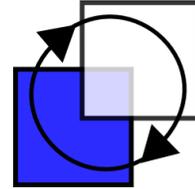


# REACTIVE SYSTEMS GROUP

Universität des Saarlandes

Prof. Bernd Finkbeiner, Ph.D.

Markus Rabe, M.Sc.



## Programmierung 1 (SS 2010) - 15. Übungsblatt

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

Lesen Sie Kapitel 15.9 bis 16.5

### Aufgabe 15.1

Betrachten Sie den Haldenzustand, der sich durch die Ausführung des folgenden Ausdrucks ergibt:

```
(release 0; putList [13,17]; putList [19,23]; update 0 1 6)
```

- Zeichnen Sie die verzeigerte Blockdarstellung der durch die Adresse 2 dargestellten Liste.
- Welche Liste liefert *getList* für die Adresse 2?

### Aufgabe 15.2

Deklariieren Sie Prozeduren, die mit der im Buch durch *putList* formulierten Haldendarstellung von Listen arbeiten:

- Eine Prozedur *null* :  $int \rightarrow bool$ , die testet, ob eine Liste leer ist.
- Eine Prozedur *head* :  $int \rightarrow int$ , die den Kopf einer Liste liefert. Falls die Liste leer ist, soll die Ausnahme *Empty* geworfen werden.
- Eine Prozedur *tail* :  $int \rightarrow int$ , die den Rumpf einer Liste liefert. Falls die Liste leer ist, soll die Ausnahme *Empty* geworfen werden.
- Eine Prozedur *cons* :  $int \rightarrow int \rightarrow int$ , die zu einer Zahl  $x$  und zu einer Darstellung einer Liste  $xr$  eine Darstellung der Liste  $x::xr$  liefert.
- Eine Prozedur *append* :  $int \rightarrow int \rightarrow int$ , die die Konkatenation zweier Listen liefert.
- Zeichnen Sie die verzweigten Blockdarstellungen der durch die Ausführung der folgenden Deklarationen in der Halde dargestellten Listen. Nehmen Sie dabei an, dass die Halde zu Beginn leer ist.

```
val a0 = putList []
val a1 = putList [1,2,3]
val a2 = append a1 a0
val a3 = append a1 a1
```

### Aufgabe 15.3

Überlegen Sie sich, wie Optionen über *int* in der Halde dargestellt werden können. Schreiben Sie entsprechende Prozeduren *putOption* und *getOption*.

### Aufgabe 15.4

Überlegen Sie sich, wie die markierten Bäume aus Kapitel 7.9 in der Halde dargestellt werden können. Schreiben Sie entsprechende Prozeduren *putLtr* und *getLtr*.

### Aufgabe 15.5

Schreiben Sie eine Prozedur *instantiate*:  $(var \rightarrow int) \rightarrow int \rightarrow unit$ , die die Variablen eines in der Halde dargestellten Ausdrucks gemäß einer Umgebung in Konstanten umwandelt. Dabei sollen keine neuen Zellen alloziert werden.

### Aufgabe 15.6

Deklariieren Sie eine Prozedur *eval*:  $(var \rightarrow int) \rightarrow int \rightarrow int$ , die einen in der Halde dargestellten Ausdruck gemäß einer Umgebung evaluiert.

### Aufgabe 15.7

Wie viele Zellen allozieren die folgenden Prozeduren für Listen der Länge  $n$ ?

- a) *length*
- b) *map* ( $fn\ x \Rightarrow x$ )
- c) *foldl op*:
- d) *exists* ( $fn\ x \Rightarrow x > 5$ )

### Aufgabe 15.8

Schreiben Sie Maschinenprogramme, die die folgenden Ausdrücke auswerten. Testen Sie Ihre Programme mit den Prozeduren *exec*, *load* und *step* der virtuellen Maschine, welche Sie hier finden: <http://www.ps.uni-saarland.de/prog-buch/interpreter.html>.

- a)  $17 - 3 - 5$
- b)  $17 - (3 - 5)$
- c)  $(7 - 3) \cdot (8 + 17) + 6$

### Aufgabe 15.9

Schreiben Sie eine Prozedur  $compile : exp \rightarrow code$ , die einen arithmetischen Ausdruck des unten stehenden Typs in ein Maschinenprogramm übersetzt, welches den Ausdruck auswertet.

```
datatype exp =  
  Con of int  
  | Add of exp * exp  
  | Sub of exp * exp  
  | Mul of exp * exp
```

### Aufgabe 15.10

Schreiben Sie ein Maschinenprogramm, das den konditionalen Ausdruck  $\text{if } 2 \cdot 3 \leq 12 - 3 \text{ then } 3 \text{ else } 5$  auswertet.

### Aufgabe 15.11

Geben Sie ein möglichst kurzes Programm  $p$  an, sodass das Programm  $(con\ k) :: p$  für positive  $k$  terminiert und für nicht positives  $k$  divergiert. Es reichen 5 Befehle.

### Aufgabe 15.12

Schreiben Sie ein Programm, das zu  $x \in \mathbb{N}$  und  $n \in \mathbb{N}$  mit einer Schleife die Potenz  $x^n$  berechnet. Schreiben Sie das Programm so wie auf Seite 332 erst in  $W$  und dann in  $M$ . Achten Sie darauf, dass Ihr Programm nur die Potenz  $x^n$  im Stapel zurücklässt.

### Aufgabe 15.13

Betrachten Sie das folgende Programm:

```
[con 2, getS 0, con 0, leq, cbranch 6, con 1, getS 0, sub, putS 0, branch ~8, halt]
```

Simulieren Sie die Ausführung des Programms auf Papier.

- Geben Sie die Folge der Adressen an, die der Programmzähler durchläuft. Tipp: Die Folge hat genau 33 Elemente.
- Skizzieren Sie das gegebene Programm durch ein Programm in  $W$ .

### Aufgabe 15.14

Schreiben Sie ein Programm in  $W$  und in  $M$ , welches zu einer gegebenen positiven Zahl  $n$  die Liste  $[n, \dots, 0]$  im Heap erstellt.

### Aufgabe 15.15

Schreiben Sie ein Maschinenprogramm, das die Länge einer Liste berechnet. Hier eine Skizze in M:

```
var xs
var n := 0
while 0<=xs do
  n := n+1 ;
  xs := !(xs + 1)
end
return n
```

Wobei  $!x$  für eine Zahl  $x$  den Wert in der Zelle mit dem Index  $x$  liefert.