1. Übungsblatt zu Logik, Semantik und Verifikation SS 2002

Prof. Dr. Gert Smolka, Dipl.-Inform. Tim Priesnitz
www.ps.uni-sb.de/courses/prog-lsv02/

Abgabe: 15. April in der Vorlesungspause

Allgemeine Hinweise: Es wird auf jedem Übungsblatt genau 50 Punkte geben. Abgabe ist normalerweise jeweils am darauffolgenden Montag in der Vorlesungspause.

Versehen Sie Ihre Lösungen bitte mit Namen, Matrikelnummer und Gruppennummer (diese finden Sie ab Freitag auf den Webseiten)!

Programmieraufgaben (also zum Beispiel Aufgabe 1.4 auf diesem Blatt) schicken Sie bitte per Email an Ihren Bremser. Die Emailadresse finden Sie auf den Webseiten der Vorlesung.

SML: Für Programmieraufgaben benutzen wir genauso wie in Programmierung Moscow ML. Mehr Details über Moscow ML und wie Sie es benutzen können, finden Sie auf den Webseiten der Vorlesung.

Aufgabe 1.1: Unter- und Teilobjekte (5) Sei das Objekt

$$x \;=\; \langle 1,\ \{2, 3, \langle 2, 3\rangle\},\ \langle 2, \langle 3, 7\rangle\rangle\rangle$$

gegeben. Beantworten Sie die folgenden Fragen gemäß der Definitionen in Kapitel 1.

(a) Ist x finitär?

(b) Geben Sie alle Unterobjekte von x an.

(c) Geben Sie alle echten Teilobjekte von x an, die keine Unterobjekte sind.

Aufgabe 1.2: Strukturelle Induktion für Listen (12) Sei X eine Menge und seien die Menge $\mathcal{L}(X)$ und die Funktionen

$$\begin{array}{rcl} @ & \in & \mathcal{L}(X) \times \mathcal{L}(X) \to \mathcal{L}(X) \\ |\_| & \in & \mathcal{L}(X) \to \mathbb{N} \\ \mathit{rev} & \in & \mathcal{L}(X) \to \mathcal{L}(X) \end{array}$$

wie in Kapitel 2 definiert. Beweisen Sie die folgenden Aussagen:

(a) $\forall xs \in \mathcal{L}(X): \quad xs@nil = xs$

(b) $\forall xs, ys \in \mathcal{L}(X): \quad |xs@ys| = |xs| + |ys|$

(c) $\forall xs \in \mathcal{L}(X): \quad |rev(xs)| = |xs|$

(d) $\forall xs, ys \in \mathcal{L}(X): \quad rev(xs@ys) = rev(ys)@rev(xs)$

(e) $\forall xs \in \mathcal{L}(X): \quad rev(rev(xs)) = xs$

Dabei dürfen Sie annehmen, dass die Konkatenationsfunktion assoziativ ist:

$$\forall xs, ys, zs \in \mathcal{L}(X): \quad (xs@ys)@zs = xs@(ys@zs)$$

Aufgabe 1.3: Strukturelle Induktion für N (18) Sei die Menge N rekursiv definiert:

$$N \stackrel{\text{def}}{=} \{\emptyset\} \cup \{\, \{x\} \mid x \in N \,\}$$

Die Elemente von N können als Darstellungen der natürlichen Zahlen aufgefasst werden:

$$\begin{array}{ll} 0 & \emptyset \\ 1 & \{\emptyset\} \\ 2 & \{\{\emptyset\}\} \\ \ldots & \end{array}$$

Eine Additionsfunktion für N sei wie folgt definiert:

$$\begin{aligned} + &\in N \times N \to N \\ \emptyset + y &= y \\ \{x\} + y &= \{x + y\} \end{aligned}$$

(a) Definieren Sie die Menge N mithilfe von Inferenzregeln.

(b) Definieren Sie eine Multiplikationsfunktion $\cdot \in N \times N \to N$.

(c) Beweisen Sie: $\forall x, y, z \in N: \quad (x + y) + z = x + (y + z)$.

(d) Beweisen Sie: $\forall x \in N: \quad x + \emptyset = x$.

(e) Beweisen Sie: $\forall x \in N: \quad \{x\} = x + \{\emptyset\}$.

(f) Beweisen Sie: $\forall x, y \in N: \quad x + y = y + x$.

Aufgabe 1.4: Arithmetische Ausdrücke (15) Sei die in Kapitel 3 definierte Syntax für arithmetische Ausdrücke wie folgt in Standard ML implementiert:

```
type con = int
type var = int

datatype exp =
    Con of con
  | Var of var
  | Add of exp * exp
  | Mul of exp * exp
```

(a) Schreiben Sie eine Prozedur

```
show : exp -> string
```

die zu einen Ausdruck seine Darstellung mit Tupeln und Variantennummern liefert. Beispielsweise soll gelten:

```
show (Add(Con 5, Mul(Con ~3, Var 1))) = "<3,<1,5>,<4,<1,~3>,<2,1>>>"
```

Verwenden Sie die Prozedur `Int.toString` um ganze Zahlen in Zeichenreihen zu konvertieren.

(b) Schreiben Sie eine Prozedur

```
subst : exp -> exp -> var -> exp
```

die zu zwei Ausdrücken a, a' und einer Variablen x den durch Substitution erhaltenen Ausdruck $a[a'/x]$ liefert.

(c) Schreiben Sie eine Prozedur

```
eval : exp -> (var -> int) -> int
```

die zu einem Ausdruck a und einer Belegung σ die Zahl $\mathcal{D}[\![a]\!]\sigma$ liefert.