Übungsblatt 02

12.05.2022

Maurice Donner
Jan Hubrich
Adrian Müller

Aufgabe 1

1. (Def) $T(n) \le \begin{cases} 14 & n = 1 \\ 3n + 4T([n/4]) & n > 1 \end{cases}$

z: $T(n) \le 20n^2 - 6n$, if $n = 4^k$ with $k \in \mathbb{N}$

IA $n=1$

$T(1) = 14$ ✓

IH Let $T(n) \le 20n^2 - 6n$ for any $n \in \mathbb{N}$.

IS z: $T(n+1) \le 20(n+1)^2 - 6(n+1)$ $\qquad 20n^2 + 20n + 20$

Def: $T(n+1) = 3(n+1) + 4T([n+1]/4)$

$= 3n + 3 + 4T(n/4) + 4T(1)$

$= 3n + 4T([n/4]) + 3 + 4T(1)$

$\overset{(IH)}{\le} 20n^2 - 6n + 20 - 6$

$n \in \mathbb{N}$

$\le 20n^2 + 20n + 20 - 6(n+1)$

$= 20(n+1)^2 - 6(n+1)$ //

2. $T(n) = \begin{cases} c_0 n & n \le n_0, \; n_0 > 20 \\ T([n/2]) + T(\frac{2}{5}n + 1) + c_1 n & n > n_0 \end{cases}$

gesucht: $f \longrightarrow T(n) \in O(f(n))$

Master-Theorem:

$T([n/2]) + T(\frac{2}{5}n) + \boxed{\overset{O(n)}{c_1 n}}$ $\qquad f(n) = n$

Let $n = 21$: $T(21) = T(21/2) + T(47/5) + 21 c_1$

$= \frac{21}{2} c_0 + \frac{47}{5} c_0 + 21 c_1$

$O(n) + O(n) + O(n) = O(n)$

Vorlesung (Vereinfacht)

$$r(n) = \begin{cases} \Theta(n) & \text{falls } d < b \\ \Theta(n \log n) & \text{falls } d = b \\ \Theta(n^{\log_b d}) & \text{falls } d > b. \end{cases}$$

Indian youtuber (d = a)

$T(n) = aT(n/b) + f(n)$ $f(n) = O(n^k \log^p n)$

$a \geq 1$
$b > 1$

1. $\log_b a$
2. k

Case 1: if $\log_b a > k$ then $\Theta(n^{\log_b a})$

Case 2: if $\log_b a = k$

if $p > -1$ $\Theta(n^k \log^{p+1} n)$

if $p = -1$ $\Theta(n^k \log \log n)$

if $p < -1$ $\Theta(n^k)$

Case 3: if $\log_b a < k$

if $p \geq 0$ $\Theta(n^k \log^p n)$

if $p < 0$ $O(n^k)$

Aufgabe 2

a) $A(1) = 1$ und für $n = 2^k$ $k \in \mathbb{N}$: $A(n) = \underbrace{1 \cdot A(n/2)}_{o(1)} + \underbrace{\tilde{c}n}_{O(n)}$

- Geometrisch schrumpfende Reihe
  -> erste Rekursionsebene kostet konstanten Teil der Arbeit

$d = 1$ $b = 2$

$\boxed{d < b}$

$\log_b d = \log_2 1 = 0$ $k = 1$ => $A(n) = \Theta(n)$

b) $B(n) = 9B(n/3) + 4n$ $d = 9$ $b = 3$

$\log_b d = \log_3 9 = 2$ $k = 1$ $\boxed{d > b}$

=> $B(n) = \Theta(n^{\log_b d}) = \Theta(n^2)$

c) $C(n) = C(n/4) + \underbrace{n}_{O(n)} + \underbrace{6}_{O(1)}$ $b = 4$ $d = 1$, $\boxed{b > d}$

$\log_b d = 0$, $k = 1$

=> $C(n) = \Theta(n)$

d) $D(n) = 4D(n/4) + \underbrace{C(n)}_{\Theta(n)}$ $b = 4, d = 4$ $\boxed{b = d}$

$\log_b d = 1$, $k = 1$

=> $D(n) = \Theta(n \log n)$

Aufgabe 3

```
procedure merge(A : Array [1..n1] of N≥0, B : Array [1..n2] of N≥0)
precondition A[i] ≤ A[j] ∀i ≤ j mit i, j ∈ {1, ..., n1}
precondition B[i] ≤ B[j] ∀i ≤ j mit i, j ∈ {1, ..., n2}
A[n1 + 1] := ∞, B[n2 + 1] := ∞
n := n1 + n2
jA := 1, jB := 1;
for i := 1 to n do
C[i] = min(A[jA], B[jB])
if A[jA] < B[jB] then
jA = jA + 1
else
jB = jB + 1
x invariant C[1..i] enthält genau A[1..jA − 1], B[1..jB − 1]
x invariant B[k] ≤ A[jA] ∀k ∈ {1..jB − 1}, A[k] ≤ B[jB] ∀k ∈ {1..jA − 1}
x invariant C[1..i] ist sortiert
x assert jA = n1 + 1, jB = n2 + 1
x postcondition C[i] ≤ C[j] ∀i ≤ j, i, j ∈ {1, ..., n}
x postcondition C[1..n] enthält genau A[1..n1], B[1..n2]
return C
```

inv $C[1..i]$ enthält genau $A[1..j_A-1]$, $B[1..j_B-1]$

init: $i = 1$ $j_A = 1$ $j_B = 1$

$C = [\ ]$ ✓

maintenance case $A[j_A] < B[j_B]$

$C[1..i] = [\underbrace{\ldots,}\ A[j_A]]$

enthält genau $A[1..j_A-1]$, $B[1,..,j_B-1]$

$\boxed{j_A \to j_A + 1}$

$C[1..i]$ enthält genau $A[1,..j_A+1-1]$, $B[1,..,j_B-1]$ ✓

Analog für case $A[j_A] > B[j_B]$

inv $B[k] \leq A[j_A]$ $\forall k \in \{1..j_B-1\}$, $A[k] \leq B[j_B]$ $\forall k \in \{1..j_A-1\}$

init $B[1] \leq A[1]$ and $A[1] \leq B[1]$ ✓ weil $A[1] = B[1] = 1$

maintenance case $A[j_A] < B[j_B]$

$C[1..i] = [\underbrace{\ldots,}\ A[j_A]]$

Kein Beweis notw. ← $B[k] \leq A[j_A]$ holds for all $k \in \{1,..j_B-1\}$

Beweis nur hier ← $A[k] \leq B[j_B]$ holds for all $k \in \{1..j_A-1\}$

$\boxed{j_A \to j_A + 1}$

$A[k] \leq B[j_B]$ holds for $k \in \{1..j_A+1-1\}$ ✓ (see case)

Analog für case $A[j_A] > B[j_B]$, Beweise stattdessen $B[k] \leq A[j_A]$ $\forall k \in \{1..j_B-1\}$

inv $C[1 \dots i]$ ist sortiert

init $C[\,]$ ist sortiert ✓

maintenance case $A[j_A] < B[j_B]$

$C[1 \dots i] = [\underbrace{\dots}_{}, A[j_A]]$

sortiert, enthält $A[1 \dots j_A - 1]$ $B[1 \dots j_B - 1]$

$A[j_A] > A[j_A - 1]$ ✓ Da sortiert

$A[j_A] > B[j_B - 1]$

wahr, da $B[j_B - 1] < A[j_A]$ aus vorheriger iteration

$\boxed{j_A \to j_A + 1}$

$A[j_A + 1] > A[j_A]$ ✓ da sortiert

$A[j_A + 1] > B[j_B - 1]$ ✓ da $A[j_A + 1] > A[j_A] > B[j_B - 1]$

Assertion $j_A = n_1 + 1$, $j_B = n_2 + 1$

Def. $n_1 + n_2 =: n$

init $j_A = 1 = j_B$

loop

Grenzfallbetrachtung

Angenommen $A[j_A] < B[j_B]$ $\forall j_A \le n_1$, $A[j_A] = \infty$ sonst

Nach n_1 iterationen ist die obrige Bedingung nicht mehr erfüllt

$\to j_A = 1 + n_1 \cdot (1) = 1 + n_1$

Danach $j_B = 1 + n_2 \cdot (1) = 1 + n_2$ (nach definition)

Angenommen $A[j_A] \ge B[j_B]$ $\forall j_B \le n_2$, $B[j_B] = \infty$ sonst

$\to j_B = 1 + n_2 \cdot (1) = 1 + n_2$

Danach $j_A = 1 + n_1 \cdot (1) = 1 + n_1$

Spezialfall $A[j_A] = B[j_B] = \infty$, $i = n_1 + n_2 = n \to$ exit condition post condition $C[i] \leq C[j] \ \forall i \leq j \quad i,j \in \{1,\dots,n\}$

Setze $i = n$ und nutze invariant 2: $C[1,\dots,i]$ ist sortiert

$\Rightarrow C[i] \leq C[j]$, wenn $i \leq j$

post condition $C[1,\dots,n]$ enthält genau $A[1,\dots,n_1]$ $B[1,\dots,n_2]$

Setze $i = n$, nutze Assertion $j_A = n_1+1$ $j_B = n_2+1$

und nutze invariant 1

$\Rightarrow C[1 .. i]$ enthält genau $A[1 ... j_A - 1]$, $B[1,..,j_B - 1]$

Assert. inv. 1 $\Rightarrow C[1 ... n]$ enthält genau $A[1 ... n_1]$, $B[1 ... n_2]$ //

Laufzeit: Betrachte for-loop

```
for i:=1 to n do
op 1      C[i] = min (A[jA], B[jB])
          if A[jA] < B[jB] then
op 2a        jA = jA+1
          else
op 2b        jB = jB+1
```

n-1 iterationen

2 operationen

$\Rightarrow \boxed{2n + c}$ Operationen

(c sind alle operationen außerhalb des loops)

$2n+c \leq \Theta(n)$

2. int n, int f = 1

for i = 1, i ≤ n

$f = f \cdot i$

invariant $f = i!$

init $i = 1$, $f = 1$, $f = 1!$ ✓

maintenance Sei $f = i!$ $\quad i' = i+1$, $f' = f \cdot i'$ $\quad$ z: $f' = i'!$

$f = i! \Rightarrow f(i+1) = i!(i+1) \Rightarrow f \cdot i' = i'! \Rightarrow f' = i'!$ //