

Programmierung und Modellierung

Aufgabe 5-1

Listen

- a) Schreiben Sie eine SML-Funktion `istSortiert(l)`, die für eine Liste `l` prüft, ob diese aufsteigend sortiert ist. Dies ist genau dann der Fall, wenn `l` entweder die leere Liste ist, oder das erste Element von `l` kleiner oder gleich zum zweiten Element ist, und der Rest der Liste sortiert ist.

Hinweis: SML erlaubt, beliebige Formen von Listen zu matchen. Sie müssen dann jedoch dafür Sorge tragen, dass alle möglichen Kombinationen bedacht werden (sonst beschwert sich SML über ein „match nonexhaustive“). Alternativ können Sie eine Hilfsfunktion verwenden.

- b) Schreiben Sie eine SML-Funktion `aussortieren(l)`, die den Typ `int list -> int list * int list` hat. Diese Funktion soll eine Liste von ganzen Zahlen in zwei Listen zerlegen, so dass die erste alle negativen Elemente der Eingabeliste und die zweite alle positiven Element der Eingabeliste enthält. Zum Beispiel soll `aussortieren([1,2,~3,4,~5,~6]) = ([~3,~5,~6],[1,2,4])` gelten.

Geben Sie Ihre Lösung in einer Datei `5-1.sml` ab.

Aufgabe 5-2

Typinferenz

Geben Sie die Herleitung einer Typaussage für die folgenden SML-Ausdrücke an:

- `(1+2,3)`
- `fn x => if x then 1::nil else nil`

Geben Sie Ihre Lösung in einer Datei `5-2.tex` ab. Wenn Sie Herleitungsbäume verwenden, sollten Sie unbedingt darauf achten, einen nicht-propotionalen Font zu verwenden. Alternativ können Sie auch eine handschriftliche Abgabe einscannen und als Bild abgeben, achten Sie dabei bitte auf die Dateigröße.

Abgabe: Montag, den 8.6.2009, 12 Uhr, per UniWorx.

Am Pfingstmontag und am 2.6.2009 finden keine Übungen statt. Dieses Übungsblatt wird – zusammen einem Übungsblatt 5a, das am 3.6. online gestellt wird – in der darauffolgenden Woche besprochen. Aus technischen Gründen sind beide Übungsblätter gleichzeitig abzugeben – in UniWorx können nicht zwei Übungsblätter gleichzeitig aktiv sein.