Verfahren zur automatischen Verifikation

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

5. Hausaufgabe

Abgabe am Dienstag, den 23.06.2009, in der Übung

Hinweise

Alle Verweise in den Aufgaben beziehen sich auf das Skriptum. Für diese Hausaufgabe gibt es **18 Punkte**.

Bei Auswertungen gehen Sie bitte davon aus, dass in allen Anwendungstests für das Öffnen einer Prozedur keine Assumption-, Execute- oder Unfoldregeln verwendet werden können, die entsprechenden Limits also 0 sind, bzw. die Markierung $\otimes$. Wesentliche Beweisschritte sind das Öffnen von Prozeduren und das Anwenden von Assumptions. Diese Schritte dürfen Sie auf keinen Fall weglassen. Die globalen Parameter für *unfold limit* und *search limit* wählen Sie bitte wie im Skript angegeben. Sollte keine weitere Regel des Kalküls anwendbar sein, setzen Sie alle Limits wieder auf den Ausgangswert zurück und starten Sie die Auswertung erneut (*Reset*). Sie dürfen zur Erstellung Ihrer Lösung elektronische Hilfsmittel wie Texteditoren verwenden. Alle Ableitungen, die in dieser Übung berechnet werden, sollten nicht länger als ca. 10 Schritte sein (unter Verwendung von $\vdash^*$). Wenn Sie den Anwendungstest einer Regel komplett angeben, können Sie sich in der Auswertung auf diese Teilableitung berufen, und so abkürzen.

Hausaufgabe 5.1 (Assumption-Menge, Unfold- und Assumptionregeln) **6 Punkte**

Gegeben seien die Prozeduren `ssort` von Übungsblatt 3 und `ordered` von Übungsblatt 4 sowie das verifizierte Lemma

```
lemma le_minimum_replace <= all k:list
 if(k=empty,
    true,
    if(minimum(k)>minimum(replace(minimum(k),hd(k),tl(k))),
       tl(k)=empty,
       true))
```

Zu beweisen ist das Lemma

```
lemma ssort_sorts <= all k:list ordered(ssort(k))
```

durch Induktion über die Relationenbeschreibung von `ssort`.

(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 Unfold- und Assumptionregeln. Folgen von einfachen Beweisschritten dürfen Sie dabei abkürzen.

Fortsetzung siehe nächste Seite

Hausaufgabe 5.2 (Assumption-Menge, Assumptionregeln) **12 Punkte**

Gegeben seien die Prozedur

```
function ack(x:nat, y:nat):nat <=
if x=0
  then succ(y)
  else if y=0
         then ack(pred(x),1)
         else ack(pred(x),ack(x,pred(y)))
       fi
fi
```

sowie die Prozedur > aus dem Skriptum. Des weiteren sind folgende verifizierte Lemmata

```
lemma >_asymmetric <= all y,x:nat if(x>y,if(y>x,false,true),true)
lemma >_irreflexive* <= all y,x:nat if(x=y,if(x>y,false,true),true)
lemma >_neg_transitive <= all z,x,y:nat if(x>z,if(x>y,true,y>z),true)
lemma >_pred_estimation <= all y,x:nat
 if(x>y,x>pred(y),if(x>pred(y),x=y,true))
lemma pred_weak_>_monotone <= all y,x:nat if(pred(x)>pred(y),x>y,true)
lemma ack_gt_y <= all y,x:nat ack(x,y)>y
lemma ack_not_0 <= all x,y:nat if(ack(x,y)=0,false,true)
lemma ack_x_1_gt_x <= all x:nat ack(x,1)>x
```

gegeben. Zu beweisen ist das Lemma

```
lemma ack_ge_x <= all y,x:nat if(x>ack(x,y),false,true)
```

durch Induktion über die Relationenbeschreibung

$$B_{ack} = \{\langle\{x \neq 0, y = 0\}, \{\{x/pred(x), y/1\}\}\rangle, \langle\{x \neq 0, y \neq 0\}, \{\{x/pred(x)\}, \{x/x, y/pred(y)\}\}\rangle\}$$

von `ack`.

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

(ii) Beweisen Sie alle Induktionsformeln durch symbolische Auswertung und notieren Sie dabei die Anwendung der Assumptionregeln. Folgen von einfachen Beweisschritten dürfen Sie dabei abkürzen.