

Übungen zu Informatik I

Aufgabe 1 Quantoren für Listen (keine Abgabe)

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

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

Aufgabe 2 Fahrzeugvermietung (keine Abgabe)

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. Zusätzlich gehört jedes Auto (je nach Leistung und Ausstattung) zu einer der Kategorien A, B, C oder D. Für Motorräder gibt es keine Kategorien. Für Autos wird auch die Kategorie, zu der sie gehören, im Programm verwaltet.

- a) Geben Sie datatype-Deklarationen für *Kategorie* und *Fahrzeug* an.
- b) Geben Sie ein SML-Programm *ist_vermietet* an, das für ein Fahrzeug bestimmt, ob es vermietet ist oder nicht.
- c) 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.
- d) 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.
- e) Geben Sie ein SML-Programm

upgrade_auto : *Fahrzeug* \rightarrow *Fahrzeug*

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

upgrade : *Fahrzeug list* \rightarrow *Fahrzeug list*

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

f) Die Tagesmieten für die einzelnen Fahrzeugtypen sind:

Fahrzeugtyp	Preis (Euro/Tag)
Auto, Kategorie A	46,00
Auto, Kategorie B	52,00
Auto, Kategorie C	64,00
Auto, Kategorie D	87,00
Motorrad	53,00

Schreiben Sie ein SML-Programm

tageseinnahmen : *Fahrzeug list* \rightarrow *real*

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

Aufgabe 3 **Kartenspiel mit Datentypen** (2+3+5+2 Punkte)

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 Liste *l* vom Typ *poker_karte list* stellt ein *Paar* dar, wenn es einen (Zahlen- oder Bild-) Wert gibt, so dass mindestens zwei Karten mit diesem Wert in der Liste *l* vorkommen. Eine Kombination ist ein *Drilling*, wenn es einen Wert gibt, der mindestens dreimal vorkommt. Schreiben Sie zwei SML-Funktionen *paar* und *drilling* vom Typ *poker_karte list* \rightarrow **bool**, die überprüfen, ob es sich bei einer gegebenen Kartenkombination um ein Paar bzw. ein Drilling handelt.

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

c) Eine Liste *l* vom Typ *poker_karte list* ist ein *Doppelpaar*, wenn es zwei verschiedene Werte gibt, so dass von jedem der beiden Werte mindestens zwei Karten in der Liste vorkommen, und ein *Full House*, wenn es zwei verschiedene Werte gibt, von denen einer mindestens zweimal, der andere mindestens dreimal in der Liste vorkommen. Schreiben Sie zwei SML-Funktionen *doppelpaar* und *fullhouse* vom Typ *poker_karte list* \rightarrow **bool**, die entscheiden, ob es sich bei einer Kartenkombination um ein Doppelpaar bzw. ein Full House handelt.

d) Eine Liste von Karten ist ein *Flush*, wenn alle Karten der Liste die gleiche Farbe haben. Schreiben Sie eine SML-Funktion *flush* vom Typ *poker_karte list* \rightarrow **bool**, die entscheidet, ob es sich bei einer Kartenkombination um einen Flush handelt.

Bitte beachten Sie, dass wir Ihre Lösung nur dann korrigieren und bewerten können, wenn sie als <i>Textdatei</i> abgegeben wird.
--

Abgabe: Dienstag, 10.12.2002, 12:00 Uhr.