Einzelprüfung „Softwaretechnologie / Datenbanksysteme (nicht vertieft)“

Einzelprüfungsnummer 46116 / 2015 / Herbst

Thema 2 / Teilaufgabe 1 / Aufgabe 3

(*ASCII*)

Stichwörter: Formale Verifikation, wp-Kalkül

Sei wp(A, Q) die schwächste Vorbedingung (weakest precondition) eines Programmfragments A bei gegebener Nachbedingung Q so, dass A alle Eingaben, die wp(A,Q) erfüllen, auf gültige Ausgaben abbildet, die Q erfüllen.

Bestimmen Sie schrittweise und formal (mittels Floyd-Hoare-Kalkül) jeweils wp(A, Q) für folgende Code-Fragmente A und Nachbedingungen Q und vereinfachen Sie dabei den jeweils ermittelten Ausdruck so weit wie möglich.

Die Variablen x, y und z in folgenden Pseudo-Codes seien ganzzahlig (vom Typ int). Zur Vereinfachung nehmen Sie bitte im Folgenden an, dass die verwendeten Datentypen unbeschränkt sind und daher keine Überläufe auftreten können.

(a) Sequenz:

```
x = -2 * (x + 2 * y);
y += 2 * x + y + z;
z -= x - y - z;
```

$Q :\equiv x = y + z$

Lösungsvorschlag

Code umformulieren:

```
x = -2 * (x + 2 * y);
y = y + 2 * x + y + z;
z = z - (x - y - z);
```

$\text{wp}(\texttt{"x=-2*(x+2*y);y=2*y+2*x+z;z=z-(x-y-z);"}, x = y + z)$

z eingesetzen

$\equiv \quad \text{wp}(\texttt{"x=-2*(x+2*y);y=2*y+2*x+z;"}, x = y + (z - (x - y - z)))$

Innere Klammer auflösen

$\equiv \quad \text{wp}(\texttt{"x=-2*(x+2*y);y=2*y+2*x+z;"}, x = y + (-x + y - 2z))$

Klammer auflösen

$\equiv \quad \text{wp}(\texttt{"x=-2*(x+2*y);y=2*y+2*x+z;"}, x = -x + 2y + 2z)$

$-x$ auf beiden Seiten

$\equiv \quad \text{wp}(\texttt{"x=-2*(x+2*y);y=2*y+2*x+z;"}, 0 = -2x + 2y + 2z)$

$\div 2$ auf beiden Seiten

$$\equiv \quad \mathrm{wp}(\texttt{"x=-2*(x+2*y);y=2*y+2*x+z;"}, 0 = -x + y + z)$$

y einsetzen

$$\equiv \quad \mathrm{wp}(\texttt{"x=-2*(x+2*y);"}, 0 = -x + (2y + 2x + z) + z)$$

Term vereinfachen

$$\equiv \quad \mathrm{wp}(\texttt{"x=-2*(x+2*y);"}, 0 = x + 2y + 2z)$$

x einsetzen

$$\equiv \quad \mathrm{wp}(\texttt{""}, 0 = (-2(x + 2y)) + 2y + 2z)$$

wp weglassen

$$\equiv \quad 0 = (-2(x + 2y)) + 2y + 2z$$

ausmultiplizieren

$$\equiv \quad 0 = (-2x - 4y) + 2y + 2z$$

Klammer auflösen, vereinfachen

$$\equiv \quad 0 = -2x - 2y + 2z$$

$\div 2$ auf beiden Seiten

$$\equiv \quad 0 = -x - y + z$$

x nach links holen mit $+x$ auf beiden Seiten

$$\equiv \quad x = -y + z$$

y ganz nach links schreiben

$$\equiv \quad x = z - y$$

$$x = -2 \cdot (x + 2 \cdot y)$$

(b) Verzweigung:

```
if (x < y) {
  x = y + z;
} else if (y > 0) {
  z = y - 1;
} else {
  x -= y -= z;
}
```

$Q :\equiv x > z$

Lösungsvorschlag

1. Fall: $x < y$

2. Fall: $x \geq y \wedge y > 0$

3. Fall: $x \geq y \wedge y \leq 0$

Code umformulieren:

```
if (x < y) {
  x = y + z;
} else if (x >= y && y > 0) {
  z = y - 1;
} else {
  y = y - z;
  x = x - y;
}
```

$\text{wp}(\texttt{"if(x<y){x=y+z;}else if(x>=y\&\&y>0){z=y-1;}else{y=y-z;x=x-y;}"}, x > z)$

$\equiv$ (In mehrere kleinere wp-Kalküle aufsplitten)

$$\Big((x < y) \wedge \text{wp}(\texttt{"x=y+z;"}, x > z)\Big) \vee$$
$$\Big((x \geq y \wedge y > 0) \wedge \text{wp}(\texttt{"z=y-1;"}, x > z)\Big) \vee$$
$$\Big((x \geq y \wedge y \leq 0) \wedge \text{wp}(\texttt{"y=y-z;x=x-y;"}, x > z)\Big)$$

$\equiv$ (Code einsetzen)

$$\Big((x < y) \wedge \text{wp}(\texttt{""}, y + z > z)\Big) \vee$$
$$\Big((x \geq y \wedge y > 0) \wedge \text{wp}(\texttt{""}, x > y - 1)\Big) \vee$$
$$\Big((x \geq y \wedge y \leq 0) \wedge \text{wp}(\texttt{"y=y-z;"}, x - y > z)\Big)$$

$\equiv$ (wp-Kalkül-Schreibweise weg lassen, Code weiter einsetzen)

$$\Big((x < y) \wedge y + z > z\Big) \vee$$
$$\Big((x \geq y \wedge y > 0) \wedge x > y - 1\Big) \vee$$
$$\Big((x \geq y \wedge y \leq 0) \wedge \mathrm{wp}(\texttt{""}, x - (y - z) > z)\Big)$$

$\equiv$ (Terme vereinfachen, wp-Kalkül-Schreibweise weg lassen)

$$\Big(x < y \wedge y > 0\Big) \vee$$
$$\Big(x \geq y^{a} \wedge y > 0\Big) \vee$$
$$\Big((x \geq y \wedge y \leq 0) \wedge x - (y - z) > z\Big)$$

$\equiv$ (letzten Term vereinfachen)

$$\Big(x < y \wedge y > 0\Big) \vee$$
$$\Big(x \geq y \wedge y > 0\Big) \vee$$
$$\Big((x \geq y \wedge y \leq 0) \wedge x - y > 0\Big)$$

$\equiv$ (ein $\wedge$ eleminieren)

$$\Big(x < y \wedge y > 0\Big) \vee$$
$$\Big(x \geq y \wedge y > 0\Big) \vee$$
$$\Big(y \leq 0 \wedge x > y\Big)$$

[a] $x > y - 1 \wedge x \geq y$ ergibt $x \geq y$
[a] $x > y - 1 \wedge x \geq y$ ergibt $x \geq y$

(c) Mehrfachauswahl:

```
switch (z) {
  case "x":
```

```
      y = "x";
    case "y":
      y = --z;
      break;
    default:
      y = 0x39 + "?";
  }
```

$Q :\equiv$ `'x'` $= y$

Hinweis zu den ASCII-Codes

- `'x'` $= 120_{(10)}$
- `'y'` $= 121_{(10)}$
- `0x39` $= 57_{(10)}$
- `'?'` $= 63_{(10)}$

Lösungsvorschlag

Mehrfachauswahl in Bedingte Anweisungen umschreiben. Dabei beachten, dass bei fehlendem `break` die Anweisungen im folgenden Fall bzw. ggf. in den folgenden Fällen ausgeführt werden:

```
if (z == "x") {
  y = "x";
  y = z - 1;
} else if (z == "y") {
  y = z - 1;
} else {
  y = 0x39 + "?";
}
```

Da kein `break` im Fall `z == "x"`. `--z` bedeutet, dass die Variable erst um eins verringert und dann zugewiesen wird.

```
if (z == 120) {
  y = 120;
  y = 120 - 1;
} else if (z == 121) {
  y = 121 - 1;
} else {
  y = 57 + 63;
}
```

Vereinfachung / Zusammenfassung:

```
if (z == 120) {
  y = 120;
  y = 119;
} else if (z == 121) {
  y = 120;
```

```
} else {
  y = 120;
}
```

$$\text{wp}(\texttt{"if(z==120)\{y=120;y=119;\}else if(z==121)\{y=120;\}else\{y=120;\}"}, 120 = y)$$

$\equiv$ (In mehrere kleinere wp-Kalküle aufsplitten)

$$\begin{aligned}&\Big((z = 120) \land \text{wp}(\texttt{"y=120;y=119;"}, 120 = y)\Big) \lor \\ &\Big(((z \neq 120) \land (z = 121)) \land \text{wp}(\texttt{"y=120;"}, 120 = y)\Big) \lor \\ &\Big(((z \neq 120) \land (z \neq 121)) \land \text{wp}(\texttt{"y=120;"}, 120 = y)\Big)\end{aligned}$$

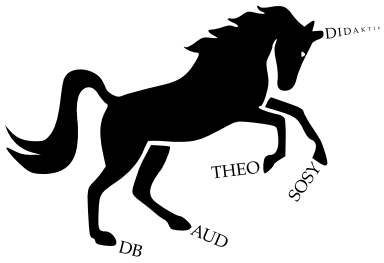
$\equiv$ (Code einsetzen)

$$\begin{aligned}&\Big((z = 120) \land \text{wp}(\texttt{"y=120;"}, 120 = 119)\Big) \lor \\ &\Big(((z \neq 120) \land (z = 121)) \land \text{wp}(\texttt{""}, 120 = 120)\Big) \lor \\ &\Big(((z \neq 120) \land (z \neq 121)) \land \text{wp}(\texttt{""}, 120 = 120)\Big)\end{aligned}$$

$\equiv$ (vereinfachen)

$$\begin{aligned}&\text{false} \lor \\ &\Big((z = 121) \land \text{true}\Big) \lor \\ &\Big(((z \neq 120) \land (z \neq 121)) \land \text{true}\Big)\end{aligned}$$

$\equiv \text{false} \lor (z = 121) \lor ((z \neq 120) \land (z \neq 121))$
$\equiv (z = 121) \lor (z \neq 121)$
$\equiv z \neq 121$
Alle Zahlen außer 120 sind möglich bzw. alle Zeichen außer 'x'.



Die Bschlangaul-Sammlung
Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net.Der TEX-Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: `https://github.com/bschlangaul-sammlung/examens-aufgaben-tex/blob/main/Examen/46116/2015/09/Thema-2/Teilaufgabe-1/Aufgabe-3.tex`