TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

WS 11/12

Einführung in die Informatik II **Übungsblatt 12**
Univ.-Prof. Dr. Andrey Rybalchenko, M.Sc. Ruslán Ledesma Garza 31.01.2012

Dieses behandelt Kapitel 11.1 – 11.3, 11.5, 11.7 – 1.11 aus dem Buch zur Vorlesung. Lesen Sie diese Kapitel!

Aufgabe 11.1 Betrachten Sie die Definition der Prozedur @.

$$@ : \mathcal{L}(X) \times \mathcal{L}(X) \to \mathcal{L}(X)$$

$$nil \ @ \ ys = ys$$
$$(x :: xs) \ @ \ ys = x :: (xs \ @ \ ys)$$

Geben Sie die Rekursionsfolge und die Laufzeit der Prozedur @ für das Argument $([1,2], [3,4,5,6])$ an.

Lösungsvorschlag 11.1:

a) Rekursionsfolge:
$$([1,2],\ [3,4,5,6]) \to ([2],\ [3,4,5,6]) \to ([],\ [3,4,5,6]) \to$$

b) Laufzeit: 3

Aufgabe 11.2 Geben Sie für die folgende Prozedur eine Größenfunktion und die entsprechende Laufzeitfunktion an:

$$revi \in \mathcal{L}(X) \times \mathcal{L}(X) \to \mathcal{L}(X)$$

$$revi \ (xs, nil) = xs$$
$$revi \ (xs, y :: ys) = revi \ (y :: xs, ys)$$

Lösungsvorschlag 11.2:

a) Größenfunktion:
$$\lambda(xs, ys) \in \mathcal{L}(X) \times \mathcal{L}(X) \,.\ \mid ys \mid$$

b) Laufzeitfunktion:
$$\lambda n \in \mathbb{N} \,.\, n + 1$$

Aufgabe 11.9 Geben Sie eine rekursive Beschreibung der Laufzeitfunktion der Prozedur @ gemäß der in §11.2.1 engegebenen Größenfunktion an.

Lösungsvorschlag 11.9:

$$r \in \mathbb{N} \to \mathbb{N}_+$$

$$r \ 0 = 1 \tag{1}$$
$$r \ n = 1 + r \ (n-1) \text{ für } n > 0 \tag{2}$$

Aufgabe 11.11.a Beweisen Sie die folgende Gleichung:

$$O(\frac{n^2}{7} + 789n \cdot \log n + 45n + 77) = O(n^2)$$

Lösungsvorschlag 11.11.a:

$$\begin{array}{lll} O(n^2) & = O(\frac{n^2}{7}) & \text{Prop. 11.4} \\ & = O(\frac{n^2}{7} + 789n \cdot \log\ n) & \text{Props. 11.3, 11.4} \\ & = O(\frac{n^2}{7} + 789n \cdot \log\ n + 45n) & \text{Props. 11.3, 11.4} \\ & = O(\frac{n^2}{7} + 789n \cdot \log\ n + 45n + 77) & \text{Props. 11.3, 11.4} \end{array}$$

Aufgabe 11.23 Betrachten Sie die Definition der Prozeduren *insert* und *isort*.

$$insert : \mathbb{Z} \times \mathcal{L}(\mathbb{Z}) \to \mathcal{L}(\mathbb{Z}) \qquad isort : \mathcal{L}(\mathbb{Z}) \to \mathcal{L}(\mathbb{Z})$$

$$\begin{array}{rl} insert\ (x, nil) = & [x] \\ insert\ (x, y :: ys) = & \text{if } x \leq y \text{ then } x :: y :: ys \\ & \text{else } y :: insert\ (x, ys) \end{array} \qquad \begin{array}{rl} isort\ nil = & nil \\ isort\ (x :: xs) = & insert\ (x, isort\ xs) \end{array}$$

Geben Sie eine explizite Darstellung der Worst-Case-Laufzeitfunktion von *isort*.

Lösungsvorschlag 11.23:

Betrachten Sie die folgenden Worst-Case Biespiel, Laufzeitfunktion und Kostenfunktion.

a) Worst-Case: umgekehrte Liste.

$$\begin{array}{ll} isort\ [1,0] & = insert\ (1, isort\ [0]) \\ & = insert\ (1, insert\ (0, isort\ [])) \\ & = insert\ (1, insert\ (0, [])) \\ & = insert\ (1, [0]) \\ & = 0 :: insert\ (1, []) \\ & = [0, 1] \end{array}$$

b) Worst-Case-Laufzeitfunktion:

$$\begin{array}{ll} r\ n & = 1 \\ r\ n & = r\ (n-1) + g\ n \quad \text{falls } 0 < n \end{array}$$

c) Worst-Case Kostenfunktion:

$$g\ n \quad = n + 1$$

Die folgende Funktion ist die explizite Darstellung der Worst-Case-Laufzeitfunktion von *isort*.

$$\begin{array}{ll} r\ n & = 1 + g(1) + \ldots + g(n) \\ & = 1 + (1+1) + \ldots + (1+n) \\ & = 1 + \underbrace{(1 + \ldots + 1)}_{\text{n-mal}} + (1 + \ldots + n) \\ & = 1 + n + (1 + \ldots + n) \\ & = 1 + n + \frac{n}{2}(n+1) \qquad \text{Gaußsche Formel} \\ & = \frac{1}{2}(n^2 + 3n + 2) \end{array}$$

Aufgabe 11.28.fact Bestimmen Sie die Komplexitäten der Prozedur *fac* mithilfe des polynomiellen Rekurrenzsatzes. Geben Sie jeweils eine rekursive Beschreibug der Laufzeitfunktion und eine explizite Beschreibung der Kostenfunktion an.

Lösungsvorschlag 11.28.fact:

a) Laufzeitfunktion:

$$\begin{array}{ll} r\ n & = 1 \\ r\ n & = r\ (n-1) + g\ n \quad \text{falls } 0 < n \end{array}$$

b) Kostenfunktion:

$$g\ n \quad = 1$$

c) Komplexität: Verwenden Sie die Polynomieller Rekurrenzsatz.

$$\left.\begin{array}{l} r\ n = r(n-1) + g\ n \text{ für alle } n > 1 \\ O(g) = O(n^0) \end{array}\right\} \Rightarrow O(r) = O(n)$$

Aufgabe 11.31.2^n Geben Sie eine möglichst einfache Prozedur $\mathbb{N} \to \{0\}$ mit der Komplexität $O(2^n)$ an.

Lösungsvorschlag 11.31.2^n**:**

a) Laufzeitfunktion:

$$\begin{array}{ll} r\ n & = 1 \\ r\ n & = 2r(n-1) + g\ n \quad \text{falls } 0 < n \end{array}$$

b) Kostenfunktion:

$$g\ n \quad = 1$$

c) Prozedur: `fun f x = if x = 0 then 0 else f (x - 1) + f (x - 1)`

Aufgabe 11.38.a Geben Sie die Komplexität der folgenden Funktion $f \in \mathbb{N} \to \mathbb{N}$ möglichst einfach an.

$$f\ n = \text{if } n < 7 \text{ then } 1 \text{ else } 3f(n-1) + n \cdot \log n + 1$$

Lösungsvorschlag 11.38.a:

a) Laufzeitfunktion:

$$\begin{array}{lll} r\ n & = 1 & \text{falls } 0 \leq n < 7 \\ r\ n & = r(n-1) + g\ n & \text{falls } n \geq 7 \end{array}$$

b) Kostenfunktion:

$$g\ n \quad = 1$$

c) Komplexität: Verwenden Sie die Polynomielle Rekurrenzsatz.

$$\left.\begin{array}{l} r\ n = r(n-1) + g\ n \text{ für alle } n \geq 7 \\ O(g) = O(n^0) \end{array}\right\} \Rightarrow O(r) = O(n)$$