Universität der Bundeswehr
Fakultät für Informatik – Institut 2
Prof. Dr. Wolfgang Hommel
Michael Grabatin, Michael Steinke

WT 2020
Übungsblatt 8

Einführung in die Informatik II

06. und 09.03.2020

1 Abstrakte Datentypen (ADT)

Es ist folgende Formalisierung für Zeichenketten gegeben:

- Relevante Mengen:

  $\mathcal{A}$: Menge aller Zeichen (das Alphabet)
  $\mathcal{S}$: Menge aller Zeichenketten

- Signatur:

$$\begin{array}{ll} \epsilon & | \rightarrow \mathcal{S} \\ append & | \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{S} \\ concat & | \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{S} \\ reverse & | \mathcal{S} \rightarrow \mathcal{S} \end{array}$$

- Axiome:

$$\begin{array}{lll} (S1) & concat(\epsilon, s) & = s \\ (S2) & concat(append(a, s_1), s_2) & = append(a, concat(s_1, s_2)) \\ (S3) & reverse(\epsilon) & = \epsilon \end{array}$$

a) Bestimmen Sie Konstruktoren und Nicht-Konstruktoren.

b) Geben Sie das Axiom $(S2)$ vollquantifiziert an.

c) Offensichtlich ist die algebraische Definition nicht vollständig. Vervollständigen Sie diese entsprechend!

d) Für Zeichenketten sollte im Allgemeinen auch folgende Regel gültig sein:

$$(R1) \quad concat(s, \epsilon) \quad = s$$

Ist diese als weiteres Axiom notwendig? Begründen Sie!

e) Die Definition soll nachträglich um eine Operation $charAt$ erweitert werden, mit der einzelne Zeichen an einem Index (mit 0 beginnend) ermittelt werden können. Führen Sie die entsprechenden Ergänzungen durch.

2 Radixsort

In der vorhergehenden Übung wurde das Sortierverfahren *Radixsort* für einen festen Typ implementiert. In dieser Aufgabe soll diese Lösung so erweitert werden, dass sie für generische Typ funktioniert.

Jeder Schlüssel ist eine Folge von m Stellen.

Zum Sortieren verwendet *Radixsort* k Fächer, je ein Fach pro möglichem Stellenwert.

Hinweis: Die Position pos wird (Scala-üblich) von 0 beginnend gezählt. Mit $pos = 2$ wird daher auf die Einerstelle einer dreistelligen Zahl (also bei $m = 3$) zugegriffen!

a) Zur Realisierung der Kellerfächer soll der folgende nicht generische Typ `Stack` in einen generischen Typ überführt werden:

```
1 //Keller
2 class StackElem(var value : Int, var next : StackElem = null)
3 class Stack(var s : StackElem = null)
4
5 //leeren Stack erstellen
6 def emptyStack : Stack = new Stack
7
8 //Pruefen ob Stack leer
9 def isEmpty(stack : Stack) : Boolean = stack.s == null
```

b) Modifizieren Sie ebenfalls die Implementierung der folgenden Prozeduren, damit diese auf dem generischen Stack funktionieren.

```
1  //Element hinzufuegen
2  def push(stack : Stack, value : Int) : Unit = {
3    stack.s = new StackElem(value, stack.s)
4  }
5
6  //Oberstes Element entfernen
7  def pop(stack : Stack) : Unit = {
8    if (stack.s == null) throw new
9      IllegalArgumentException("Stack is empty!")
10   else stack.s = stack.s.next
11 }
12
13 //Oberstes Element vom Stack liefern
14 def top(stack : Stack) : Int = {
15   if (isEmpty(stack)) throw new
16     IllegalArgumentException("Stack is empty!")
17   else return stack.s.value
18 }
```

c) In der nicht generischen Version wurde eine Prozedur `getNumPos`, welche bei gegebenen m maximalen Stellen pro Schlüssel, an der Stelle pos, den Wert von num zurückgibt, verwendet. Zum Beispiel wurde für $m = 3, pos = 2, num = 216$ das Ergebnis 6 zurückgegeben. Für einen beliebigen Typ `T` funktioniert diese Funktion nicht mehr. Sie benötigen jetzt eine Funktion, die Ihnen zu einem belibigen Datentyp eine für die Sortierung in die Keller geeignete Darstellung zurück gibt. Definieren Sie nun die Prozedur `radixsort`, welche einen Array `keys : Array[T]` nach dem *Radixsort*-Verfahren sortiert und dabei eine als (curri-sierten) Parameter übergebene Funktion `getKeyPos: (Int, T, Int)=>Int` zur Bestimmung der Kellerfächer verwendet. Die bekannte Funktion `getNumPos` soll zum Beispiel für die Sortierung von Integern als `getKeyPos` übergeben werden können.

d) Welche Werte müssen für k und m gewählt werden, wenn das Eingabearray für den Radixsort aus Zeichenketten mit Buchstaben besteht und wie muss eine Funktion `getKeyPos` für Strings aussehen? Implementieren Sie diese Funktion und testen Sie Ihre Funktion an folgendem Beispiel: `Array("alice", "bob", "eve", "trent")`