Verfahren zur automatischen Verifikation

Prof. Dr. Christoph Walther • Markus Aderhold
FG Programmiermethodik • FB Informatik • TU Darmstadt
Sommersemester 2009

3. Übungsblatt

Übung am Dienstag, 19.05.2009

Hinweis Alle Verweise in den Aufgaben beziehen sich auf das Skriptum.

Aufgabe 3.1 (Relationenbeschreibungen von Prozeduren)
Gegeben sei die Prozedur

```
function value(x:IF.Expr, alist:association.list):bool <=
if x=IF(test(x),left(x),right(x))
  then if value(test(x),alist)
         then value(left(x),alist)
         else value(right(x),alist)
       fi
  else assignment(x,alist)
 fi .
```

(i) Bestimmen Sie die Relationenbeschreibung $B_{\texttt{value}}$ gemäß Definition 4.18.

(ii) Bestimmen Sie alle range-Generalisierungen von $B_{\texttt{value}}$ gemäß Definition 4.21. Welche dieser range-Generalisierungen sind range maximal?

(iii) Bestimmen Sie alle domain-Generalisierungen der Relationenbeschreibungen aus Teilaufgabe (ii) gemäß Definition 4.23. Welche dieser domain-Generalisierungen sind domain maximal und welche sind maximal?

(iv) Bestimmen Sie alle optimalen Relationenbeschreibungen der Relationenbeschreibungen aus Teilaufgabe (iii) gemäß Definition 4.25.

(v) Bestimmen Sie ein optimales *grd-set*(`value`) (vgl. Skript Seite 38).

Aufgabe 3.2 (Relationenbeschreibungen von Prozeduren)
Gegeben sei die Prozedur

```
function mult(x,y:nat):nat <=
if x=0
  then 0
  else if even(x)
         then mult(half(x),plus(y,y))
         else plus(y,mult(half(x),plus(y,y)))
       fi
 fi ,
```

wobei die Prozedur `even` entscheidet, ob das Argument des Prozeduraufrufs eine gerade Zahl ist, und `half(x)` die nach unten gerundete Division von `x` durch 2 berechnet.

(i) Bestimmen Sie die Relationenbeschreibung $B_{\texttt{mult}}$ gemäß Definition 4.18.

(ii) Bestimmen Sie alle range-Generalisierungen von $B_{\texttt{mult}}$ gemäß Definition 4.21. Welche dieser range-Generalisierungen sind range maximal?

(iii) Bestimmen Sie alle domain-Generalisierungen der Relationenbeschreibungen aus Teilaufgabe (ii) gemäß Definition 4.23. Welche dieser domain-Generalisierungen sind domain maximal und welche sind maximal?

(iv) Bestimmen Sie alle optimalen Relationenbeschreibungen der Relationenbeschreibungen aus Teilaufgabe (iii) gemäß Definition 4.25.

(v) Bestimmen Sie *grd-set*(`mult`), so daß diese Menge optimal ist (vgl. Skript Seite 38).

Aufgabe 3.3 (Subsumption von Relationenbeschreibungen)

Vergleichen Sie alle Relationenbeschreibungen aus *grd-set*(`mult`), *grd-set*(`N-half`) und aus $\{B_{\texttt{nat}}, B_{\texttt{plus}}\}$ bzgl. Subsumption $\sqsubseteq$ (vgl. Definition 4.27) miteinander, wobei Variablennamen keine Rolle spielen (d. h. Variablen dürfen konsistent umbenannt werden).

Aufgabe 3.4 (Aussagenlogik, Symbolische Auswertung)

(i) Formulieren Sie die aussagenlogische Formel $(a \to b) \to ((\neg a \to b) \to b)$ als booleschen Term t und werten Sie t mit den Auswertungsregeln aus Abschnitt 5.3.1 symbolisch aus. Geben Sie Ihre symbolische Auswertung so wie im Beispiel am Ende von Abschnitt 5.3.1 an. Was schließen Sie über die Formel aus dem Ergebnis der symbolischen Auswertung?

(ii) Formulieren Sie die aussagenlogische Formel $\neg(\neg b \to a) \to \neg(a \to b)$ als booleschen Term t und werten Sie t mit den Auswertungsregeln aus Abschnitt 5.3.1 symbolisch aus. Geben Sie Ihre symbolische Auswertung so wie im Beispiel am Ende von Abschnitt 5.3.1 an. Was schließen Sie über die Formel aus dem Ergebnis der symbolischen Auswertung?

Aufgabe 3.5 (Markierung von Prozedurrümpfen)

(i) Gegeben sei die Prozedur

```
function bsort(k:list):list <=
if k=empty
  then empty
  else add(last(bubble(k)),bsort(but_last(bubble(k))))
 fi
```

mit *grd-set*$(bsort) = \{\{\langle k \neq empty, \{\{k/but\_last(bubble(k))\}\}\rangle\}\}$. Markieren Sie den Prozedurrumpf von *bsort* so, wie in Definition 5.1 angegeben. (Dazu können Sie dieses Blatt kopieren und Ihre Markierungen darauf – nicht in schwarz oder blau – eintragen).

(ii) Gegeben sei die Prozedur

```
function ssort(k:list):list <=
if k=empty
  then empty
  else add(minimum(k),ssort(replace(minimum(k),hd(k),tl(k))))
 fi
```

mit *grd-set*$(ssort) = \{\{\langle k \neq empty, \{\{k/replace(minimum(k), hd(k), tl(k))\}\}\rangle\}\}$. Markieren Sie den Prozedurrumpf von *ssort* so, wie in Definition 5.1 angegeben.