

## 2. Übung „Prinzipien von Programmiersprachen“ Bearbeitung bis zum 9. November 2004

---

### Programm für die Übung am 4. November

- Besprechung letzte Übung
- abstrakte Datentypen, Beispiel Mengen, Konstruktoren

### Aufgabe 1

Nachdem wir in der letzten Übung eine Big-Step-Semantik für die einfache imperative Programmiersprache der Vorlesung definiert haben, sollen Sie in dieser Aufgabe die Äquivalenz dieser Semantik zur Small-Step-Semantik aus der Vorlesung beweisen. Hinweis: Führen Sie den Beweis analog zu dem in der Übung geführten Beweis.

### Aufgabe 2

In der Übungsstunde am 4. November werden/haben wir den abstrakten Datentyp  $\text{Set}(\alpha)$  definiert. Erweitern Sie diesen Datentyp um folgende Operationen:

```
delete   :  $\alpha$       ->  $\text{Set}(\alpha)$  ->  $\text{Set}(\alpha)$ 
union    :  $\text{Set}(\alpha)$  ->  $\text{Set}(\alpha)$  ->  $\text{Set}(\alpha)$ 
intersect :  $\text{Set}(\alpha)$  ->  $\text{Set}(\alpha)$  ->  $\text{Set}(\alpha)$ 
toList   :  $\text{Set}(\alpha)$           ->  $\text{List}(\alpha)$ 
```

Definieren Sie außerdem einen abstrakten Datentyp  $\text{Nat}$  mit folgender Schnittstelle:

```
datatype Nat
  o      :                -> Nat
  s      : Nat            -> Nat
operations
  plus   : Nat          -> Nat -> Nat
  length :  $\text{List}(\alpha)$  -> Nat
  card   :  $\text{Set}(\alpha)$  -> Nat
end
```