Verfahren zur automatischen Verifikation

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

4. Übungsblatt

Übung am Dienstag, 26.05.2009

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

Aufgabe 4.1 (Execution Guard)

Gegeben sei die Prozedur

```
function normalized(x:IF.Expr):bool <=
if x=IF(test(x),left(x),right(x))
  then if test(x)=IF(test(test(x)),left(test(x)),right(test(x)))
         then false
         else if normalized(left(x))
                then normalized(right(x))
                else false
              fi
       fi
  else true
fi .
```

Berechnen Sie den Execution Guard $exec_{\texttt{normalized}}$ wie in Definition 5.2 angegeben.

Aufgabe 4.2 (Symbolische Auswertung von Induktionsformeln)

Gegeben sei ein funktionales Program Pr mit den Prozeduren `even` und `dbl` aus Abschnitt 11, sowie das Lemma `lemma even_dbl <= all x:nat even(dbl(x))`.

(i) Bestimmen Sie die optimalen Mengen für *grd-set*$(even)$ und *grd-set*(dbl).

(ii) Wählen Sie eine Relationenbeschreibung $B \in$ *grd-set*$(even) \cup$ *grd-set*(dbl) und bilden Sie $\mathcal{INDUCTIVE}(B, goal[x])$ mit $goal[x] = even(dbl(x))$ gemäß Definition 4.13.

(iii) Geben Sie $goal_{\perp_{\langle Pr,H,\varnothing,\{\otimes\},\varnothing\rangle}}$ durch Angabe der symbolischen Auswertung von *goal* unter den Kontrollparametern $\langle Pr, H, \varnothing, \{\otimes\}, \varnothing\rangle$ für jede Sequenz $\langle H, IH, goal\rangle \in \mathcal{INDUCTIVE}(B, goal[x])$ an. Notieren Sie dabei wie im Skript, welche Auswertungsregeln Sie bei jedem Schritt verwendet haben, und insbesondere, welche der Anwendungsbedingungen von Regel 5.19 *Execute procedure call* bei Verwendung dieser Auswertungsregel jeweils erfüllt ist.

(iv) Vergleichen Sie $goal_{\perp_{\langle Pr,H,\varnothing,\{\otimes\},\varnothing\rangle}}$ für jede Sequenz $\langle H, IH, goal\rangle \in \mathcal{INDUCTIVE}(B, goal[x])$, für die $IH \neq \varnothing$ gilt, mit den Elementen von IH und interpretieren Sie das Ergebnis Ihres Vergleichs.

Aufgabe 4.3 *(siehe Folgeseite)*

Aufgabe 4.3 (Assumption-Menge, Unfold- und Assumptionregeln)

Gegeben seien die Prozeduren

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

```
function bubble(k:list):list <=
if k=empty
  then empty
  else if tl(k)=empty
        then k
        else if hd(k)>hd(tl(k))
               then add(hd(k),bubble(tl(k)))
               else add(hd(tl(k)),bubble(add(hd(k),tl(tl(k)))))
             fi
       fi
fi
```

```
function ordered(k:list):bool <=
if k=empty
  then true
  else if tl(k)=empty
         then true
         else if hd(k)>hd(tl(k))
                then false
                else ordered(tl(k))
              fi
       fi
fi
```

sowie die Prozeduren `last` aus dem Skriptum und `bsort` von Übungsblatt 3. Des weiteren ist folgendes verifizierte Lemma

```
lemma last_bubble_le_2nd_last_bubble <= all k:list
 if(k=empty,
    true,
    if(tl(k)=empty,
       true,
       if(last(bubble(k))>last(bubble(but_last(bubble(k)))),false,true)))
```

gegeben. Zu beweisen ist das Lemma

```
lemma bsort_sorts <= all k:list ordered(bsort(k))
```

durch Induktion über die Relationenbeschreibung von `bsort`.

(i) Berechnen Sie für den Schrittfall des Beweises die verfügbare Assumption-Menge wie im Skript angegeben.

(ii) Beweisen Sie den Schrittfall durch symbolische Auswertung und notieren Sie dabei die Anwendung von Execute-, Unfold- und Assumptionregeln. Folgen von einfachen Beweisschritten dürfen Sie dabei abkürzen.