4. Arbeitsblatt

Woche 8.12.1997 und Woche 15.12.1997

Lektüre

Literatur zu abstrakten und polymorphen Datentypen.

Leitfragen

1. Welche Vorteile haben abstrakte Datentypen (ADTs)?
2. Welche Konzepte gibt es in *Hugs*, mit denen sich ADTs realisieren lassen?
3. Was ist ein polymorpher Datentyp?

Anwendungsaufgaben (die in der Gruppe entwickelt und analysiert werden sollen)

1. **Deque**
   Eine *Deque* ist eine Queue, der Elemente an beiden Enden hinzugefügt oder entnommen werden können. Definiert einen ADT für eine Deque. Überlegt Euch dazu zunächst die Signaturen von Operationen wie `emptyDQ`, `isemptyDQ`, `addDQ` und `remDQ`.

2. **Mengen**
   Der Datentyp *Menge* (`Set`) mit den Operationen `singleton`, `empty` und `union` sei wie folgt definiert:

```
type Set a = [a]
 in
  union     :: Set a-> Set a-> Set a,
  singleton :: a-> Set a,
  empty     :: Set a

empty = []
singleton x = [x]
union x y = x ++ y
```

   - Erweitert den ADT um die Operationen $\in$ (`el`) und $\subseteq$ (`subset`). Wie verändert sich dabei die Signatur?
   - Wie läßt sich die Funktion `union` optimiert realisieren, d. h. so, daß keine Elemente mehrfach in der Liste repräsentiert werden?
   - Benutzt die neuen Operationen, um Gleichheit auf Mengen zu definieren und damit den ADT `Set a` als eine Instanz der Typklasse `Eq` zu deklarieren.