

## Übungen zu Informatik I

### Aufgabe 10-1

### Kassensystem

(keine Abgabe)

Der Supermarkt Ludwigshöhe will auf ein neues Kassensystem umsteigen. Sie wurden beauftragt, die dazu nötige Software zu implementieren – natürlich in SML.

Jeder Verkaufsvorgang läuft folgendermaßen ab: Die Strichcodes der verkauften Waren werden vom Kassensystem mit einem Scanner erfasst, das Kassensystem speichert Strichcodes und Stückzahl der verkauften Artikel. Nachdem alle Artikel eingescannt wurden, wird aus diesen Informationen ein Kassenzettel erstellt. Auf dem Kassenzettel wird für jeden verkauften Artikel die Artikelbezeichnung, der Einzelpreis, die verkaufte Stückzahl dieses Artikels und der sich daraus ergebende Gesamtpreis für diesen Artikel ausgedruckt. Ferner wird am Ende des Kassenzettels der Gesamtpreis für den Verkaufsvorgang ausgegeben. Um diese Funktionalität zur Verfügung stellen zu können, ist im Kassensystem eine Tabelle gespeichert, in der jedem Strichcode die Artikelbezeichnung und der Preis zugeordnet sind.

- Geben Sie eine Signatur *TABELLESig* an, die alle für die im Kassensystem gespeicherte Tabelle notwendigen Operationen beschreibt. Beachten Sie, dass Artikel neu in das Sortiment aufgenommen und aus dem Sortiment entfernt werden können und dass sich die Preise von Artikeln ändern können.
- Geben Sie eine Struktur *TABELLE* an, die die Signatur *TABELLESig* implementiert.
- Geben Sie eine Signatur *VERKAUFSSig* an, die alle für einen Verkaufsvorgang notwendigen Operationen beschreibt.
- Geben Sie eine Struktur *VERKAUF* an, die die Signatur *VERKAUFSSig* implementiert.

### Aufgabe 10-2

### Endliche Mengen

(5 Punkte)

Endliche Mengen bilden eine Rechenstruktur *SET*. (Diese Rechenstruktur wird in einer der kommenden Vorlesungen noch genauer behandelt). Eine endliche Menge enthält Elemente eines Datentyps, auf dem die Gleichheit definiert ist. Ein Element darf höchstens einmal in der Menge enthalten sein. In den folgenden Aufgaben können Sie die auf der Homepage der Vorlesung zu findende Datei *set.sml* verwenden. Diese enthält die Definition der Rechenstruktur *SET*.

- Definieren Sie eine SML-Funktion *subset* vom Typ  $"a \text{ set} * "a \text{ set} \rightarrow \text{bool}$ , die für zwei Mengen  $x$  und  $y$  vom Typ  $"a \text{ set}$  bestimmt, ob  $x$  eine Teilmenge von  $y$  ist.
- Definieren Sie eine SML-Funktion *seteq* vom Typ  $"a \text{ set} * "a \text{ set} \rightarrow \text{bool}$ , die für zwei Mengen vom Typ  $"a \text{ set}$  bestimmt, ob sie gleich sind (d.h. die gleichen Elemente enthalten).
- Definieren Sie eine SML-Funktion *intersect* vom Typ  $"a \text{ set} * "a \text{ set} \rightarrow "a \text{ set}$ , die für zwei Mengen den Durchschnitt der beiden Mengen berechnet.
- Definieren Sie eine SML-Funktion *cartprod* vom Typ  $"a \text{ set} * "b \text{ set} \rightarrow ("a * "b) \text{ set}$ , die für zwei Mengen vom Typ  $"a \text{ set}$  bzw.  $"b \text{ set}$  deren kartesisches Produkt berechnet.

Ihre Funktionen sollen hierbei *ausschließlich* die in der Signatur *SETSig* enthaltenen Funktionen zur Konstruktion bzw. Manipulation von Mengen verwenden.

**Aufgabe 10-3****Schlangen**

(7 Punkte)

Die Basisfunktionen auf (in Abschnitt 4.4 der Vorlesung kurz erwähnten) *Schlangen* (*queues*) seien wie folgt bezeichnet:

<i>emptyqueue</i> :	die leere Schlange
<i>isemptyqueue</i> :	prüft, ob eine Schlange leer ist
<i>enter</i> :	hängt („hinten“) ein neues Element an die Schlange an
<i>first</i> :	liefert das erste Element einer Schlange
<i>dequeue</i> :	entfernt das erste Element einer Schlange

- Geben Sie eine SML-Signatur *QUEUESig* an, die die Spezifikationen des Typs *'a queue* und der Basisfunktionen enthält.
- Geben Sie eine SML-Struktur *QUEUE* an, die den Typ und sämtliche Funktionen der Signatur *QUEUESig* implementiert.
- Definieren Sie eine SML-Funktion *posqueue* vom Typ **int queue**  $\rightarrow$  **int queue**, so dass ein Aufruf *posqueue*(*q*) eine Schlange zurückgibt, die alle nichtnegativen Elemente von *q* enthält. Die Elemente sollen im Ergebnis in der gleichen Reihenfolge wie in *q* angeordnet sein.
- Definieren Sie eine SML-Funktion *revqueue* vom Typ *'a queue*  $\rightarrow$  *'a queue*, so dass ein Aufruf *revqueue*(*q*) eine Schlange zurückgibt, die dieselben Elemente wie *q* in umgekehrter Reihenfolge enthält.

Die Funktionen in Teilaufgaben c) und d) sollen *ausschließlich* die in der Signatur *QUEUESig* enthaltenen Funktionen zur Konstruktion bzw. Manipulation von Schlangen verwenden.