

## Programmierung und Modellierung

### Aufgabe 11-1

### Wiederholungen

In dieser Aufgabe wollen wir Verzeichnis-Bäume untersuchen. Ein Verzeichnisbaum ist entweder eine Datei (die einen Namen hat) oder ein Verzeichnis, das einen Namen und eine Liste von enthaltenen Verzeichnisbäumen hat. Schreiben Sie Ihre Lösungen in eine Datei `11-1.sml`, und sparen Sie nicht mit Kommentaren.

- a) Geben Sie einen rekursiven Datentyp `filesystem` für Verzeichnisbäume an.
- b) Wenn Sie wollen, probieren Sie aus, mit den in der Struktur `OS.FileSys` definierten Funktionen einen realen Verzeichnisbaum auszulesen. Beachten Sie dabei:
  - Die Dokumentation von `OS.FileSys` finden Sie unter <http://www.smlnj.org/doc/basis/pages/os-file-sys.html>. Leider ist die Dokumentation nicht ganz exakt.
  - Insbesondere liefert `OS.FileSys.readDir` keinen `string`, sondern `string option` (vgl. Foliensatz 8, Seite 26).
  - Dabei wird nur der lokale Dateiname zurückgegeben, Funktionen wie `OS.FileSys.openDir` brauchen jedoch absolute Verzeichnisnamen, deren Erstellung Sie selbst regeln müssen. Die Konkatenation von `string` ist in SML per Infix-Operator `^` möglich.
  - Das Vorgehen ist wie folgt: mit `OS.FileSys.openDir` wird ein Verzeichnis geöffnet, und ein Handle zurückgegeben. Dies kann solange mit `OS.FileSys.readDir` ausgelesen werden, bis `NONE` zurückgegeben wird. `OS.FileSys.readDir` liefert sonst die lokalen Dateinamen. Mit `OS.FileSys.isDir` kann geprüft werden, ob es sich um ein Unterverzeichnis oder eine Datei handelt.
  - Verschenden Sie nicht zuviel Zeit mit den Ungereimtheiten dieser API. Insbesondere ist diese API selbst kein Klausurstoff. Sie sollten aber in der Lage sein, mit den oben genannten Hilfen und etwas Geduld eine Lösung hinzubekommen.

- c) Schreiben Sie jetzt eine höhere Funktion

```
traverse : (filesystem -> 'a) -> ('a list -> 'a) -> filesystem -> 'a
```

die, aufgerufen mit den Parametern `f1 f2 fs`, den Verzeichnisbaum `fs` durchläuft und für jede Datei den Wert von `f1` berechnet, sowie für jedes Verzeichnis den Wert von `f2` angewendet auf die berechneten Werte der enthaltenen Verzeichnisbäume.

- d) Implementieren Sie eine Funktion `listsum : int list -> int`, die die Summe einer Liste von `int`-Werten berechnet. Achten Sie darauf, dass diese Funktion endrekursiv ist.
- e) Verwenden Sie Ihre Funktion `traverse`, um eine Funktion `dirsize` zur Berechnung der Grösse aller Dateien in einem Verzeichnisbaum zu implementieren. Verstecken Sie dabei Ihre Hilfsfunktionen geeignet. Wenn Sie ohne `OS.FileSys.fileSize` auskommen wollen, definieren Sie sich einen geeigneten Ersatz.
- f) Zeigen Sie die Terminierung Ihrer `listsum`-Funktion für alle Listen durch Angabe einer Abstiegsfunktion.

### Aufgabe 11-2

### Fixpunkte

- a) Zeigen Sie: Eine partielle Ordnung  $\triangleleft$  auf einer *endlichen* Menge  $M$  ist auch vollständig.

b) Geben Sie die denotationelle Semantik der BNF-Grammatik

$$S ::= "0" S "1" \mid \varepsilon$$

an.

**Abgabe:** Montag, den 20.7.2009, 12 Uhr, per UniWorx.