Ein Beispielbeweis zur Verifikation des Programms zur Division mit Rest auf der Basis der Hoare-Regeln

Ralf Lämmel

24. Dezember 2014

1 Hoare-Regeln

Im folgenden müssen wie folgende Hoare-Regeln benutzen:

Axiom der Zuweisung

$$\{\ P[x/t]\ \}\ x = t;\ \{\ P\ \}$$

Also wird die Vorbedingung aus der Nachbedingung erhalten, indem man die Variable der linken Seite der Zuweisung durch den Ausdruck der rechten Seite ersetzt. Man rechnet also die Vorbedingung aus der Nachbedingung aus.

Regel der Anweisungssequenz

$$\frac{\{\ P\ \}\ S_1\ \{\ R\ \}, \{\ R\ \}\ S_2\ \{\ Q\ \}}{\{\ P\ \}\ S_1;\ S_2\ \{\ Q\ \}}$$

Also wird ein Beweis einer Anweisungssequenz in zwei Teile zerlegt so dass man eine 'Zwischenbedingung' R zu finden, welche zur Nachbedingung der ersten Anweisung und zur Vorbedingung der zweiten Anweisung wird. Die Idee ist wieder, dass man R von hinten kommend aus dem Beweis für die zweite Anweisung ermittelt.

Regel der While-Schleife

$$\frac{\{\ I \&\& B\ \}\ S\ \{\ I\ \}}{\{\ I\ \}\ \textbf{while}\ (B)\ S\ \{\ I \&\& !B\ \}}$$

Also wird ein Beweis zu einer Schleife vereinfacht zu einem Beweis über dem Schleifenkörper. Allerdings muss eine Invariante I ausgemacht werden, welche den wesentlichen Anteil an einer typischerweise gegebenen Nachbedingung hat. Die Regel modelliert, dass die Invariante vor und nach der Schleife sowie am Anfang und am Ende des Schleifenkörpers gelten soll. Die Regel ist 'offensichtlich korrekt', da eine Schleifenausführung auf eine gewisse Anzahl von Ausführungen des Schleifenkörpers reduziert werden kann. Wenn also I jeweils vor und nach dem Schleifenkörper gilt, dann gilt I auch vor und nach der Schleife insgesamt. Hinsichtlich der Bedingung B ist ausserdem bekannt, dass die Bedingung ganz sicher nicht mehr nach der Schleifenausführung gilt und dass sie ganz sicher gilt beim Eintritt in den Schleifenkörper; siehe das entsprechende positive bzw. negative Vorkommen von B in der Regel.

Regel der Anpassung der Vorbedingung

$$\frac{P \Rightarrow R, \{\ R\ \}\ S\ \{\ Q\ \}}{\{\ P\}\ S\ \{\ Q\ \}}$$

Um einen Beweis $\{\ P\}\ S\ \{\ Q\ \}$ zu führen, reicht es also auch einen Beweis mit einer schwächeren Vorbindung R zu führen. Die Intuition ist hier, dass wir einer Vorbedingung immer verstärken können, ohne Korrektheit zu kompromitieren, denn die Vorbedingung schrenkt nur ein, unter welcher Voraussetzung wir die Korrektheit im Sinne der Nachbedingung annehmen.

2 Beispielbeweis

Gegeben ist das folgende Programm:

```
q = 0;
r = x;
while (r >= y) {
    r = r - y;
    q = q + 1;
}
```

Wir möchten zum Ausdruck bringen, dass dieses Programm ‘Division mit Rest’ implementiert. Dazu statten wir das Programm mit einer Vor- und einer Nachbedingung aus:

```
{ x >= 0 && y > 0 }
q = 0;
r = x;
while (r >= y) {
    r = r - y;
    q = q + 1;
}
{ x == q*y + r && r < y && q >= 0 && r >= 0 }
```

Die Nachbedingung modelliert, dass die Variable `q` den Quotient der Division und die Variable `r` den Rest enthalten sollen. Die Vorbedingung schränkt die Argumente auf nichtnegative Werte ein. Die Variable `y` soll auch ungleich 0 sein, um damit den Fall der Division durch 0 aus der Betrachtung auszuschließen. (Wir könnten auch negative Werte betrachten, aber dies soll hier der Einfachheit halber nicht geschehen.)

Um den großen Beweis zu führen, müssen wir einige kleinere Schritte gemäß der Programmstruktur führen. Dazu führen wir zur Klarheit einmal Zusicherungen in Ergänzung von Vor- und Nachbedingung an allen möglichen Stellen des Programmes ein.

```
{ Z1: x >= 0 && y > 0 }
q = 0;
{ Z2: ? }
r = x;
{ Z3: ? }
while (r >= y) {
    { Z4: ? }
    r = r - y;
    { Z5: ? }
    q = q + 1;
    { Z6: ? }
}
{ Z7: x == q*y + r && r < y && q >= 0 && r >= 0 }
```

Hier sind $Z1$ die Vorbedingung und $Z7$ die Nachbedingung von eben. Die anderen Zusicherungen sind offensichtlich noch unbekannt und müssen im Laufe des weiteren Beweises ermittelt werden.

Wir beginnen den Beweis (immer) von hinten. Dies ist praktikabel, da uns typischerweise 'starke' Nachbedingungen vorliegen, aus denen wir 'schwächste' Vorbedingungen ausrechnen. Dazu ist insbesondere zu beachten, wie das Axiom der Zuweisung das Ausrechnen einer Vorbedingung aus einer gegebenen Nachbedingung ermöglicht.

Wenn wir also von hinten beginnen, dann ist die letzte Anweisung die While-Schleife. Wir müssen die Regel der While-Schleife für das gegebene Programm instantiieren. Gesucht ist somit eine Invariante I. Die Idee ist, dass I zusammen mit der Negation der Bedingung aus der Schleife der bereits bekannten Nachbedingung ($Z7$) entsprechen soll. Diese Intuition funktioniert für das gegebene Beispiel direkt ohne nennenswerte Anpassungen. Dies ist die Invariante:

```
x == q*y + r && q >= 0 && r >= 0
```

Die negierte Bedingung der Schleife ist dies:

```
!(r >= y)
```

Nach Umformulierung:

```
r < y
```

Nach Kombination mit der Invariante (einschliesslich der Umsortierung der Operanden der Konjunktion) erhalten wir genau $Z7$:

```
x == q*y + r && r < y && q >= 0 && r >= 0
```

Nach der Regel der While-Schleife ist nun folgender Beweis zu führen; siehe dazu die Prämisse der Regel:

```
{ I && B } // Z4
r = r - y;
{ Z5 }
q = q + 1;
{ I } // Z6
```

Wir setzen I und B ein:

```
{ x == q*y + r && q >= 0 && r >= 0 && r >= y }
r = r - y;
q = q + 1;
{ x == q*y + r && q >= 0 && r >= 0 }
```

Wir wenden die Regel der Anweisungssequenz an.
Dazu ersetzen wir `q` in I durch `q + 1` und erhalten:

```
x == (q+1)*y + r && (q+1) >= 0 && r >= 0
```

Nun ersetzen wir `r` in dem Zwischenergebnis durch `r - y` und erhalten:

```
x == (q+1)*y + (r-y) && (q+1) >= 0 && (r-y) >= 0
```

Wir multiplizieren aus:

```
x == q*y + y + (r-y) && (q+1) >= 0 && (r-y) >= 0
```

Wir gleichen y aus:

```
x == q*y + r && (q+1) >= 0 && (r-y) >= 0
```

Wir bringen y auf die andere Seite:

```
x == q*y + r && (q+1) >= 0 && r >= y
```

Bisher haben wir also dies bewiesen:

```
{ x == q*y + r && (q+1) >= 0 && r >= y }
r = r - y;
q = q + 1;
{ x == q*y + r && q >= 0 && r >= 0 }
```

Es gibt eine Abweichung bei der Vorbedingung.

'Haben':

```
x == q*y + r && (q+1) >= 0 && r >= y
```

‘Soll’ (siehe Beginn des Beweises zum Schleifenkörper):

```
x == q*y + r && q >= 0 && r >= 0 && r >= y
```

Wir entfernen die übereinstimmenden Glieder.

‘Haben’ (Rest):

```
(q+1) >= 0
```

‘Soll’ (Rest):

```
q >= 0 && r >= 0
```

Es ist leicht zu sehen, dass ‘Soll’ stärker ist als ‘Haben’. Insbesondere gilt, dass wenn eine Zahl grösser oder gleich als 0 ist, dann ist jeder grössere Zahl auch grösser oder gleich als 0.

Damit ist also die Regel der Anpassung der Vorbedingung anwendbar. Damit ergibt sich aus dem geführten Beweis zu ‘Haben’ als Vorbedingung zusammen mit der Implikation ‘Soll’ $\Rightarrow$ ‘Haben’ entsprechend der beiden Prämissen der Regel die Schlussfolgerung—also der Beweis mit ‘Soll’ als Vorbedingung. Dies ist also bewiesen:

```
{ x == q*y + r && q >= 0 && r >= 0 && r >= y }
r = r - y;
q = q + 1;
{ x == q*y + r && q >= 0 && r >= 0 }
```

Daraus folgt nach der Regel der While-Schleife, dass wir auch dies bewiesen haben:

```
{ x == q*y + r && q >= 0 && r >= 0 }
while (r >= y) {
    r = r - y;
    q = q + 1;
}
{ x == q*y + r && r < y && q >= 0 && r >= 0 }
```

Damit bleibt nur noch dies zu zeigen:

```
{ x >= 0 && y > 0 }
q = 0;
r = x;
{ x == q*y + r && q >= 0 && r >= 0 }
```

Entsprechend der Regel zur Anweisungssequenz und dem Axiom der Zuweisung rechnen wir die Vorbedingung für die hintere Zuweisung aus:

```
x == q*y + x && q >= 0 && x >= 0
```

Wir rechnen die Vorbedingung für die vordere Zuweisung aus:

```
x == 0*y + x && 0 >= 0 && x >= 0
```

Nach trivialer Vereinfachung:

```
x >= 0
```

Gegeben war allerdings diese Vorbedingung:

```
x >= 0 && y > 0
```

Gemäß der Regel der Anpassung der Vorbedingung gilt, dass aus der gegebenen Vorbedingung (‘Soll’) die aktuell errechnete Vorbindung (‘’Haben’) folgt und damit ist auch der gewünschte Gesamtbeweis vollendet.

q.e.d.

Es sei noch betont, dass wir lediglich die schwächste Vorbedingung ausgerechnet haben, für welche eben noch partielle Korrektheit im Sinne des Erfüllens der Nachbedingung $Z7$ gegeben ist. Es steht uns frei mehr Forderungen mittels der Vorbedingung zu stellen. Insbesondere haben wir mittels $Z1$ gefordert, dass y nicht 0 sein soll, da wir wissen, dass sonst eine Division mit 0 vorliegt, welche über eine Nichttermination ‘beobachtbar’ wäre. Der Beweis der partiellen Korrektheit bleibt aber davon unberührt, da er keinen Terminationsbeweis beinhaltet.