Alp2 - Übungsblatt 5 (Aufgabe 2)

Bearbeitet von: Jasmine Cavael & Alexander Chmielus
Tutor: Fabian Halama
Tutorium 10 (Do. 16-18)

Mit diesem Schreibprogramm ist es leider etwas umständlich, mathematische Zeichen einzufügen in den Formeln. Wir werden daher Python-Syntax verwenden:
&& als logisches UND
|| als logisches ODER
! als logische Negation
--> Implikation
== ist gleich (wir benutzen = als Zuweisung, daher der Unterschied.)
!= ist nicht gleich

```
{P} = {x >= 0 && (x * y + z) == c}
if x % 2 == 0:
        y = y + y
        x = x // 2
else:
        z = z + y
        x = x - y
{Q} = { x >= 0 && (x * y + z) == c}
```

Bedingungsregel:

Fall 1) x % 2 == 0:
{P && x % 2 == 0} = {x >= 0 && (x * y + z) == c && (x % 2) == 0}
y = y + y
x = x // 2
{Q} = {x >= 0 && (x * y + z) == c} muss gültig sein.

Zuweisungsaxiom (x = x // 2):
{Q1} = {(x // 2) >= 0 && ((x // 2)*y + z) == c}

Zuweisungsaxiom (y = y + y):
{Q2} = {(x // 2) >= 0 && ((x // 2)*(y + y) + z) == c}

{Q1} und {Q2} sind nach dem Zuweisungsaxiom gültige Programmformeln. Für die **Konsequenzregel (der stärkeren Vorbedingung)** müssen wir nun zeigen, dass {P && (x % 2 == 0)} --> {Q2}. Zuerst zeigen wir, dass **{P && (x % 2 == 0)} --> (x // 2) >= 0:**
(x >= 0 && (x % 2 == 0) --> (x //2 >= 0) ist offensichtlich. x ist positiv,oder 0. Demnach ist (x // 2)

auch positiv, oder Null.

Nun noch für die zweite Bedingung, **{P && (x % 2 == 0)} --> ((x // 2)*(y + y) + z) == c):**
Wir wissen, dass (x % 2 == 0), also dass durch 2 teilbar ist. Demnach können wir (x //2) durch (x / 2) ersetzen. Dadurch erhalten wir:
((x / 2) * (2 * y) + z == c) = (((x * 2 * y) / 2) + z == c) = ((x * y + z) == c). Dieser Teil kommt sowohl in {Q2} als auch in {P && (x % 2 == 0)} vor, also passt das.
((x % 2 == 0) && ((x * y + z) == c)) --> (((x // 2)*(y + y) + z) == c))

Damit erhalten wir: {P && (x % 2 == 0)} --> {Q2}

Der erste Teil des Programms sieht also so aus:
{P && (x % 2 == 0)}
{Q2}
y = y + y
{Q1}
x = x // 2
{Q}
, wobei {Q2] S {Q1} durch die **Konsequenzregel** gültig ist und die gesamte Formel nach der **Sequenzregel**.

Fall 2) x % 2 != 0:
{P && x % 2 == 0} = {x >= 0 && (x * y + z) == c && (x % 2) != 0}
z = z + y
x = x - 1
{Q} = {x >= 0 && (x * y + z) == c} muss gültig sein.

Zuweisungsaxiom (x = x - 1):
{Q1} = {(x - 1) >= 0 && ((x - 1) * y + z) == c

Zuweisungsaxiom (z = z + y):
{Q2} = {(x - 1) >= 0 && ((x - 1) * y + (z + y)) == c}

{Q1} und {Q2} sind nach dem Zuweisungsaxiom gültige Programmformeln. Für die **Konsequenzregel (der stärkeren Vorbedingung)** müssen wir nun zeigen, dass {P && (x % 2 != 0)} --> {Q2}. Zuerst zeigen wir, dass **{P && (x % 2 != 0)} --> (x -1) >= 0:**
Da (x % 2) != 0 wissen wir, dass x != 0. Und weil zusätzlich (x >= 0) gilt, ist (x - 1) >= 0.
((x >= 0) && (x %2 != 0)) --> (x - 1) >= 0

Jetzt zeigen wir, dass **{P && (x % 2 != 0)} --> ((x - 1) * y + (z + y)) == c:**
Dazu stellen wir die Formel um:
((x - 1) * y + (z + y) == c) = ((x * y - y + z + y) == c) = (x * y + z == c)

Dieser Teil kommt sowohl in {Q2} als auch in {P && (x % 2 == 0)} vor, also passt das.
<u>((x * y + z) == c)) --> ((x - 1) * y + (z + y) == c)</u>

Damit erhalten wir: <u>{P && (x % 2 != 0)} --> {Q2}.</u>

Der zweite Teil des Programms sieht also so aus:
{P && (x % 2 == 0)}
{Q2}
y = y + y
{Q1}
x = x // 2
{Q}
, wobei {Q2] S {Q1} durch die **Konsequenzregel** gültig ist und die gesamte Formel nach der **Sequenzregel.**

<u>**Beide Fälle der Bedingungsregel sind gültig, wodurch die gesamte Programmformel gültig ist.**</u>