REACTIVE SYSTEMS GROUP

Universität des Saarlandes

Prof. Bernd Finkbeiner, Ph.D.
Markus Rabe, M.Sc.

Programmierung 1 (SS 2010) - 11. Übungsblatt

http://react.cs.uni-saarland.de/prog1/

Lesen Sie im Buch bis zum Ende von Kapitel 11

Aufgabe 11.1

Geben Sie die Rekursionsfolge und die Laufzeit von *foldl* für das Argument (f, 5, $[2, 6, 3]$) mit $f = \lambda\,(x, y).\ x + y$ an.

Aufgabe 11.2

Betrachten Sie die Prozedur *insert*, die ein neues Element in eine Liste einfügt (siehe Sortieren durch Einfügen, Kapitel 5.1 im Buch):

$$\begin{aligned}&\mathit{insert} : \mathbb{Z} \times \mathcal{L}(\mathbb{Z}) \to \mathcal{L}(\mathbb{Z})\\ &\mathit{insert}(x, \mathit{nil}) = [x]\\ &\mathit{insert}(x, y :: \mathit{yr}) = \text{if } x \leq y \text{ then } x :: y :: \mathit{yr} \text{ else } y :: \mathit{insert}(x, \mathit{yr})\end{aligned}$$

a) Wählen Sie eine Größenfunktion für *insert*.

b) Geben Sie für die Größe 4 ein Argument mit minimaler Laufzeit und ein Argument mit maximaler Laufzeit an (in Bezug auf Argumente der Größe 4).

c) Welche minimale und welche maximale Laufzeit hat die Prozedur *insert* für Argumente der Größe n?

d) Geben Sie die Laufzeitfunktion von *insert* an.

Aufgabe 11.3

Wir betrachten die bekannte Prozedur zur Berechnung natürlicher Quadratwurzeln:

$$\begin{aligned}&p : \mathbb{N}^2 \to \mathbb{N}\\ &p(n, x) = \text{if } n^2 \leq x \text{ then } p(n + 1,\ x) \text{ else } n - 1\end{aligned}$$

a) Geben Sie die Laufzeit von p für ein Argument (n, x) an.

b) Geben Sie die Laufzeitfunktion der Prozedur p an gemäß der Größenfunktion

$$\lambda\,(n, x).\ \text{if } n^2 > x \text{ then } 0 \text{ else } x - n^2 + 1.$$

c) Geben Sie die Komplexität der Laufzeitfunktion an.

Aufgabe 11.4

Geben Sie eine zu *ntree* (siehe Buch, S. 221) semantisch äquivalente Prozedur $tree : \mathbb{N} \to \mathcal{T}$ an, die die Laufzeitfunktion $\lambda n.\, n+1$ hat. Schreiben Sie die Prozeduren *tree* und *ntree* in Standard ML und überzeugen Sie sich mit einem Interpreter davon, dass *tree* sehr viel schneller rechnet als *ntree*.

Aufgabe 11.5

Beweisen Sie unter Verwendung von Propositionen 11.3 und 11.4:

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

Aufgabe 11.6

Sei $O(1) \subseteq O(f)$. Beweisen Sie:

$$O(fn) \subset O(n \cdot fn)$$

Hinweis: Zeigen Sie zuerst $O(fn) \subseteq O(n \cdot fn)$ und zeigen Sie dann $O(n \cdot fn) \nsubseteq O(fn)$ durch Widerspruch. Benutzen Sie dabei jeweils Proposition 11.2.

Aufgabe 11.7

Zeigen Sie: $(\lambda\, x.\ \text{if } x \text{ div } 2 = 0 \text{ then } 1 \text{ else } 2) \preceq (\lambda\, x.\ \text{if } x \text{ div } 2 = 0 \text{ then } 2 \text{ else } 1)$

Aufgabe 11.8

Geben Sie eine baumrekursive Prozedur $\mathbb{N} \to \mathbb{N}$ an, deren Komplexität konstant ist. Hinweis: Formulieren Sie die Prozedur so, dass sie ab einer bestimmten Argumentgröße nicht mehr rekurriert. Das Beispiel zeigt, dass man pathologische Prozeduren konstanter Komplexität konstruieren kann, die für alle praktisch relevanten Argumente sehr hohe Laufzeiten haben.

Aufgabe 11.9

a) Geben Sie die Komplexität der durch die folgende Gleichung rekursiv definierten Funktion $f \in \mathbb{N} \to \mathbb{N}$ an:

$$f\ n = \text{if } n = 0 \text{ then } 0 \text{ else } f(n-1)$$

b) Geben Sie die Komplexität der durch die folgende Gleichung rekursiv definierten Prozedur $f : \mathbb{N} \to \mathbb{N}$ an:

$$f\ n = \text{if } n = 0 \text{ then } 0 \text{ else } f(n-1)$$

Aufgabe 11.10

Bestimmen Sie die Laufzeitfunktion und Komplexität der Prozedur *merge*, die beim Sortieren durch Mischen verwendet wird:

$$\begin{aligned}
&\mathit{merge} : \mathcal{L}(\mathbb{Z}) \times \mathcal{L}(\mathbb{Z}) \to \mathcal{L}(\mathbb{Z})\\
&\mathit{merge}\ (\mathit{nil}, \mathit{ys}) = \mathit{ys}\\
&\mathit{merge}\ (\mathit{xs}, \mathit{nil}) = \mathit{xs}\\
&\mathit{merge}\ (x :: \mathit{xr}, y :: \mathit{yr}) = \text{if } x \leq y \text{ then } x :: \mathit{merge}(\mathit{xr}, y :: \mathit{yr}) \text{ else } y :: \mathit{merge}(x :: \mathit{xr}, \mathit{yr})
\end{aligned}$$

Dabei soll die Größenfunktion $\lambda\,(\mathit{xs}, \mathit{ys}).\ |\mathit{xs}| + |\mathit{ys}|$ zugrunde gelegt werden.

Aufgabe 11.11

Bestimmen Sie die Komplexität der Prozedur *revi* (siehe Aufgabe 10.10 auf dem vorigen Übungsblatt) mit Hilfe eines Rekurrenzsatzes. Geben Sie eine rekursive Beschreibung der Laufzeitfunktion und eine explizite Beschreibung der Kostenfunktion an.

Aufgabe 11.12

Geben Sie die Komplexitäten der im Folgenden rekursiv definierten Funktionen $f \in \mathbb{N} \to \mathbb{R}$ möglichst einfach an.

a) $f(n) = \text{if } n < 3 \text{ then } 1 \text{ else } f(n-2) + 5$

b) $f(n) = \text{if } n < 27 \text{ then } 1 \text{ else } f(n-7) + 3n^7 + 7n^3$

Aufgabe 11.13

Geben Sie möglichst einfache Prozeduren $\mathbb{N} \to \{0\}$ an, die für $\lambda\, n \in \mathbb{N}.\ n$ als Größenfunktion die Komplexitäten $O(1)$, $O(\log n)$, $O(n \log n)$, $O(n^2)$ und $O(2^n)$ haben. Für $O(1)$ und $O(\log n)$ sollen keine Nebenkosten anfallen. Für $O(n \log n)$ und $O(n^2)$ sollen lineare Nebenkosten anfallen.

Aufgabe 11.14

Seien zwei Prozeduren wie folgt gegeben:

$$\begin{aligned}
&p : \mathbb{N} \to \mathbb{N}\\
&p\ n = \text{if } n < 5 \text{ then } n \text{ else } p(n-2)
\end{aligned}$$

$$\begin{aligned}
&q : \mathbb{N} \to \mathbb{N}\\
&q\ n = \text{if } n < 12 \text{ then } 3n \text{ else } q(n-1) + q(n-1) + p\ n
\end{aligned}$$

Bestimmen Sie die Komplexität der Prozedur q für die Größenfunktion $\lambda\, n \in \mathbb{N}.\ n$.

Aufgabe 11.15

Sei $f : \mathbb{N} \to \mathbb{N}$ eine Funktion. Wir nennen f *monoton*, wenn

$$\forall n \in \mathbb{N}\ \forall k \in \mathbb{N} : k \leq n \Rightarrow f\ k \leq f\ n.$$

Stellen Sie die Laufzeitfunktion von *exp* (siehe Seite 234 im Buch) rekursiv dar und beweisen Sie mithilfe von Induktion, dass sie monoton ist.

Aufgabe 11.16

Betrachten Sie die Prozedur

$$\begin{aligned} &p : \mathbb{N} \to \mathbb{N} \\ &p(n) = \text{if } n < 23 \text{ then } 7 \text{ else } p(\lfloor \frac{n}{3} \rfloor) \end{aligned}$$

a) Geben Sie die Komplexität der Prozedur für die Größenfunktion $\lambda\, n \in \mathbb{N}.\ n$ an.

b) Geben Sie die Komplexität der Ergebnisfunktion der Prozedur an.

Aufgabe 11.17

Geben Sie Deklarationen an (in Standard ML), die den Bezeichner e an die abstrakte Darstellung des folgenden Ausdrucks binden:

$$\mathit{fn}\ f : \mathit{int} \to \mathit{int} \Rightarrow\ \mathit{fn}\ n : \mathit{int} \Rightarrow\ \text{if } n \leq \mathit{0} \text{ then } \mathit{1} \text{ else } n * f(n - \mathit{1})$$

Aufgabe 11.18

Geben Sie Typumgebungen an, für die der Ausdruck if *true* then x else y zulässig beziehungsweise unzulässig ist.