III. Übungsblatt "Informatik I" WS 95/96

Christian Schulte, Prof. Gert Smolka

Ausgabe: 3. November 1995 **Abgabe:** 10. November 1995 vor der Vorlesung

Aufgabe III.1 (Schnelle Exponentialfunktion (8)) Um die Frustration mit dem Konditional vom letzten Übungsblatt zu überwinden, wendet sich Alyssa P. Hacker der Aufgabe zu, eine besonders effiziente Prozedur zur Berechnung der Exponentialfunktion zu entwickeln. Dabei stößt sie auf die beiden folgenden Gleichungen:

$$\exp(n, 2k) = \exp(n^2, k) \qquad \text{falls } n, k \in \mathbb{N}$$

und

$$\exp(n, 2k + 1) = n * \exp(n, 2k) \qquad \text{falls } n, k \in \mathbb{N}.$$

$\mathbb{N}$ bezeichnet die Menge der natürlichen Zahlen. Davon ausgehend definiert Alyssa ihre effiziente Exponentialfunktion.

1. Tun Sie es Alyssa gleich: Geben Sie eine induktive Definition der Exponentialfunktion an, die von den beiden obigen Gleichungen Gebrauch macht.

2. Schreiben Sie eine Scheme Prozedur `(gerade? n)`, die genau dann zu `#t` evaluiert, wenn `n` gerade ist.

   Hinweis: Sie können die Scheme Prozedur `(remainder n m)` benutzen, die den Rest der exakten Division von `n` durch `m` liefert. Zum Beispiel evaluiert `(remainder 4 3)` zu `1`, `(remainder 5 3)` zu `2` und `(remainder 6 3)` zu `0`.

3. Schreiben Sie entsprechend Ihrer induktiven Definition eine rekursive Scheme Prozedur `(exp n m)`. Benutzen Sie dabei die Prozedur `gerade?`.

4. Zeichnen Sie die Aufrufbäume für `(exp 2 7)` und `(exp 2 8)`. Wie tief sind diese Bäume im Vergleich zu denen, die Sie mit der Definition der Exponentialfunktion vom Übungsblatt I erhalten?

5. Wie muß die Zahl `m` bei der Berechnung von `(exp n m)` beschaffen sein, so daß jeweils in Relation zur Größe von `m`, die Tiefe des Aufrufbaumes minimal beziehungsweise maximal ist?

6. Schreiben Sie nun eine iterative Version der obigen Exponentialfunktion (d.h., eine, die nur konstant viel Speicher braucht).

Aufgabe III.2 (Logarithmusfunktion (6)) Gegeben sei eine natürliche Zahl $n \geq 1$. Der ganzzahlige Logarithmus zur Basis 2 von n, geschrieben als $\mathrm{ld}(n)$[1], ist die größte natürliche Zahl $k \geq 0$, so daß gilt:

$$n \geq 2^k.$$

Beispielsweise ist: $\mathrm{ld}(1) = 0$, $\mathrm{ld}(2) = 1$, $\mathrm{ld}(3) = 1$, $\mathrm{ld}(4) = 2$, $\mathrm{ld}(5) = 2$ und $\mathrm{ld}(8) = 3$.

1. Geben Sie eine induktive Definition von $\mathrm{ld}(n)$ an.

2. Programmieren Sie analog dazu eine rekursive Scheme Prozedur `(ld n)`.

[1] ld kommt aus dem lateinischen: **l**ogarithmus **d**ualis.

Aufgabe III.3 (Ist das Konditional primitiv? (4)) Nachdem Alyssa P. Hacker ein Erfolgserlebnis mit der Entwicklung einer schnellen Exponentialfunktion hatte, geht Sie ihr Problem mit dem selbstdefinierten Konditional, das nur `if` benutzt, noch einmal an. Dabei kommt ihr eine geniale Idee. Sie definiert sich erneut eine Prozedur `(konditional bedingung konsequent alternative)` und testet diese mit den folgenden Beispielen:

```
(define (doppelt x) (x x))
(define (divergiere) (doppelt doppelt))
(konditional (= 2 3) divergiere (lambda () 5))
(konditional (= 2 2) (lambda () 5) divergiere)
```

Endlich hat Sie ihr Problem gelöst, aber wie?

1. Geben Sie eine Definition von `konditional` an, und testen Sie diese mit den obigen Beispielen.
2. Erklären Sie den Unterschied zu der Definition, die Alyssa in Aufgabe II.3 benutzt hat.
3. Definieren Sie mit Hilfe des korrigierten Konditionals die Fakultätsfunktion.

Aufgabe III.4 (Primzahltest (4)) Gegeben sei eine natürliche Zahl `n`. Definieren Sie eine Prozedur `(prim? n)`, die genau dann zu `#t` evaluiert, wenn `n` eine Primzahl ist.

Hinweis: Sie können die Scheme Prozedur `(remainder n m)` benutzen, die Sie bereits aus Aufgabe III.1 kennen.

Bitte denken Sie daran, Ihre Aufgabenblätter mit Namen und Übungsgruppennummer deutlich zu kennzeichnen!