als Ergebnis zurück.
(2) Sei t_1 die Ausgabe von *Algo 1* bei Eingabe des Tupels $(a_1, \ldots, a_{2^{k-1}})$.
(3) Sei t_2 die Ausgabe von *Algo 1* bei Eingabe des Tupels $(a_{2^{k-1}+1}, \ldots, a_{2^k})$.
(4) Gib $merge(t_1, t_2)$ zurück.

Hierbei ist $merge()$ eine Funktion, die bei Eingabe zweier sortierter Tupel t_1 und t_2 mit jeweils 2^{k-1} Komponenten ein sortiertes Tupel mit 2^k Komponenten zurückgibt. Man kann sich leicht überlegen, dass die Funktion $merge()$ so implementiert werden kann, dass sie nicht mehr als $3 \cdot 2^k$ Schritte benötigt. Deshalb braucht *Algo 1* insgesamt für ein Tupel der Länge 2^k nicht mehr als $f_1(k) = 2f_1(k-1) + 3 \cdot 2^k + 4$ Schritte, wobei $f_1(0) = 2$.

Algo 2 (bei Eingabe eines Tupels $t = (a_1, \ldots, a_{2^k})$):

(1) Wiederhole für jedes i von 1 bis $(2^k - 1)$:
(2) Wiederhole für jedes j von 1 bis $(2^k - i)$:
(3) Falls $a_j > a_{j+1}$, dann tausche die Komponenten a_j und a_{j+1} in t.
(4) Gib t zurück.

Insgesamt braucht *Algo 2* für ein Tupel der Länge 2^k höchstens $f_2(k) = 3 \cdot 2^{2k} + 2^k - 1$ Schritte.

(a) Welcher der beiden Algorithmen läuft im Allgemeinen schneller? D.h. welche der beiden Funktionen f_1 und f_2 liefert kleinere Funktionswerte?

(b) Beweisen Sie, dass Ihre Antwort aus (a) korrekt ist. D.h. falls Sie in (a) geantwortet haben, dass *Algo i* im Allgemeinen schneller als *Algo j* ist, dann finden Sie eine Zahl $n_0 \in \mathbb{N}$ und beweisen Sie per Induktion nach n, dass für alle $n \in \mathbb{N}$ mit $n \geq n_0$ gilt: $f_i(n) < f_j(n)$.

Aufgabe 3: **(30 Punkte)**

Die Menge UPNZ (Umgekehrte Polnische Notation auf Ziffern) sei die rekursiv wie folgt definierte Teilmenge von A^* für $A := \{0,1,2,3,4,5,6,7,8,9,+,*\}$:

Basisregel: (B) Jede Ziffer, also jedes Zeichen aus $\{0,1,2,3,4,5,6,7,8,9\}$ ist in UPNZ

Rekursive Regeln: (R1) Sind w_1 und w_2 in UPNZ, so ist auch $w_1 w_2 +$ in UPNZ
(R2) Sind w_1 und w_2 in UPNZ, so ist auch $w_1 w_2 *$ in UPNZ

Somit gilt beispielsweise $7 \in$ UPNZ und $12+ \in$ UPNZ, während $42 \notin$ UPNZ.

Sei z eine Ziffer, also ein Zeichen aus $\{0,1,2,3,4,5,6,7,8,9\}$ und seien w_1, w_2 Worte aus UPNZ. Die Funktionen $f : \text{UPNZ} \to \mathbb{N}$ und $g : \text{UPNZ} \to \{0,1\}$ sind rekursiv entsprechend der Definition von UPNZ wie folgt definiert:

$$f(z) = z \qquad\qquad g(z) = \begin{cases} 0, & \text{falls } z \text{ gerade ist} \\ 1, & \text{sonst} \end{cases}$$

$$f(w_1 w_2 +) = f(w_1) + f(w_2) \qquad\qquad g(w_1 w_2 +) = \begin{cases} 0, & \text{falls } g(w_1) = g(w_2) \\ 1, & \text{sonst} \end{cases}$$

$$f(w_1 w_2 *) = f(w_1) \cdot f(w_2) \qquad\qquad g(w_1 w_2 *) = \begin{cases} 0, & \text{falls } g(w_1) = 0 \text{ oder } g(w_2) = 0 \\ 1, & \text{sonst} \end{cases}$$

Nach diesen Definitionen ist zum Beispiel $f(32+47+*) = 55$ und $g(32+47+*) = 1$.

(a) Welche der folgenden Wörter w_1, w_2, w_3, w_4 gehören zur Menge UPNZ, welche nicht? Berechnen Sie $f(w_i)$ und $g(w_i)$ falls $w_i \in$ UPNZ für $i \in \{1,2,3,4\}$.

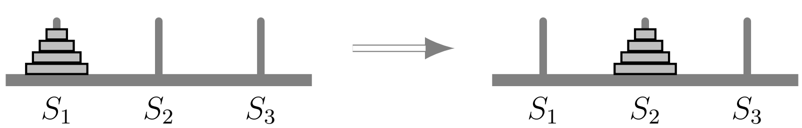
(i) $w_1 := 84+$ **(iii)** $w_3 := 8+2*1$

(ii) $w_2 := 32+12-$ **(iv)** $w_4 := 19+2*14++0+$

(b) Beweisen Sie den folgenden Zusammenhang durch vollständige Induktion:
Für jedes Wort $w \in$ UPNZ gilt: $f(w)$ ist gerade $\iff$ $g(w) = 0$.

Aufgabe 4: Türme von Hanoi **(25 Punkte)**

Ein Turm aus $n \in \mathbb{N}_{>0}$ unterschiedlich großen gelochten Scheiben soll von einem Stab S_1 auf einen zweiten Stab S_2 unter Zuhilfenahme eines Hilfsstabes S_3 verschoben werden (das folgende Bild zeigt die Situation für den Fall $n = 4$).



Dabei müssen die folgenden beiden Regeln beachtet werden: Erstens darf immer nur eine Scheibe pro Zug bewegt werden (d.h., es darf immer nur die oberste Scheibe eines Turms bewegt werden). Zweitens darf nie eine größere Scheibe auf einer kleineren Scheibe liegen.

(a) Beschreiben Sie, wie der Turm im Fall $n = 4$ von S_1 nach S_2 verschoben werden kann.

(b) Beweisen Sie, dass es für alle $n \in \mathbb{N}_{>0}$ möglich ist, die n Scheiben in $2^n - 1$ Schritten von S_1 nach S_2 zu verschieben.

Hinweis: Beweisen Sie zuerst durch vollständige Induktion nach n, dass die folgende Aussage für alle $n \in \mathbb{N}$ mit $n \geq 1$ gilt:

A(n): Seien $i, j \in \{1,2,3\}$ mit $i \neq j$, sei $m \in \mathbb{N}$ mit $m \geq n$, und seien m Scheiben so auf die drei Stäbe verteilt, dass gilt: (1) auf S_i liegen mindestens n Scheiben, und (2) falls auf den anderen beiden Stäben Scheiben liegen, so sind diese jeweils größer als die obersten n Scheiben auf S_i. Dann lassen sich die obersten n Scheiben auf S_i in $2^n - 1$ Schritten nach S_j verschieben, ohne eine der anderen Scheiben zu bewegen.