9. Übungsblatt zu Informatik 1 WS 98/99

Andreas Rossberg, Prof. Gert Smolka

Abgabe: 15. Januar 1999 vor der Vorlesung

Hinweis: Bitte alle Programme zusätzlich per Email abgeben.

Aufgabe 9.1: Typen und Werte (4 Punkte)

Geben Sie jeweils einen SML-Ausdruck an, der den folgenden Typ hat:

(a) `int list list list`

(b) `(string * bool list) list`

(c) `(int * real) * ('a list * int)`

(d) `('a -> int) * ('b -> ('c -> 'c * 'b)) list`

Aufgabe 9.2: Rekursive Prozeduren für Listen (4 Punkte)

Geben Sie für folgende Prozeduren einen geeigneten Typ an und schreiben Sie die Prozeduren in endrekursiver Form:

(a) `nth(`n`,`l`)` liefert das n-te Element der Liste l (bei 0 angefangen zu zählen).

(b) `max` l liefert das größte Element der ganzzahlige Liste l.

Aufgabe 9.3: Standardstruktur für Strings (3 Punkte)

Im Interpreter von Moscow ML können Sie mit dem Prozeduraufruf `load "String"` die Standardstruktur `String` laden (in anderen SML-Systemen steht es automatisch zur Verfügung). Danach können Sie bespielsweise die Prozedur `String.size: string -> int` benutzen, um die Länge eines Strings zu ermitteln, oder `String.sub: string * int -> char`, die ein einzelnes Zeichen aus einem String extrahiert. Über weitere Standardstrukturen und ihren Inhalt können Sie sich informieren, indem Sie auf den CIP-Rechnern das Verzeichnis `~rossberg/mosml/lib` untersuchen: jede Datei *Name*`.sig` beschreibt ein Standardmodul *Name*.

Schreiben Sie mithilfe der beiden genannten Prozeduren eine Prozedur

```
explode: string -> char list
```

die aus dem String die Liste der in dem String enthaltenen Zeichen erzeugt (in derselben Reihenfolge).

Aufgabe 9.4: Listen von Listen (3 Punkte)

Schreiben Sie eine Prozedur `flatten: 'a list list -> 'a list`, die die alle Elemente der inneren Listen zu einer Liste zusammenfasst. So soll zum Beispiel `flatten[[2,3],[4],[0,1]]` die Liste `[2,3,4,0,1]` ergeben. Ihre Prozedur soll endrekursiv sein. Tip: Benutzen Sie die vordefinierte Prozedur `rev`.

Aufgabe 9.5: Quicksort (4+3 Punkte)

(a) Schreiben Sie eine abgewandelte Quicksort-Prozedur `quicker(`xs`,`ys`)`, die im Gegensatz zu der in der Vorlesung (und in Paulson) vorgestellten auf die Benutzung des Append-Operators `@` verzichtet, indem sie das bereits sortierte Ergebnis in ys akkumuliert.

(b$\star$) Transformieren Sie die Prozedur `quick` in endrekursive Form. Gehen Sie der Einfachheit halber von der ursprünglichen Fassung aus, die `part` als getrennte Prozedur realisiert. Formulieren Sie zunächst `part` und dann `quick` selbst endrekursiv.

(c) Modifizieren sie die Quicksort-Prozedur aus der Vorlesung zu einer allgemeineren Prozedur höhrerer Ordnung

```
quick' : ('a * 'a -> bool) -> 'a list -> 'a list
```

Die übergebene Prozedur repräsentiert die zu benutzende Ordnung, und es wird eine Sortierungsprozedur geliefert, die diese Ordnung zur Partitionierung benutzt.

Aufgabe 9.6: Funktionale für Listen (5 Punkte)

Nennen Sie geeignete Typen für folgende polymorphe Prozeduren und schreiben Sie die Prozeduren:

(a) `map` f l wendet die Prozedur f auf jedes Element der Liste l an und setzt die Ergebnisse wieder zu einer Liste zusammen. Beispiel: `map ~ [1,2,3]` liefert die Liste `[~1,~2,~3]`.

(b) `exists` p l liefert `true`, wenn in der Liste l mindestens ein Element existiert, für das das Prädikat p den Wert `true` liefert (ein Prädikat ist eine Prozedur $\tau \to$ `bool`).

(c) `elem` x l testet, ob der Wert x in der Liste l enthalten ist. Verwenden Sie `exists`.

Aufgabe 9.7: Infix-Deklarationen (4 Punkte)

(a) Geben Sie geeignete Infix-Deklarationen für SML an, die den Infix-Status von Identifikatoren so deklarieren, dass er der Infix-Grammatik $G = (A, O, \mathrm{Pr}, \mathrm{As})$ mit $A = \{\texttt{+}, \texttt{x}, \texttt{y}, \texttt{z}\}$ und $O = \{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}, \texttt{E}\}$ sowie Präzedenzen und Assoziativitäten nach folgender Tabelle entspricht:

	A	B	C	D	E
Pr	7	7	5	5	3
As	r	r	l	r	l

(b) Welche der folgenden SML-Ausdrücke sind nach diesen Deklarationen syntaktisch zulässig? Geben Sie gegebenenfalls eine vollständige Klammerung an:

(i) `A + B + C`

(ii) `x E y A z B y E x`

(iii) `x A + C + x`

(iv) `x D y B z C y D x`

Aufgabe 9.8: Infix-Grammatik (4 Punkte)

Gegeben sei die Infix-Grammatik $I = (A, O, \mathrm{Pr}, \mathrm{As})$ mit $A = \{0, 1, 2, 3\}$, $O = \{<, >\}$, und den folgenden Präzedenzen und Assoziativitäten:

	$<$	$>$
Pr	0	0
As	l	r

Geben Sie eine Grammatik G mit einem Nonterminal a in BNF an, so dass für alle $w \in (A \cup O)^*$ gilt: w ist Phrase für a in G gdw. ein $b \in \mathrm{ABaum}(I)$ existiert mit $w = \mathrm{Inp}(I)$.