

Übungen zu Informatik I

Aufgabe 8-1 Quantoren für Listen (keine Abgabe)

Geben Sie polymorphe SML-Funktionen *exists* und *all* mit folgenden Eigenschaften an:

- a) Die Funktion *exists* ist vom Typ $('a \rightarrow \mathbf{bool}) \rightarrow 'a \text{ list} \rightarrow \mathbf{bool}$. Sie liefert bei Eingabe einer totalen Funktion f vom Typ $'a \rightarrow \mathbf{bool}$ eine Funktion vom Typ $'a \text{ list} \rightarrow \mathbf{bool}$, die bei Eingabe einer Liste vom Typ $'a \text{ list}$ als Ergebnis *true* liefert, wenn die Liste (mindestens) ein Element b mit $f(b) = \mathbf{true}$ enthält und *false* sonst.
- b) Die Funktion *all* ist ebenfalls vom Typ $('a \rightarrow \mathbf{bool}) \rightarrow 'a \text{ list} \rightarrow \mathbf{bool}$. Sie liefert bei Eingabe einer totalen Funktion f vom Typ $'a \rightarrow \mathbf{bool}$ eine Funktion, die bei Eingabe einer Liste vom Typ $'a \text{ list}$ als Ergebnis *true* liefert, wenn für alle Elemente b der Liste $f(b) = \mathbf{true}$ gilt und *false* sonst.

Aufgabe 8-2 Kartenspiel mit Datentypen (keine Abgabe)

Sie sind fanatischer Pokerspieler und hoffen, die langen Nächte im Spielkasino mit etwas Computerhilfe besser durchstehen zu können.

- a) Als erstes gilt es, einen Datentyp *poker_karte* zu definieren. Ihre Modellierung ist hierbei von dem Gedanken geleitet, dass jede Karte eine Farbe hat (Herz, Karo, Kreuz, Pik) und entweder einen Zahlenwert (2, 3, ..., 10) oder einen Bildwert (Bube, Dame, König, Ass) besitzt. Jedes Element des Datentyps muss eine Karte repräsentieren.
- b) Beim Pokerspiel interessieren Sie sich für verschiedene Kartenkombinationen. Eine Kartenkombination sei als Liste vom Typ *poker_karte list* modelliert. Eine derartige Liste l stellt ein *Paar* dar, wenn es einen (Zahlen- oder Bild-) Wert gibt, so dass genau zwei Karten mit diesem Wert in l vorkommen. Eine Kombination ist ein *Drilling*, wenn es einen Wert gibt, der genau dreimal vorkommt. Schreiben Sie zwei SML-Funktionen *paar* und *drilling* vom Typ *poker_karte list* $\rightarrow \mathbf{bool}$, die überprüfen, ob es sich bei einer gegebenen Kartenkombination um ein Paar bzw. einen Drilling handelt.

Hinweis: Die Funktionen *all* und *exists* aus der ersten Aufgabe leisten Ihnen gute Dienste.

- c) Eine Liste l von Karten ist ein *Doppelpaar*, wenn es zwei verschiedene Werte gibt, so dass von jedem der beiden Werte genau zwei Karten in l vorkommen, und ein *Full House*, wenn es zwei verschiedene Werte gibt, von denen einer genau zweimal, der andere genau dreimal in l vorkommen. Schreiben Sie zwei SML-Funktionen *doppelpaar* und *fullhouse* vom Typ *poker_karte list* $\rightarrow \mathbf{bool}$, die entscheiden, ob es sich bei einer Kartenkombination um ein Doppelpaar bzw. ein Full House handelt.
- d) Eine Liste l von Karten ist ein *Flush*, wenn alle Karten in l die gleiche Farbe haben. Schreiben Sie eine SML-Funktion *flush* vom Typ *poker_karte list* $\rightarrow \mathbf{bool}$, die entscheidet, ob es sich bei einer Kartenkombination um einen Flush handelt.

Aufgabe 8-3**Sortierte Liste**

(2 Punkte)

Schreiben Sie ein SML-Programm *ist_sortiert*, das für eine Liste von ganzen Zahlen prüft, ob diese aufsteigend sortiert ist.

Aufgabe 8-4**Fahrzeugvermietung**

(1+1+1+2+2+3 Punkte)

Die Firma Siva vermietet Autos und Motorräder. Zur Verwaltung des Fahrzeugbestandes wird ein SML-Programm verwendet. Zu jedem Fahrzeug werden folgende Informationen gespeichert: Die Fahrgestellnummer des Fahrzeugs und ob das Fahrzeug vermietet ist oder nicht. Für jedes Auto (je nach Leistung und Ausstattung) wird zusätzlich noch eine von vier Kategorien A, B, C oder D gespeichert. Für Motorräder gibt es keine Kategorien.

- Geben Sie passende datatype-Deklarationen an.
- Geben Sie ein SML-Programm *ist_vermietet* an, das für ein Fahrzeug bestimmt, ob es vermietet ist oder nicht.
- Geben Sie SML-Programme *ein_fahrzeug_vermietet* und *alle_fahrzeuge_vermietet* an, die für eine Liste von Fahrzeugen bestimmen, ob mindestens ein bzw. alle Fahrzeuge in dieser Liste vermietet sind.
- Geben Sie SML-Programme *ein_auto* und *nur_autos* an, die für eine Liste von Fahrzeugen bestimmen, ob es sich bei mindestens einem bzw. bei allen Fahrzeugen dieser Liste um ein Auto handelt.
- Geben Sie ein SML-Programm

$$\textit{upgrade_auto} : \textit{Fahrzeug} \rightarrow \textit{Fahrzeug}$$

an, das für jedes Auto der Kategorie B ein Auto der Kategorie C mit gleicher Fahrgestellnummer und gleichem Vermietungs-Zustand als Ergebnis hat und jede andere Eingabe unverändert lässt. Geben Sie ferner ein SML-Programm

$$\textit{upgrade} : \textit{Fahrzeug list} \rightarrow \textit{Fahrzeug list}$$

an, das die Funktion *upgrade_auto* auf jedes Element der Eingabe anwendet.

- Die Tagesmieten für die einzelnen Fahrzeugtypen sind:

Fahrzeugtyp	Preis (Euro/Tag)
Auto, Kategorie A	46
Auto, Kategorie B	52
Auto, Kategorie C	64
Auto, Kategorie D	87
Motorrad	53

Schreiben Sie ein SML-Programm

$$\textit{tageseinnahmen} : \textit{Fahrzeug list} \rightarrow \textit{int}$$

das die Summe der Mieteinnahmen aller vermieteten Fahrzeuge der Eingabeliste berechnet.

Hinweis: Die Funktionen *exists* und *all* aus Aufgabe 8-1 werden Ihnen bei der Lösung dieser Aufgabe hilfreich sein.

Abgabe: Montag, 09.01.2006, 8:00 Uhr.