Programmierung 2 (SS 2021)
Universität des Saarlandes
Fakultät MI
Compiler Design Lab

Musterlösung 14
WP & Korrektheit

Prof. Dr. Sebastian Hack
Julian Rosemann, B. Sc.
Thorsten Klößner, M. Sc.
Julia Wichlacz, M. Sc.
Maximilian Fickert, M. Sc.

Im Vorlesungskalender finden Sie Informationen über die Kapitel des Skripts, die parallel zur Vorlesung bearbeitet werden sollen bzw. dort besprochen werden. Die Übungsaufgaben dienen der Vertiefung des Wissens, das in der Vorlesung vermittelt wird und als Vorbereitung auf Minitests und Klausur.
Weitere Aufgaben zu den Themen finden Sie jeweils am Ende der Skriptkapitel.
Die Schwierigkeitsgrade sind durch Steine des 2048-Spiels gekennzeichnet, von 512 „leicht“ bis 2048 „schwer“. 4096 steht für Knobelaufgaben.

Aufgabe 14.0: Schwächste Vorbedingung

Berechnen Sie

1. $\text{wp}(x = x - 1; \mid x = 5)$
2. $\text{wp}(x = 7; \mid x = 5)$
3. $\text{wp}(x = 7; \mid x = 7)$
4. $\text{wp}(a = a - b; \mid a > b)$

Lösung

1. $\text{wp}(x = x - 1; \mid x = 5)$
   $= (x - 1 = 5)$
   $\Leftrightarrow x = 6$
2. $\text{wp}(x = 7; \mid x = 5)$
   $= (7 = 5)$
   $\Leftrightarrow$ false
3. $\text{wp}(x = 7; \mid x = 7)$
   $= (7 = 7)$
   $\Leftrightarrow$ true
4. $\text{wp}(a = a - b; \mid a > b)$
   $= (a - b > b)$
   $\Leftrightarrow a > 2b$

Aufgabe 14.1: Schwächste Vorbedingung

Berechnen Sie mit Hilfe des *wp*-Kalküls:

1. $wp($`a = a + 2;` $\mid$ `a = 42`$)$
2. $wp($`a = b * c; b = c + a; c = c - 17;` $\mid$ `a = b` $\wedge$ `b = c + 59`$)$
3. $wp($`if (a == 0) abort(); else x = x + 1;` $\mid$ `x > 0`$)$
4. $wp($`if (x < 0) y = -x; else y = x;` $\mid$ `y > 0` $\wedge$ `X * X = y * y`$)$
5. $wp($`x = x / z;` $\mid$ `x != 0` $\wedge$ `z = 0`$)$

Hinweis: In 4. steht `X` für den Wert, der `x` vor der Ausführung des Programms zugewiesen ist.

Lösung

1. $wp(\texttt{a = a + 2;} \mid \texttt{a = 42})$
   $= \texttt{a + 2 = 42}$
   $\Leftrightarrow \texttt{a = 40}$

2. $wp(\texttt{a = b * c; b = c + a; c = c - 17;} \mid \texttt{a = b} \wedge \texttt{b = c + 59})$
   $= wp(\texttt{a = b * c;} \mid wp(\texttt{b = c + a; c = c - 17;} \mid \texttt{a = b} \wedge \texttt{b = c + 59}))$
   $= wp(\texttt{a = b * c;} \mid wp(\texttt{b = c + a;} \mid wp(\texttt{c = c - 17;} \mid \texttt{a = b} \wedge \texttt{b = c + 59})))$
   $= wp(\texttt{a = b * c;} \mid wp(\texttt{b = c + a;} \mid \texttt{a = b} \wedge \texttt{b = c + 42}))$
   $= wp(\texttt{a = b * c;} \mid \texttt{a = c + a} \wedge \texttt{c + a = c + 42})$
   $\Leftrightarrow wp(\texttt{a = b * c;} \mid \texttt{0 = c} \wedge \texttt{a = 42})$
   $= \texttt{0 = c} \wedge \texttt{b * c = 42}$
   $\Leftrightarrow \texttt{false}$

3. $wp(\texttt{if (a == 0) abort(); else x = x + 1;} \mid \texttt{x > 0})$
   $= (\texttt{a = 0} \wedge wp(\texttt{abort();} \mid \texttt{x > 0})) \vee (\neg(\texttt{a = 0}) \wedge wp(\texttt{x = x + 1;} \mid \texttt{x > 0}))$
   $\Leftrightarrow (\texttt{a = 0} \wedge wp(\texttt{abort();} \mid \texttt{x > 0})) \vee (\texttt{a} \neq \texttt{0} \wedge wp(\texttt{x = x + 1;} \mid \texttt{x > 0}))$
   $= (\texttt{a = 0} \wedge \texttt{false}) \vee (\texttt{a} \neq \texttt{0} \wedge \texttt{x + 1 > 0})$
   $\Leftrightarrow \texttt{a} \neq \texttt{0} \wedge \texttt{x + 1 > 0}$
   $\Leftrightarrow \texttt{a} \neq \texttt{0} \wedge \texttt{x > -1}$

4. $wp(\texttt{if (x < 0) y = -x; else y = x;} \mid \texttt{y} > \texttt{0} \wedge \texttt{X * X = y * y})$
   $= (\texttt{x < 0} \wedge wp(\texttt{y = -x;} \mid \texttt{y} > \texttt{0} \wedge \texttt{X * X = y * y})) \vee (\neg(\texttt{x < 0}) \wedge wp(\texttt{y = x;} \mid \texttt{y} > \texttt{0} \wedge \texttt{X * X = y * y}))$
   $= (\texttt{x < 0} \wedge \texttt{-x} > \texttt{0} \wedge \texttt{X * X = (-x) * (-x)}) \vee (\neg(\texttt{x < 0}) \wedge \texttt{x} > \texttt{0} \wedge \texttt{X * X = x * x})$
   $\Leftrightarrow (\texttt{x < 0} \wedge \texttt{-x} > \texttt{0} \wedge \texttt{X * X = x * x}) \vee (\texttt{x} \geq \texttt{0} \wedge \texttt{x} > \texttt{0} \wedge \texttt{X * X = x * x})$
   $\Leftrightarrow (\texttt{x < 0} \wedge \texttt{x} < \texttt{0} \wedge \texttt{X * X = x * x}) \vee (\texttt{x} \geq \texttt{0} \wedge \texttt{x} > \texttt{0} \wedge \texttt{X * X = x * x})$
   $\Leftrightarrow (\texttt{x < 0} \wedge \texttt{X * X = x * x}) \vee (\texttt{x > 0} \wedge \texttt{X * X = x * x})$
   $\Leftrightarrow (\texttt{x < 0} \vee \texttt{x > 0}) \wedge \texttt{X * X = x * x}$
   $\Leftrightarrow \texttt{x} \neq \texttt{0} \wedge \texttt{X * X = x * x}$
   $\overset{\texttt{X = x}}{\Leftrightarrow} \texttt{x} \neq \texttt{0}$

5. $wp(\texttt{x = x / z;} \mid \texttt{x != 0} \wedge \texttt{z = 0})$
   $= \texttt{x / z != 0} \wedge \texttt{z = 0}$
   $\Leftrightarrow \texttt{false}$

Aufgabe 14.2: Korrekt?

Betrachten Sie das folgende Programm P. Zeigen oder widerlegen Sie, dass P korrekt ist.
Zeigen Sie dazu, dass [V]P[N] gilt, wobei V die Vorbedingung und N die Nachbedingung darstellt.
Gehen Sie dazu wie folgt vor:

- Überlegen Sie sich eine sinnvolle Belegung für V und N, welche der folgenden Beschreibung entspricht.
- Berechnen Sie $wp(P \mid N)$.
- Zeigen oder widerlegen Sie $V \Rightarrow wp(P \mid N)$.

Falls [V]P[N] nicht total korrekt ist, korrigieren Sie P und zeigen Sie die Korrektheit von P bzgl. V und N.
Das folgende Programm P implementiert den >-Operator. Das heißt, für eine Eingabe $rnd\,x, y$ wird x > y berechnet. Falls x > y gilt, soll res den Wert 1 haben, ansonsten 0. Zusätzlich soll P auf beliebigen Eingaben für x und y das korrekte Ergebnis liefern:

```
{
    r = x - y;
    if (r <= 0)
        res = 0;
    else ;
}
```

Lösung

- Vorbedingung: true
- Nachbedingung: $(\texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee (\texttt{res} = 0 \wedge \texttt{x} \leq \texttt{y})$
- Weakest Precondition bestimmen:

$$
\begin{aligned}
& wp(\texttt{r = x - y; if (r <= 0) res = 0; else ;} \mid (\texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee (\texttt{res} = 0 \wedge \texttt{x} \leq \texttt{y})) \\
= {} & wp(\texttt{r = x - y;} \mid \\
& \quad wp(\texttt{if (r <= 0) res = 0; else ;} \mid (\texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee (\texttt{res} = 0 \wedge \texttt{x} \leq \texttt{y}))) \\
= {} & wp(\texttt{r = x - y;} \mid \\
& \quad (\texttt{r} \leq 0 \wedge wp(\texttt{res = 0;} \mid (\texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee (\texttt{res} = 0 \wedge \texttt{x} \leq \texttt{y}))) \vee \\
& \quad (\neg(\texttt{r} \leq 0) \wedge wp(\texttt{;} \mid (\texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee (\texttt{res} = 0 \wedge \texttt{x} \leq \texttt{y})))) \\
= {} & wp(\texttt{r = x - y;} \mid \\
& \quad (\texttt{r} \leq 0 \wedge ((0 = 1 \wedge \texttt{x} > \texttt{y}) \vee (0 = 0 \wedge \texttt{x} \leq \texttt{y}))) \vee \\
& \quad (\neg(\texttt{r} \leq 0) \wedge ((\texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee (\texttt{res} = 0 \wedge \texttt{x} \leq \texttt{y})))) \\
\Leftrightarrow {} & wp(\texttt{r = x - y;} \mid \\
& \quad (\texttt{r} \leq 0 \wedge ((\texttt{false} \wedge \texttt{x} > \texttt{y}) \vee (\texttt{true} \wedge \texttt{x} \leq \texttt{y}))) \vee \\
& \quad (\texttt{r} > 0 \wedge ((\texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee (\texttt{res} = 0 \wedge \texttt{x} \leq \texttt{y})))) \\
\Leftrightarrow {} & wp(\texttt{r = x - y;} \mid \\
& \quad (\texttt{r} \leq 0 \wedge \texttt{x} \leq \texttt{y}) \vee \\
& \quad (\texttt{r} > 0 \wedge \texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee (\texttt{r} > 0 \wedge \texttt{res} = 0 \wedge \texttt{x} \leq \texttt{y})) \\
= {} & (\texttt{x - y} \leq 0 \wedge \texttt{x} \leq \texttt{y}) \vee (\texttt{x - y} > 0 \wedge \texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee (\texttt{x - y} > 0 \wedge \texttt{res} = 0 \wedge \texttt{x} \leq \texttt{y}) \\
\Leftrightarrow {} & (\texttt{x} \leq \texttt{y} \wedge \texttt{x} \leq \texttt{y}) \vee (\texttt{x} > \texttt{y} \wedge \texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee (\texttt{x} > \texttt{y} \wedge \texttt{res} = 0 \wedge \texttt{x} \leq \texttt{y}) \\
\Leftrightarrow {} & \texttt{x} \leq \texttt{y} \vee (\texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \vee \texttt{false} \\
\Leftrightarrow {} & \texttt{x} \leq \texttt{y} \vee (\texttt{res} = 1 \wedge \texttt{x} > \texttt{y}) \\
\Leftrightarrow {} & (\texttt{x} \leq \texttt{y} \vee \texttt{res} = 1) \wedge (\texttt{x} \leq \texttt{y} \vee \texttt{x} > \texttt{y}) \\
\Leftrightarrow {} & (\texttt{x} \leq \texttt{y} \vee \texttt{res} = 1) \wedge \texttt{true} \\
\Leftrightarrow {} & \texttt{x} \leq \texttt{y} \vee \texttt{res} = 1
\end{aligned}
$$

- $\Rightarrow$ [V]P[N] ist nicht erfüllt, da

$$true \not\Rightarrow \texttt{x} \leq \texttt{y} \vee \texttt{res} = 1$$

  Gegenbeispiel: $[\texttt{x} \to 1,\ \texttt{y} \to 0,\ \texttt{res} \to 0]$

- Verbessertes Programm:

```
{
    r = x - y;
    if(r <= 0)
        res = 0;
    else
        res = 1;
}
```

- Weakest Precondition bestimmen:

$$
\begin{aligned}
& wp(\texttt{r = x - y; if (r <= 0) res = 0; else res = 1;} \mid (\texttt{res} = 1 \land \texttt{x} > \texttt{y}) \lor (\texttt{res} = 0 \land \texttt{x} \leq \texttt{y})) \\
=\ & wp(\texttt{r = x - y;} \mid \\
& \quad wp(\texttt{if (r <= 0) res = 0; else res = 1;} \mid (\texttt{res} = 1 \land \texttt{x} > \texttt{y}) \lor (\texttt{res} = 0 \land \texttt{x} \leq \texttt{y}))) \\
=\ & wp(\texttt{r = x - y;} \mid \\
& \quad (\texttt{r} \leq 0 \land wp(\texttt{res = 0;} \mid (\texttt{res} = 1 \land \texttt{x} > \texttt{y}) \lor (\texttt{res} = 0 \land \texttt{x} \leq \texttt{y}))) \lor \\
& \quad (\neg(\texttt{r} \leq 0) \land wp(\texttt{res = 1;} \mid (\texttt{res} = 1 \land \texttt{x} > \texttt{y}) \lor (\texttt{res} = 0 \land \texttt{x} \leq \texttt{y})))) \\
=\ & wp(\texttt{r = x - y;} \mid \\
& \quad (\texttt{r} \leq 0 \land ((0 = 1 \land \texttt{x} > \texttt{y}) \lor (0 = 0 \land \texttt{x} \leq \texttt{y}))) \lor \\
& \quad (\neg(\texttt{r} \leq 0) \land ((1 = 1 \land \texttt{x} > \texttt{y}) \lor (1 = 0 \land \texttt{x} \leq \texttt{y})))) \\
\Leftrightarrow\ & wp(\texttt{r = x - y;} \mid \\
& \quad (\texttt{r} \leq 0 \land ((\texttt{false} \land \texttt{x} > \texttt{y}) \lor (\texttt{true} \land \texttt{x} \leq \texttt{y}))) \lor \\
& \quad (\texttt{r} > 0 \land ((\texttt{true} \land \texttt{x} > \texttt{y}) \lor (\texttt{false} \land \texttt{x} \leq \texttt{y})))) \\
\Leftrightarrow\ & wp(\texttt{r = x - y;} \mid (\texttt{r} \leq 0 \land \texttt{x} \leq \texttt{y}) \lor (\texttt{r} > 0 \land \texttt{x} > \texttt{y})) \\
=\ & (\texttt{x - y} \leq 0 \land \texttt{x} \leq \texttt{y}) \lor (\texttt{x - y} > 0 \land \texttt{x} > \texttt{y}) \\
\Leftrightarrow\ & (\texttt{x} \leq \texttt{y} \land \texttt{x} \leq \texttt{y}) \lor (\texttt{x} > \texttt{y} \land \texttt{x} > \texttt{y}) \\
\Leftrightarrow\ & \texttt{x} \leq \texttt{y} \lor \texttt{x} > \texttt{y} \\
\Leftrightarrow\ & \texttt{true}
\end{aligned}
$$

  $\Rightarrow$ `[V]P[N]` ist erfüllt, da *true* $\Rightarrow$ *true* eine Tautologie ist.

Aufgabe 14.3: triviale Zustandsmengen

Welche Zustandsmengen werden durch die Zusicherungen *true* und *false* beschrieben?

2 12 512 16

Lösung

Die Zusicherung *true* beschreibt die Menge aller Zustände, also Σ.
Die Zusicherung $false$ beschreibt die leere Menge, also $\varnothing$.

Aufgabe 14.4: Entscheidunsprobleme

Gegeben ist folgendes C0pb-Programm:

256 25 1024 4

```
{
    a = A;
    b = B;
    while (b != 0) {
        if (a > b)
            a = a - b;
```

```
        else
            b = b - a;
    }
    r = a;
}
```

mit der Vorbedingung $V = A > 0 \wedge B \geq 0$.

1. Begründen oder widerlegen Sie, dass die folgenden Ausdrücke Invarianten der Schleife des Programms beschreiben.
   (a) $I_1 = a > 0 \wedge b \geq 0$
   (b) $I_2 = a > 0 \wedge b > 0$
   (c) $I_3 = A \mod a \neq 0 \wedge B \mod a \neq 0$

Lösung

1. Sei im folgenden `S = if (a > b) a = a - b; else b = b - a;`.
   (a) Ja, I_1 ist eine gültige Schleifeninvariante.

$$\begin{aligned} wp(S|I_1) &= wp(\text{if } (a > b) a = a - b; \text{ else } b = b - a; \mid a > 0 \wedge b \geq 0) \\ &= (a > b \wedge a - b > 0 \wedge b \geq 0) \vee (a \leq b \wedge a > 0 \wedge b - a \geq 0) \\ &\Leftarrow a > 0 \wedge b > 0 \Leftrightarrow I_1 \wedge b \neq 0 \end{aligned}$$

   (b) Nein, denn für alle $a = b$ mit $a, b > 0$ ist $b - a > 0$ bzw. $a - b > 0$ nicht erfüllt. Daher $I_2 \wedge b \neq 0 \not\Rightarrow wp(S \mid I_2)$. Wobei

$$wp(S|I_2) = (a > b \wedge a - b > 0 \wedge b > 0) \vee (a \leq b \wedge a > 0 \wedge b - a > 0) \not\Leftarrow I_2 \wedge b \neq 0$$

   (c) Nein, denn für z.B. $A = 7, B = 3, a = 4$ und $b = 3$ ist $I_3 \wedge b \neq 0$ erfüllt, aber $a \leq b$ bzw. $A \bmod a - b \neq 0 \Leftrightarrow A \bmod 1 \neq 0$ sind nicht erfüllt.
   Daher $I_3 \wedge b \neq 0 \not\Rightarrow wp(S \mid I_3)$. Wobei

$$\begin{aligned} wp(S|I_3) =& (a > b \wedge A \bmod a - b \neq 0 \wedge B \bmod a - b \neq 0) \\ & \vee (a \leq b \wedge A \bmod a \neq 0 \wedge B \bmod a \neq 0) \wedge b \neq 0 \end{aligned}$$

Aufgabe 14.5: Schleifeninvarianten und Terminierungsfunktionen

Betrachten Sie folgende Programme, die Invarianten `I` und die Funktionen `t`:

(1)
```
x = 10;
y = 5;
while (x <= y) {
    y = y - x;
}
```

mit `I` $:= \mathit{true}$ und `t` $:= x$

(2)
```
x = 0;
b = B;
while (b > 0) {
    x = x + a;
    b = b - 1;
}
```

mit $\texttt{I} := x = (B - b) * a$ und $\texttt{t} := b$

(3)
```
a = 0;
x' = x;
while (x > 0) {
    t = x % 10;
    a += t;
    x -= t;
    x /= 10;
}
```

mit $\texttt{I} := a = x + t$ und $\texttt{t} := x$

(a) Handelt es sich bei den Angaben um gültige Schleifeninvarianten? Argumentieren Sie jeweils kurz!

(b) Beweisen Sie Ihre Antworten nun formal mit Hilfe des wp-Kalküls!

(c) Falls es sich um keine gültigen Schleifeninvarianten handelt, finden Sie eigene (möglichst starke) und beweisen Sie deren Gültigkeit!

Lösung

(a) (1) I ist trivialerweise eine Invariante .

(2) I ist eine Invariante, da `B - b` gerade die Anzahl der Schleifeniterationen zählt und x in jeder solchen Iteration um a erhöht wird.

(3) I ist keine Invariante, da $\texttt{I} \wedge \texttt{x > 0} \not\Rightarrow wp(\texttt{s} \mid \texttt{I})$, wobei S der Schleifenrumpf ist.

(b) Es gilt im Allgemeinen:

- I ist eine Schleifeninvariante, wenn gilt:
  Schleifenbedingung $\wedge$ I $\Rightarrow$ wp(Schleifenrumpf | I)

(1) $\texttt{I} := \texttt{true}$ ist eine Schleifeninvariante:

$$\begin{aligned} & wp(\texttt{y = y - x;} \mid \texttt{true}) \\ =\ & \texttt{true} \end{aligned}$$

(2) $\texttt{I} := \texttt{x} = (\texttt{B - b}) * \texttt{a}$ ist eine Schleifeninvariante:

$$\begin{aligned} & wp(\texttt{x = x + a; b = b - 1;} \mid \texttt{x} = (\texttt{B} - \texttt{b}) * \texttt{a}) \\ =\ & wp(\texttt{x = x + a;} \mid wp(\texttt{b = b - 1;} \mid \texttt{x} = (\texttt{B} - \texttt{b}) * \texttt{a})) \\ =\ & wp(\texttt{x = x + a;} \mid \texttt{x} = (\texttt{B} - \texttt{b} + 1) * \texttt{a}) \\ =\ & \texttt{x} + \texttt{a} = (\texttt{B} - \texttt{b} + 1) * \texttt{a} \\ \Leftrightarrow\ & \texttt{x} = (\texttt{B} - \texttt{b}) * \texttt{a} \end{aligned}$$

(3) $\texttt{I} := \texttt{a} = \texttt{x} + \texttt{t}$ ist *keine* Schleifeninvariante:

$$\begin{aligned} & wp(\texttt{t = x \% 10; a = a + t; x = x - t; x = x / 10;} \mid \texttt{a = x + t}) \\ =\ & wp(\texttt{t = x \% 10;} \mid wp(\texttt{a = a + t;} \mid wp(\texttt{x = x - t;} \mid wp(\texttt{x = x / 10;} \mid \texttt{a = x + t})))) \\ =\ & wp(\texttt{t = x \% 10;} \mid wp(\texttt{a = a + t;} \mid wp(\texttt{x = x - t;} \mid \texttt{a = x / 10 + t}))) \\ =\ & wp(\texttt{t = x \% 10;} \mid wp(\texttt{a = a + t;} \mid \texttt{a = (x - t) / 10 + t})) \\ =\ & wp(\texttt{t = x \% 10;} \mid \texttt{a + t = (x - t) / 10 + t}) \\ \Leftrightarrow\ & wp(\texttt{t = x \% 10;} \mid \texttt{a = (x - t) / 10 }) \\ =\ & \texttt{a = (x - x \% 10) / 10} \\ \not\Leftarrow\ & \texttt{x > 0} \wedge \texttt{a = x + t} \quad (\text{z.B. } [\texttt{a} \mapsto 10, \texttt{x} \mapsto 9, \texttt{t} \mapsto 1]) \end{aligned}$$

(c) (1) Bereits in Teil (b) vollständig. Aber es fällt auf, dass wir eine stärkere Invariante benutzt haben. Dies war nötig, um `t'` als Terminierungsfunktion zu beweisen. Nachfolgend noch der Beweis, dass `I' := x > 0` eine Schleifeninvariante ist:

$$
\begin{aligned}
& wp(\texttt{y = y - x;} \mid \texttt{x > 0}) \\
=\ & \texttt{x > 0} \\
\Leftarrow\ & \texttt{x > 0} \land \texttt{x} \leq \texttt{y} \quad \checkmark
\end{aligned}
$$

(2) Bereits in Teil (b) vollständig.

(3) Wähle `I' := x ≥ 0` und zeige, dass es sich um eine Schleifeninvariante handelt:

$$
\begin{aligned}
& wp(\texttt{t = x \% 10; a = a + t; x = x - t; x = x / 10;} \mid \texttt{x} \geq 0) \\
=\ & wp(\texttt{t = x \% 10;} \mid wp(\texttt{a = a + t;} \mid wp(\texttt{x = x - t;} \mid wp(\texttt{x = x / 10;} \mid \texttt{x} \geq 0)))) \\
=\ & wp(\texttt{t = x \% 10;} \mid wp(\texttt{a = a + t;} \mid wp(\texttt{x = x - t;} \mid \texttt{x / 10} \geq 0))) \\
=\ & wp(\texttt{t = x \% 10;} \mid wp(\texttt{a = a + t;} \mid \texttt{(x - t) / 10} \geq 0)) \\
=\ & wp(\texttt{t = x \% 10;} \mid \texttt{(x - t) / 10} \geq 0) \\
=\ & \texttt{(x - x \% 10) / 10} \geq 0 \\
\Leftrightarrow\ & \texttt{x - x \% 10} \geq 0 \\
\Leftrightarrow\ & \texttt{x} \geq \texttt{x \% 10} \\
\Leftarrow\ & \texttt{x} > 0 \\
\Leftrightarrow\ & \texttt{x} > 0 \land \texttt{x} \geq 0
\end{aligned}
$$

Aufgabe 14.6: Schon wieder Schleifen ...

Im Folgendenden sollen Sie die totale Korrektheit von kleinen Programmen `P` zeigen. Finden Sie dazu zu jeder While-Schleife eine ausreichend starke Invariante `I` und eine Terminierungsfunktion `t`. Beweisen Sie danach die Gültigkeit der Invariante `I` und der Terminierungsfunktion `t`. Nutzen Sie dann beides, um die totale Korrektheit von `P` zu zeigen.
(Hinweis: Vergessen Sie nicht zu zeigen, dass die Vorbedingung eine Teilmenge der Menge `Wp` darstellt, welche wir mit Hilfe des wp-Kalküls erhalten.)

1. Multiplikation

```
[ x ≥ 0 ∧ y ≥ 0 ∧ res = 0 ]

y' = y;
while (y > 0) {
    res = res + x;
    y = y - 1;
}

[ res = x * y' ]
```

2. Fakultät

```
[ x ≥ 0 ∧ res = 1 ]

if (x == 0) ;
else
    while (x > 0) {
        res = res * x;
        x = x - 1;
    }

[ res = X! ∧ x = 0 ]
```

3. Quadratwurzel

```
[ x ≥ 0 ∧ y = 0 ∧ res = 0 ]
while (y * y <= x ) {
    y = y + 1;
}
res = y - 1;
[ res² ≤ x ]
```

Lösung

Zunächst eine kurze Übersicht, wie wir die totale Korrektheit eines Programms `P` mit Vorbedingung `V`, Nachbedingung `N` und Schleife `S'` mit Schleifenrumpf `s` und Bedingung `b` beweisen:

- Suche eine Schleifeninvariante `I` für `S'` und beweise, dass es sich dabei um eine gültige Schleifeninvariante handelt (Zeige, dass $\mathtt{B} \cap \mathtt{I} \subseteq \mathtt{Wp(s \mid I)}$ hält).
- Finde eine Terminierungsfunktion `t` für `S'` und beweise, dass es sich dabei um eine gültige Terminierungsfunktion für `S'` handelt (Zeige, dass $\forall \mathtt{k} \in \mathbb{N}.[\mathtt{b} \wedge \mathtt{I} \wedge 0 \leq \mathtt{t} \leq k+1]$ `s` $[\mathtt{I} \wedge 0 \leq \mathtt{t} \leq k]$ hält).
- Beweise, dass das Programm nach der Schleife, mit Vorbedingung $\mathtt{I} \wedge \neg \mathtt{b}$, in der Nachbedingung `N` terminiert (Zeige, dass $[\mathtt{I} \wedge \neg \mathtt{b}]$ $\mathtt{s}_{\mathtt{nach\_der\_Schleife}}$ $[\mathtt{N}]$ hält).
- Beweise, dass das Programm vor der Schleife, mit Vorbedingung `V`, in der Nachbedingung $\mathtt{I} \wedge \mathtt{t} \geq 0$ terminiert (Zeige, dass $[\mathtt{V}]$ $\mathtt{s}_{\mathtt{vor\_der\_Schleife}}$ $[\mathtt{I} \wedge \mathtt{t} \geq 0]$ hält).

Nach Konvention des Skripts beschreiben Großbuchstaben Zustandsmengen und Kleinbuchstaben Prädikate ($[\![\mathtt{b}]\!] = \mathtt{B}$).

1. Wähle $\mathtt{I} := \mathtt{res = (y' - y) * x \wedge y \geq 0}$ und $\mathtt{t} := \mathtt{y}$.

   - Zeige `I` ist eine Schleifeninvariante, dazu muss $\mathtt{B} \cap \mathtt{I} \subseteq \mathtt{Wp(s \mid I)}$ gelten:

$$
\begin{aligned}
&\mathtt{wp(res = res + x; y = y - 1; \mid res = (y' - y) * x \wedge y \geq 0)} \\
=\ &\mathtt{wp(res = res + x; \mid wp(y = y - 1; \mid res = (y' - y) * x \wedge y \geq 0))} \\
=\ &\mathtt{wp(res = res + x; \mid res = (y' - (y - 1)) * x \wedge y - 1 \geq 0)} \\
=\ &\mathtt{res + x = (y' - (y - 1)) * x \wedge y - 1 \geq 0} \\
\Leftrightarrow\ &\mathtt{res + x = (y' - y + 1) * x \wedge y \geq 1} \\
\Leftrightarrow\ &\mathtt{res + x = (y' - y) * x + x \wedge y \geq 1} \\
\Leftrightarrow\ &\mathtt{res = (y' - y) * x \wedge y \geq 1} \\
\Leftarrow\ &\mathtt{res = (y' - y) * x \wedge y \geq 0 \wedge y > 0}
\end{aligned}
$$

   - Zeige `t` ist eine Terminierungsfunktion, dazu muss gelten: $[\mathtt{b} \wedge \mathtt{I} \wedge 0 \leq \mathtt{t} \leq k+1]$ `s` $[\mathtt{I} \wedge 0 \leq \mathtt{t} \leq k]$

$$
\begin{aligned}
&\mathtt{wp(res = res + x; y = y - 1; \mid res = (y' - y) * x \wedge y \geq 0 \wedge 0 \leq y \leq} k\mathtt{)} \\
=\ &\mathtt{wp(res = res + x; \mid wp(y = y - 1; \mid res = (y' - y) * x \wedge y \geq 0 \wedge 0 \leq y \leq} k\mathtt{))} \\
=\ &\mathtt{wp(res = res + x; \mid res = (y' - (y - 1)) * x \wedge y - 1 \geq 0 \wedge 0 \leq y - 1 \leq} k\mathtt{)} \\
=\ &\mathtt{res + x = (y' - (y - 1)) * x \wedge y - 1 \geq 0 \wedge 0 \leq y - 1 \leq} k \\
\Leftrightarrow\ &\mathtt{res = (y' - y) * x \wedge 1 \leq y \leq} k + 1 \\
\Leftarrow\ &\mathtt{res = (y' - y) * x \wedge y \geq 0 \wedge y > 0 \wedge 0 \leq y \leq} k + 1
\end{aligned}
$$

- Zeige wir nun, dass nach der letzten Schleifendurchführung die Schleifeninvariante die Nachbedingung impliziert, also $\texttt{I} \wedge \neg \texttt{b} \Rightarrow \texttt{res = x * y'}$:

$$
\begin{aligned}
& \texttt{I} \wedge \neg(\texttt{y > 0}) \\
\Leftrightarrow\ & \texttt{res = (y' - y) * x} \wedge \texttt{y} \geq 0 \wedge \texttt{y} \leq 0 \\
\Leftrightarrow\ & \texttt{res = (y' - y) * x} \wedge \texttt{y} = 0 \\
\Leftrightarrow\ & \texttt{res = (y' - 0) * x} \\
\Leftrightarrow\ & \texttt{res = x * y'}
\end{aligned}
$$

- Zeige zuletzt, dass $\texttt{V} \Rightarrow \texttt{wp}(\texttt{s}_{\texttt{vor\_der\_Schleife}} \mid \texttt{I} \wedge \texttt{t} \geq 0)$, also:
  $\texttt{x} \geq 0 \wedge \texttt{y} \geq 0 \wedge \texttt{res = 0} \Rightarrow \texttt{wp(y' = y; |res = (y' - y) * x} \wedge \texttt{y} \geq 0 \wedge \texttt{y} \geq 0)$ gilt:

$$
\begin{aligned}
& \texttt{wp(y' = y; |res = (y' - y) * x} \wedge \texttt{y} \geq 0 \wedge \texttt{y} \geq 0) \\
=\ & \texttt{res = (y - y) * x} \wedge \texttt{y} \geq 0 \\
\Leftrightarrow\ & \texttt{res = 0} \wedge \texttt{y} \geq 0 \\
\Leftarrow\ & \texttt{x} \geq 0 \wedge \texttt{y} \geq 0 \wedge \texttt{res = 0}
\end{aligned}
$$

Damit haben wir die totale Korrektheit von P bezüglich V und N bewiesen.

2. Wähle $\mathtt{I} := \mathtt{res} = \mathtt{X!}/\mathtt{x!} \wedge \mathtt{x} \geq 0$ und $\mathtt{t} := \mathtt{x}$.

   - Zeige $\mathtt{I}$ ist eine Schleifeninvariante , dazu muss $\mathtt{B} \cap \mathtt{I} \subseteq \mathtt{Wp}(\mathtt{s} \mid \mathtt{I})$ gelten:

$$
\begin{aligned}
& \mathtt{wp(res = res * x; x = x - 1; \mid res = X!/x! \wedge x \geq 0)} \\
=\ & \mathtt{res * x = X!/(x - 1)! \wedge x - 1 \geq 0} \\
\Leftrightarrow\ & \mathtt{res * x = X!/(x - 1)! \wedge x \geq 1} \\
\Leftarrow\ & \mathtt{res = X!/((x - 1)! * x) \wedge x > 0} \\
\Leftrightarrow\ & \mathtt{res = X!/x! \wedge x \geq 0 \wedge x > 0}
\end{aligned}
$$

   - Zeige $\mathtt{t}$ ist eine Terminierungsfunktion: $[\mathtt{b} \wedge \mathtt{I} \wedge 0 \leq \mathtt{t} \leq k+1]\ \mathtt{s}\ [\mathtt{I} \wedge 0 \leq \mathtt{t} \leq k]$

$$
\begin{aligned}
& \mathtt{wp(res = res * x; x = x - 1; \mid res = X!/x! \wedge x \geq 0 \wedge 0 \leq x \leq} k) \\
\Leftrightarrow\ & \mathtt{wp(res = res * x; x = x - 1; \mid res = X!/x! \wedge 0 \leq x \leq} k) \\
=\ & \mathtt{res * x = X!/(x - 1)! \wedge 0 \leq x - 1 \leq} k \\
\Leftrightarrow\ & \mathtt{res * x = X!/(x - 1)! \wedge 1 \leq x \leq} k+1 \\
\Leftarrow\ & \mathtt{res = X!/((x - 1)! * x) \wedge 1 \leq x \leq} k+1 \\
\Leftrightarrow\ & \mathtt{res = X!/x! \wedge x > 0 \wedge 0 \leq x \leq} k+1
\end{aligned}
$$

   - Zeige $\mathtt{I} \wedge \neg(\mathtt{x} > 0) \Rightarrow \mathtt{res = X! \wedge x = 0}$:

$$
\begin{aligned}
& \mathtt{I} \wedge \neg(\mathtt{x} > 0) \\
\Leftrightarrow\ & \mathtt{res = X! / x! \wedge x \geq 0 \wedge x \leq 0} \\
\Leftrightarrow\ & \mathtt{res = X! / x! \wedge x = 0} \\
\Leftrightarrow\ & \mathtt{res = X! / 1 \wedge x = 0} \\
\Leftrightarrow\ & \mathtt{res = X! \wedge x = 0}
\end{aligned}
$$

   - Zeige $\mathtt{x} \geq 0 \wedge \mathtt{res} = 1 \Rightarrow$
     $\mathtt{wp(if\ (x == 0)\ ;\ else\ while\ (x > 0)\ \{\ res = res * x;\ x = x - 1;\ \}\mid res = X!\ \wedge\ x = 0)}$:

$$
\begin{aligned}
& \mathtt{wp(if\ (x == 0)\ ;\ else\ while\ (x > 0)\ \{\ res = res * x;\ x = x - 1;\ \} \mid res = X! \wedge x = 0)} \\
=\ & (\mathtt{x = 0} \wedge \mathtt{wp(; \mid res = X! \wedge x = 0)}) \vee \\
& (\mathtt{x} \neq 0 \wedge \mathtt{wp(while\ (x > 0)\ \{\ res = res * x;\ x = x - 1;\ \} \mid res = X! \wedge x = 0)}) \\
=\ & (\mathtt{x = 0} \wedge \mathtt{res = X! \wedge x = 0}) \vee (\mathtt{x} \neq 0 \wedge \mathtt{res = X! / x! \wedge x \geq 0 \wedge x \geq 0}) \\
\Leftrightarrow\ & (\mathtt{x = 0} \wedge \mathtt{res = X!}) \vee (\mathtt{x} > 0 \wedge \mathtt{res = X! / x!}) \\
\overset{\mathtt{X = x}}{\Leftrightarrow}\ & (\mathtt{x = 0} \wedge \mathtt{res = 1}) \vee (\mathtt{x} > 0 \wedge \mathtt{res = 1}) \\
\Leftrightarrow\ & (\mathtt{x = 0} \vee \mathtt{x} > 0) \wedge \mathtt{res = 1} \\
\Leftrightarrow\ & \mathtt{x} \geq 0 \wedge \mathtt{res = 1}
\end{aligned}
$$

3. Wähle $\texttt{I} := ((\texttt{y}-1)^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq 0$ und $\texttt{t} := \texttt{x - y + 1}$.

   - Zeige I ist eine Schleifeninvariante:

$$\begin{aligned}
&\texttt{wp(y = y + 1; } | \ ((\texttt{y}-1)^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq 0) \\
&= ((\texttt{y}+1-1)^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y}+1 \geq 0 \\
&\Leftrightarrow (\texttt{y}^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq \texttt{-1} \\
&\Leftarrow \texttt{y}^2 \leq \texttt{x} \wedge \texttt{y} \geq 0 \\
&\Leftarrow ((\texttt{y}-1)^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq 0 \wedge \texttt{y}^2 \leq \texttt{x}
\end{aligned}$$

   - Zeige t ist eine Terminierungsfunktion:

$$\begin{aligned}
&\texttt{wp(y = y + 1; } | \ ((\texttt{y}-1)^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq 0 \wedge 0 \leq \texttt{x}-\texttt{y}+1 \leq k) \\
&= (\texttt{y}^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq \texttt{-1} \wedge 0 \leq \texttt{x} - (\texttt{y}+1) + 1 \leq k \\
&\Leftrightarrow (\texttt{y}^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq \texttt{-1} \wedge 0 \leq \texttt{x}-\texttt{y} \leq k \\
&\Leftrightarrow (\texttt{y}^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq \texttt{-1} \wedge 1 \leq \texttt{x}-\texttt{y}+1 \leq k+1 \\
&\Leftarrow \texttt{y}^2 \leq \texttt{x} \wedge \texttt{y} \geq 0 \wedge 1 \leq \texttt{x}-\texttt{y}+1 \leq k+1 \\
&\Leftarrow ((\texttt{y}-1)^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq 0 \wedge \texttt{y}^2 \leq \texttt{x} \wedge 0 \leq \texttt{x}-\texttt{y}+1 \leq k+1
\end{aligned}$$

     Dabei gilt die letzte Implikation, da:

$$\begin{aligned}
&((\texttt{y}-1)^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq 0 \wedge \texttt{y}^2 \leq \texttt{x} \wedge 0 \leq \texttt{x}-\texttt{y}+1 \leq k+1 \\
&\Rightarrow 0 \leq \texttt{y} \leq \texttt{y}^2 \leq \texttt{x} \\
&\Rightarrow 1 \leq \texttt{y}+1 \leq \texttt{y}^2+1 \leq \texttt{x}+1 \\
&\Rightarrow \texttt{y}+1 \leq \texttt{x}+1 \\
&\Rightarrow 1 \leq \texttt{x}-\texttt{y}+1
\end{aligned}$$

   - Zeige $\texttt{I} \wedge \neg(\texttt{y}^2 \leq \texttt{x}) \Rightarrow \texttt{wp(res = y - 1; } | \ \texttt{res}^2 \leq \texttt{x})$:

$$\begin{aligned}
&\texttt{I} \wedge \neg(\texttt{y}^2 \leq \texttt{x}) \\
&\Leftrightarrow ((\texttt{y}-1)^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq 0 \wedge \texttt{y}^2 > \texttt{x} \\
&\Rightarrow (\texttt{y}-1)^2 \leq \texttt{x} \\
&= \texttt{wp(res = y - 1; } | \ \texttt{res}^2 \leq \texttt{x})
\end{aligned}$$

   - Zeige $\texttt{x} \geq 0 \wedge \texttt{y} = 0 \wedge \texttt{res} = 0 \Rightarrow ((\texttt{y}-1)^2 \leq \texttt{x} \vee \texttt{x} = 0) \wedge \texttt{y} \geq 0 \wedge \texttt{x}-\texttt{y}+1 \geq 0$:
     - Fall $\texttt{x} = 0$:

$$\begin{aligned}
&(0-1)^2 \leq 0 \vee 0 = 0 \quad \checkmark \\
&0 \geq 0 \quad \checkmark \\
&0-0+1 \geq 0 \quad \checkmark
\end{aligned}$$

     - Fall $\texttt{x} > 0$:

$$\begin{aligned}
&(0-1)^2 = 1 \leq \texttt{x} \vee \texttt{x} = 0 \quad \checkmark \\
&0 \geq 0 \quad \checkmark \\
&\texttt{x}-0+1 \geq 0 \quad \checkmark
\end{aligned}$$

Aufgabe 14.7: Wp Konjunktion

Beweisen Sie $\text{Wp}(S \mid A \cap B) = \text{Wp}(S \mid A) \cap \text{Wp}(S \mid B)$.

Lösung

Mit Definition 6.14 ($\mathrm{Wp}(P \mid N) := \{\sigma | \langle P|\sigma\rangle \Downarrow \sigma' \wedge \sigma' \in N\}$) folgt:

$$\begin{aligned}\mathrm{Wp}(S \mid A \cap B) &= \{\sigma | \langle S|\sigma\rangle \Downarrow \sigma' \wedge \sigma' \in A \cap B\} \\ &= \{\sigma | \langle S|\sigma\rangle \Downarrow \sigma' \wedge \sigma' \in A\} \cap \{\sigma | \langle S|\sigma\rangle \Downarrow \sigma' \wedge \sigma' \in B\} \\ &= \mathrm{Wp}(S \mid A) \cap \mathrm{Wp}(S \mid B)\end{aligned}$$