Christian-Albrechts-Universität zu Kiel
Institut für Informatik
Lehrstuhl für Programmiersprachen und Übersetzerkonstruktion

Prof. Dr. Michael Hanus, Jan Tikovsky

Institut für Informatik

4. Übung zur Vorlesung „Programmierung“

Serie 4

Abgabe am Freitag, 21. November 2014 - 08:00

Gehen Sie bei allen Programmieraufgaben nach der Konstruktionsanleitung aus der Vorlesung vor:

1. Schreiben Sie einen kurzen Kommentar, der beschreibt was die Prozedur berechnet.
2. Geben Sie die Signaturvereinbarung der Prozedur an.
3. Definieren Sie mindestens drei Testfälle für alle von Ihnen implementierten Prozeduren, wobei insbesondere auch Randfälle getestet werden sollten. Zusätzlich sollten Sie darauf achten, dass alle Programmteile durch ihre Testfälle mindestens einmal ausgeführt werden. Erkennen können Sie das in DrRacket daran, dass nach Drücken des “Start”-Knopfes kein Programmteil mehr dunkel unterlegt ist.

Präsenzaufgabe 1 - Quadratsumme

Die Funktion $f : \mathbb{N} \mapsto \mathbb{N}$ sei definiert durch $f(n) = \sum_{i=1}^{n} i^2$.

1. Schreiben Sie eine Funktion `f-rek`, die f in einem rekursiven Prozess berechnet.
2. Schreiben Sie eine Funktion `f-iter`, die f in einem iterativen Prozess berechnet.
3. Geben Sie die Größenordnung für die Laufzeit und den Speicherverbrauch von `f-rek` und `f-iter` an.

Präsenzaufgabe 2 - Zweierlei Addition

Die Funktionen `add1` und `add2` seien definiert durch:

```
; Berechnet die Summe zweier natürlicher Zahlen
(: add1 (natural natural -> natural))
(define add1
  (lambda (n m)
    (if  (= n 0)
         m
         (inc (add1 (dec n) m)))))

; Berechnet die Summe zweier natürlicher Zahlen
(: add2 (natural natural -> natural))
(define add2
  (lambda (n m)
    (if  (= n 0)
         m
         (add2 (dec n) (inc m)))))
```

Die beiden Funktionen addieren jeweils zwei natürliche Zahlen. Sie basieren beide auf der elementaren Inkrementierung `inc` und Dekrementierung `dec` definiert durch:

```
; Inkrementiert eine natürliche Zahl
(: inc (natural -> natural))
(define inc (lambda (n) (+ n 1)))
```

```
; Dekrementiert eine natürliche Zahl
(: dec (natural -> natural))
(define dec (lambda (n) (- n 1)))
```

1. Erläutern Sie mit Hilfe des Substitutionsmodells die Berechnungsprozesse, die bei der Auswertung von `(add1 1 3)` und `(add2 1 3)` in Scheme erzeugt werden.
2. Sind diese Prozesse rekursiv oder iterativ? In welchem Zusammenhang steht die Struktur des rekursiven Aufrufes mit der Art des Berechnungsprozesses?

Aufgabe 3 - Binomialkoeffizienten

8 Punkte

Die Zahlen, die Sie in Serie 3, Aufgabe 3 berechnet haben, heißen *Binomialkoeffizienten*. Man schreibt $\binom{n}{k}$ für die k-te Zahl in der n-ten Zeile des Pascalschen Dreiecks.

Es lässt sich zeigen, dass für alle $n, k \in \mathbb{N}$ mit $n \geq k$ gilt:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

1. Nutzen Sie diese Formel und die Fakultätsfunktion, um eine Funktion `binom` zu definieren, die den Wert von $\binom{n}{k}$ berechnet.
2. Aus der obigen Formel lassen sich einige Faktoren herauskürzen. Nutzen Sie das, indem Sie eine Prozedur `binom` definieren, die den Wert von $\binom{n}{k}$ in einem *iterativen* Prozess berechnet. Es sollen hierbei so viele Faktoren wie möglich herausgekürzt werden.
3. Geben Sie den Platzbedarf und die Laufzeit der beiden `binom`-Funktionen unter Verwendung der $\mathcal{O}$-Notation an. Bestimmen Sie zusätzlich den Platzbedarf und die Laufzeit der Funktion `pascal` aus Serie 3, Aufgabe 3. Begründen Sie kurz Ihre Antworten.

Aufgabe 4 - Schnelles iteratives Potenzieren

5 Punkte

In der Vorlesung haben Sie die folgende, rekursive Funktion zur schnellen Berechnung von Potenzen kennengelernt:

```
; Berechnet die n-te Potenz einer Zahl
(: expnt (number natural -> number))
(define expnt
  (lambda (b n)
    (cond ((= n 0) 1)
          ((even? n) (square (expnt b (/ n 2))))
          (else (* b (expnt b (- n 1)))))))
```

Schreiben Sie eine Funktion `expnt-iter`, die das schnelle Potenzieren in einem iterativen Prozess durchführt.

Aufgabe 5 - Rekursiv und iterativ

7 Punkte

Die Funktion $f : \mathbb{N}_0 \mapsto \mathbb{N}_0$ sei definiert durch:

$$f(n) = n \text{ für } n \leq 2$$
$$f(n) = f(n-1) + 2f(n-2) + 3f(n-3) \text{ für } n > 2$$

1. Schreiben Sie eine Prozedur `f-rek`, die f in einem rekursiven Prozess berechnet.
2. Schreiben Sie eine Prozedur `f-iter`, die f in einem iterativen Prozess berechnet.
3. Geben Sie die Laufzeiten der beiden Prozeduren in der $\mathcal{O}$-Notation an. Begründen Sie Ihre Antworten kurz.