

### Aufgabe 3: „Formale Verifikation“

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) Verzweigung:

(b) Mehrfachauswahl:

```
1  switch (z) {
2    case "x":
3      y = "x";
4    case "y":
5      y = --z;
6      break;
7    default:
8      y = 0x39 + "?";
9  }
```

$Q \equiv \text{"x"} = y$

Hinweis zu den ASCII-Codes

- `"x"` = 120<sub>(10)</sub>
- `"y"` = 121<sub>(10)</sub>
- `0x39` = 57<sub>(10)</sub>
- `"?"` = 63<sub>(10)</sub>

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

```
1  if (z == "x") {
2    y = "x";
3    y = z - 1;
4  } else if (z == "y") {
5    y = z - 1;
6  } else {
7    y = 0x39 + "?";
8  }
```

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

```
1  if (z == 120) {
2    y = 120;
3    y = 120 - 1;
```

```
4 } else if (z == 121) {  
5   y = 121 - 1;  
6 } else {  
7   y = 57 + 63;  
8 }
```

Vereinfachung / Zusammenfassung:

```
1 if (z == 120) {  
2   y = 120;  
3   y = 119;  
4 } else if (z == 121) {  
5   y = 120;  
6 } else {  
7   y = 120;  
8 }
```