Übungsaufgaben zum SML-Kurs, WS 2000/2001

Hans Leiß

Abgabe: Di, 14.11.2000

1 Aussagenlogik

Wir wollen die Auswertung $[\![\varphi]\!]h$ einer aussagenlogischen Formel φ bei einer Belegung h (der Aussagenvariablen durch Wahrheitswerte) und einfache Normalformen von Formeln programmieren. Dazu definieren wir folgenden Datentyp:

$\langle prop \rangle \equiv$

```
datatype 'a prop =
    Atom of 'a
  | Conj of 'a prop * 'a prop
  | Disj of 'a prop * 'a prop
  | Neg of 'a prop
```

Den Typ `'a` bei atomaren Formeln lassen wir offen. Unter atomaren Formeln verstehen wir (normalerweise) Aussagenvariablen. Dann können wir `int prop` bzw. `string prop` benutzen, wenn Aussagenvariablen x_i durch ihre Nummer i oder Variablen p, q, r durch Zeichenreihen `"p"`, `"q"`, `"r"` identifiziert werden sollen. Aussagenlogische Formeln sind z.B.

$\langle expl \rangle \equiv$

```
- Conj (Atom "p", Atom "q");
val it = Conj (Atom "p",Atom "q") : string prop
- Neg (Disj (Atom 2, Atom 4));
val it = Neg (Disj (Atom 2,Atom 4)) : int prop
```

Aufgabe 1.1 Definiere eine Funktion `eval : ('a -> bool) -> 'a prop -> bool`, die eine Formel auswertet.

Das erste Argument von `eval` ist eine Belegung (engl.:*environment*) `env : 'a -> bool` der Aussagenvariablen, das zweite die Eingabeformel. Es soll also z.B. für atomare Formeln gelten: `eval env (Atom n) = env n`. Zum Testen definiere man sich eine Beispielbelegung, etwa (für `int prop`) die Belegung h, die nur bei geraden Zahlen den Wert `true` liefert:

$\langle testenv \rangle \equiv$

```
fun h n = if (n mod 2) = 0 then true else false
```

(6 Punkte)

Aufgabe 1.2 Definiere eine Funktion `nnf : 'a prop -> 'a prop`, die die *Negationsnormalform* einer Formel berechnet, d.h. eine äquivalente Formel, bei der `Neg` nur vor atomaren Teilformeln steht. Die Negationsnormalform $nnf(\varphi)$ von φ ist definiert durch die Fälle

$$
\begin{array}{lcllcl}
nnf(\varphi) & := & \varphi, \quad \text{falls } \varphi \text{ atomar ist,} & & & \\
nnf(\neg\varphi) & := & \neg\varphi, \quad \text{falls } \varphi \text{ atomar ist,} & nnf(\neg\neg\varphi) & := & \varphi, \quad \text{falls } \varphi \text{ atomar ist,} \\
nnf(\varphi \wedge \psi) & := & (nnf(\varphi) \wedge nnf(\psi)), & nnf(\neg(\varphi \wedge \psi)) & := & (nnf(\neg\varphi) \vee nnf(\neg\psi)), \\
nnf(\varphi \vee \psi) & := & (nnf(\varphi) \vee nnf(\psi)), & nnf(\neg(\varphi \vee \psi)) & := & (nnf(\neg\varphi) \wedge nnf(\neg\psi)),
\end{array}
$$

(4 Punkte)