

---

# DEPENDABLE SYSTEMS AND SOFTWARE

Fachrichtung 6.2 — Informatik

Prof. Dr.-Ing. Holger Hermanns

Alexander Graf-Brill, B.Sc.

Gilles Nies, B.Sc.



## Übungsblatt A zu Programmierung 1 (WS 13/14)

---

Lesen Sie im Buch: Kapitel 1.1 bis 1.9

<http://d.cs.uni-sb.de/index.php?575>

---

### Aufgabe A.1

Installieren Sie einen Standard ML-Interpreter und spielen Sie damit herum. Empfehlungen finden Sie unter <http://www.ps.uni-saarland.de/prog-buch/interpreter.html>.

### Aufgabe A.2

Betrachten Sie das folgende Programm:

```
val x = 7 + 4
val y = x * (x - 1)
val z = x * (y - 2)
```

Welche Bezeichner, Konstanten, Operatoren und Schlüsselwörter kommen in dem Programm vor? An welche Werte bindet das Programm die vorkommenden Bezeichner?

### Aufgabe A.3

Bezeichner, Konstanten, Operatoren und Schlüsselwörter kommen nicht nur in SML vor, sondern sind Bestandteile aller Programmiersprachen. Betrachten Sie folgenden Code-Ausschnitt in der Programmiersprache C.

```
void main() {
    int a = 5, b = 6;
    int decision;

    if (decision == 0) {
        b = a * a;
    } else {
        b = -5 + (8 - a);
    }
}
```

Welche Wörter sind Bezeichner, Konstanten, Operatoren und welche sind Schlüsselwörter? Begründen Sie Ihre Aussagen. (Achtung: Es geht hier nicht darum, eine C-Referenz oder das Internet zu befragen, sondern darum, die im ersten Kapitel gegebenen Begriffsbestimmungen zu benutzen.)

#### Aufgabe A.4

Handelt es sich bei dem Wort `int` in SML um ein Schlüsselwort, einen Bezeichner, eine Konstante oder einen Operator? Ändert sich Ihre Ansicht, wenn Sie bedenken, dass man in SML selbst neue Typen deklarieren kann?

#### Aufgabe A.5

Deklarieren Sie eine Prozedur  $p : int \rightarrow int$ , die für  $x$  das Ergebnis  $2x^2 - 2$  liefert. Identifizieren Sie das Argumentmuster, die Argumentvariable und den Rumpf Ihrer Prozedurdeklaration.

#### Aufgabe A.6

Schreiben Sie eine Prozedur  $neg : bool \rightarrow bool$ , die die Boole'sche Negation implementiert. Verwenden Sie dabei keine Operatoren.

#### Aufgabe A.7

Schreiben Sie eine Prozedur  $signum : int \rightarrow int$ , die für negative Argumente  $-1$ , für positive Argumente  $1$ , und für  $0$  das Ergebnis  $0$  liefert.

#### Aufgabe A.8

Schreiben Sie eine Prozedur  $hoch17 : int \rightarrow int$ , die zu einer Zahl  $x$  die Potenz  $x^{17}$  berechnet. Dabei sollen möglichst wenig Multiplikationen verwendet werden. Schreiben Sie die Prozedur auf zwei Arten: Mit einer Hilfsprozedur und mit lokalen Deklarationen.

#### Aufgabe A.9

- Geben Sie ein Tupel mit 3 Positionen und nur einer Komponente an.
- Geben Sie einen Tupelausdruck an, der den Typ  $int * (bool * (int * unit))$  hat.
- Geben Sie ein Paar an, dessen erste Komponente ein Paar und dessen zweite Komponente ein Tripel ist.

#### Aufgabe A.10

Wie viele verschiedene zusammengesetzte Typen können Sie aufschreiben, in denen genau vier mal der elementare Typ `int` auftaucht, und sonst keine weiteren elementaren Typen (wie z.B. `bool` oder `unit`).

### Aufgabe A.11

Schreiben Sie eine Prozedur  $min : int * int \rightarrow int$ , die zu zwei Zahlen die kleinere liefert. Deklarieren Sie  $min$  analog zu  $swap$  auf 3 verschiedene Arten: mit einem kartesischen Argumentmuster, mit einer lokalen Deklaration, und mit Projektionen.

### Aufgabe A.12

Schreiben Sie eine Prozedur  $max : int * int * int \rightarrow int$ , die zu drei Zahlen die Größte liefert, auf zwei Arten:

- Benutzen Sie drei Konditionale und keine Hilfsprozedur.
- Benutzen Sie eine Hilfsprozedur und insgesamt nur einen Konditional.

### Aufgabe A.13

Schreiben Sie eine Prozedur  $flatten : (int * int) * (bool * bool) \rightarrow int * int * bool * bool$  so, dass das SML-Programm

```
val x_1 = ...
val x_2 = ...
val b_1 = ...
val b_2 = ...
it = ((x_1,x_2,b_1,b_2) = flatten ((x_1,x_2),(b_1,b_2)))
```

für alle (typgerechten) Bindungen von  $x_1, x_2, b_1, b_2$  den Ergebnisbezeichner  $it$  an  $true$  bindet. Deklarieren Sie  $flatten$  analog zu  $swap$  auf 3 verschiedene Arten: mit einem kartesischen Argumentmuster, mit einer lokalen Deklaration, und mit Projektionen. Was sind die Vor- und Nachteile der einzelnen Deklarationen von  $flatten$ ?

### Aufgabe A.14 ((nicht nur) für Programmiermasochisten)

Versuchen Sie möglichst viele, inhaltlich verschiedene Fehlermeldungen des SML-Compilers zu erzeugen. Sie dürfen gerne die spannendsten Fehlermeldungen als Herausforderung für Ihre Kommilitonen im Forum posten. Verraten Sie dabei jedoch keinesfalls, wie Sie die Fehlermeldungen erzeugt haben.

### Aufgabe A.15 (Sortieren von Tripeln)

- Schreiben Sie eine rekursive Prozedur  $sort : int * int * int \rightarrow int * int * int$ , die ganzzahlige Tripel sortiert. Beispielsweise soll  $sort(7, 2, 5) = (2, 5, 7)$  gelten. Verwenden Sie nur 2 Konditionale.
- Schreiben Sie mithilfe von  $sort$  eine Prozedur  $max : int * int * int \rightarrow int$ , die zu drei Zahlen die Größte liefert. Verwenden Sie dabei keine kartesischen Muster.

### Aufgabe A.16

Führen Sie jede bisher erstellte Prozedur aus und geben Sie ein Ausführungsprotokoll an. Für Ihre Ausführung können Sie beliebige Argumentwerte annehmen.

### Aufgabe A.17

Gegeben sei die rekursive Prozedurdeklaration

```
fun f(n:int, a:int) : int = if n=0 then a else f(n-1, a*n)
```

- a) Geben Sie die Rekursionsfolge für den Aufruf  $f(3, 1)$  an.
- b) Geben Sie ein detailliertes Ausführungsprotokoll für den Ausdruck  $f(3, 1)$  an. Wenn es mehrere direkt ausführbare Teilausdrücke gibt, soll immer der am weitesten links stehende zuerst ausgeführt werden. Sie sollten insgesamt 18 Ausführungsschritte bekommen.