

Formale Verifikation auf Grundlage der operationellen Semantik

Jan Peleska
Universität Bremen
FB 3 - Informatik
AG Betriebssysteme und verteilte Systeme
jp@informatik.uni-bremen.de

15. Dezember 2008

Zusammenfassung

Anhand von Beispielen wird die formale Verifikation auf Basis der operationellen Semantik für while-Sprachen illustriert. Die Semantik ist beispielsweise in *Gunter Saake, Kai-Uwe Sattler: Algorithmen und Datenstrukturen. dpunkt 2004* definiert.

Invarianten und induktives Schließen: Verifikation von while-Schleifen

Das Beispiel: Ziel unser Verifikation ist der Nachweis, dass folgendes Java Codefragment ein größtes Element des Arrays `a` ermittelt.

```
1      // Deklaration des Arrays a.
2      // M wurde vorher definiert als beliebiger int-Wert >= 0
3      int a[] = new int[M];
4
5      // ... a[] wird mit Werten belegt ...
6      // Jetzt gilt der fuer uns relevante Vorzustand Z_Pre
7      int max = a[0];
8      int i = 1;
9
10     while ( i < a.length ) {
11         if ( max < a[i] ) {
12             max = a[i];
13         }
```

```

14      i = i + 1;
15  }
16  // Nachbedingung POST soll hier gelten.
17  // Der Nachzustand ist mit Z_Post bezeichnet.

```

Die formale Spezifikation: Die Präzisierung der oben informell beschriebenen Spezifikation erfolgt durch die Nachbedingung **POST**:

$$\begin{aligned}
\mathbf{POST} &\equiv_{\text{def}} \mathbf{MAX} \wedge \mathbf{MAXINA} \wedge \mathbf{UNCHGD} \\
\mathbf{MAX} &\equiv_{\text{def}} \forall k \in \{0, \dots, Z_{\text{Post}}(\mathbf{a.length}) - 1\} : Z_{\text{Post}}(\mathbf{a}[k]) \leq Z_{\text{Post}}(\mathbf{max}) \\
\mathbf{MAXINA} &\equiv_{\text{def}} \exists k \in \{0, \dots, Z_{\text{Post}}(\mathbf{a.length}) - 1\} : Z_{\text{Post}}(\mathbf{a}[k]) = Z_{\text{Post}}(\mathbf{max}) \\
\mathbf{UNCHGD} &\equiv_{\text{def}} Z_{\text{Post}}(\mathbf{a.length}) = Z_{\text{Pre}}(\mathbf{a.length}) \wedge \\
&\quad (\forall k \in \{0, \dots, Z_{\text{Post}}(\mathbf{a.length}) - 1\} : Z_{\text{Post}}(\mathbf{a}[k]) = Z_{\text{Pre}}(\mathbf{a}[k]))
\end{aligned}$$

Als Hilfe für den Nachweis werden wir zeigen, dass **INV** eine Invariante der while-Schleife ist:

$$\begin{aligned}
\mathbf{INV} &\equiv_{\text{def}} \mathbf{MAXI} \wedge \mathbf{MAXINAI} \wedge \mathbf{UNCHGDI} \wedge Z(\mathbf{i}) \leq Z(\mathbf{a.length}) \\
\mathbf{MAXI} &\equiv_{\text{def}} \forall k \in \{0, \dots, Z(\mathbf{i}) - 1\} : Z(\mathbf{a}[k]) \leq Z(\mathbf{max}) \\
\mathbf{MAXINAI} &\equiv_{\text{def}} \exists k \in \{0, \dots, Z(\mathbf{i}) - 1\} : Z(\mathbf{a}[k]) = Z(\mathbf{max}) \\
\mathbf{UNCHGDI} &\equiv_{\text{def}} Z(\mathbf{a.length}) = Z_{\text{Pre}}(\mathbf{a.length}) \wedge \\
&\quad (\forall k \in \{0, \dots, Z(\mathbf{a.length}) - 1\} : Z(\mathbf{a}[k]) = Z_{\text{Pre}}(\mathbf{a}[k]))
\end{aligned}$$

Die induktive Verifikationsstrategie: Die formale Verifikation wird jetzt nach dem Induktionsprinzip aufgebaut:

1. Lemma 1 zeigt, dass **INV** vor Erreichen von Zeile 9 mit dem dort gültigen Zustand Z_8 gilt.
2. Lemma 2 zeigt, dass, wenn **INV** bei Eintritt in den while-Block im aktuell vorliegenden Zustand Z gilt, **INV** auch am Ende des while-Blocks (also nach Zeile 13) im dann erreichten Zustand Z' gilt.

Aus Lemma 1 und 2 folgt, dass Erreichen von Zeile 15 im dort vorliegenden Zustand Z_{Post}

$$\mathbf{INV}[Z_{\text{Post}}/Z] \wedge Z_{\text{Post}}(\mathbf{i}) \geq Z_{\text{Post}}(\mathbf{a.length})$$

gilt¹. Hieraus folgern wir in Lemma 3, dass dies die gewünschte Nachbedingung **POST** impliziert.

¹Die Schreibweise $\mathbf{INV}[p/q]$ bedeutet: "Ersetze jedes Vorkommen von q durch p im Ausdruck $\mathbf{INV}$ ".

Die Beweise der Lemmata:

Lemma 1 INV gilt vor *Eintritt* in die *while-Schleife* nach *Ausführung* von Zeile 8 vorliegenden Zustand Z_8 .

Beweis: Auf dem Vorzustand Z_{Pre} wirken die Anweisungen aus Zeile 7 und 8 wie folgt:

$$\begin{aligned} Z_8 = & \\ & \llbracket \text{int max} = \text{a}[0]; \text{int i} = 1 \rrbracket (Z_{\text{Pre}}) = \\ & \llbracket \text{int i} = 1 \rrbracket (\llbracket \text{int max} = \text{a}[0] \rrbracket (Z_{\text{Pre}})) = \\ & \llbracket \text{int i} = 1 \rrbracket (Z_{\text{Pre}} \oplus \{\text{max} \mapsto Z_{\text{Pre}}(\text{a}[0])\}) = \\ & Z_{\text{Pre}} \oplus \{\text{max} \mapsto Z_{\text{Pre}}(\text{a}[0]), \text{i} \mapsto 1\} \end{aligned}$$

Wir betrachten jetzt die Gestalt der Invarianten in Zustand Z_8 :

$$\begin{aligned} \text{INV}[Z_8/Z] &\equiv \text{MAXI}[Z_8/Z] \wedge \text{MAXINAI}[Z_8/Z] \wedge \\ &\quad \text{UNCHGDI}[Z_8/Z] \wedge Z_8(\text{i}) \leq Z_8(\text{a.length}) \\ \text{MAXI}[Z_8/Z] &\equiv \forall k \in \{0, \dots, Z_8(\text{i}) - 1\} : Z_8(\text{a}[k]) \leq Z_8(\text{max}) \\ \text{MAXINAI}[Z_8/Z] &\equiv \exists k \in \{0, \dots, Z_8(\text{i}) - 1\} : Z_8(\text{a}[k]) = Z_8(\text{max}) \\ \text{UNCHGDI}[Z_8/Z] &\equiv Z_8(\text{a.length}) = Z_{\text{Pre}}(\text{a.length}) \wedge \\ &\quad (\forall k \in \{0, \dots, Z_8(\text{a.length}) - 1\} : Z_8(\text{a}[k]) = Z_{\text{Pre}}(\text{a}[k])) \end{aligned}$$

Da nach obiger Berechnung

$$Z_8(\text{i}) = Z_{\text{Pre}} \oplus \{\text{max} \mapsto Z_{\text{Pre}}(\text{a}[0]), \text{i} \mapsto 1\}(\text{i}) = 1$$

und

$$Z_8(\text{max}) = Z_{\text{Pre}} \oplus \{\text{max} \mapsto Z_{\text{Pre}}(\text{a}[0]), \text{i} \mapsto 1\}(\text{i}) = Z_{\text{Pre}}(\text{a}[0])$$

und

$$Z_8(\text{a}[0]) = Z_{\text{Pre}} \oplus \{\text{max} \mapsto Z_{\text{Pre}}(\text{a}[0]), \text{i} \mapsto 1\}(\text{i}) = Z_{\text{Pre}}(\text{a}[0])$$

und

$$Z_8(\text{a.length}) = Z_{\text{Pre}} \oplus \{\text{max} \mapsto Z_{\text{Pre}}(\text{a}[0]), \text{i} \mapsto 1\}(\text{a.length}) = Z_{\text{Pre}}(\text{a.length})$$

gilt, können wir die Invariante in Zustand Z_8 zu

$$\begin{aligned} \text{INV}[Z_8/Z] &\equiv \text{MAXI}[Z_8/Z] \wedge \text{MAXINAI}[Z_8/Z] \wedge \\ &\quad \text{UNCHGDI}[Z_8/Z] \wedge 1 \leq Z_8(\text{a.length}) \\ \text{MAXI}[Z_8/Z] &\equiv \forall k \in \{0\} : Z_8(\text{a}[k]) \leq Z_{\text{Pre}}(\text{a}[0]) \\ \text{MAXINAI}[Z_8/Z] &\equiv \exists k \in \{0\} : Z_8(\text{a}[k]) = Z_{\text{Pre}}(\text{a}[0]) \\ \text{UNCHGDI}[Z_8/Z] &\equiv Z_8(\text{a.length}) = Z_{\text{Pre}}(\text{a.length}) \wedge \\ &\quad (\forall k \in \{0, \dots, Z_8(\text{a.length}) - 1\} : Z_8(\text{a}[k]) = Z_{\text{Pre}}(\text{a}[k])) \end{aligned}$$

und dann weiter zu

$$\begin{aligned}
\text{INV}[Z_8/Z] &\equiv \text{MAXI}[Z_8/Z] \wedge \text{MAXINAI}[Z_8/Z] \wedge \\
&\quad \text{UNCHGDI}[Z_8/Z] \wedge 1 \leq Z_{\text{Pre}}(\text{a.length}) \\
\text{MAXI}[Z_8/Z] &\equiv Z_{\text{Pre}}(\text{a}[0]) \leq Z_{\text{Pre}}(\text{a}[0]) \\
\text{MAXINAI}[Z_8/Z] &\equiv Z_{\text{Pre}}(\text{a}[0]) = Z_{\text{Pre}}(\text{a}[0]) \\
\text{UNCHGDI}[Z_8/Z] &\equiv Z_{\text{Pre}}(\text{a.length}) = Z_{\text{Pre}}(\text{a.length}) \wedge \\
&\quad (\forall k \in \{0, \dots, Z_8(\text{a.length}) - 1\} : Z_8(\text{a}[k]) = Z_{\text{Pre}}(\text{a}[k]))
\end{aligned}$$

vereinfachen, und letzteres Prädikat ist wahr, weil der Array in den Anweisungen nicht verändert wurde. Damit ist Lemma 1 bewiesen. $\square$

Lemma 2 *Gilt INV beim Eintritt in die while-Schleife mit Valuation Z, so gilt INV auch nach Ausführung des while-Blocks in der dann vorliegenden Valuation Z'.*

Beweis: Wir berechnen zunächst, wie sich Zustand Z' aus Z ergibt und wenden hierzu die semantische Regel für die if-Bedingung und die sequenzielle Komposition an:

$$\begin{aligned}
Z' &= \llbracket \text{if}(\text{max} < \text{a}[i]) \{ \text{max} = \text{a}[i]; \} i = i + 1 \rrbracket (Z) \\
&= \begin{cases} \llbracket i = i + 1 \rrbracket (\llbracket \text{max} = \text{a}[i] \rrbracket (Z)) & \text{falls } Z(\text{max}) < Z(\text{a}[Z(i)]) \\ \llbracket i = i + 1 \rrbracket (Z) & \text{sonst} \end{cases} \\
&= \begin{cases} \llbracket i = i + 1 \rrbracket (Z \oplus \{ \text{max} \mapsto Z(\text{a}[Z(i)]) \}) & \text{falls } Z(\text{max}) < Z(\text{a}[Z(i)]) \\ \llbracket i = i + 1 \rrbracket (Z) & \text{sonst} \end{cases} \\
&= \begin{cases} Z \oplus \{ \text{max} \mapsto Z(\text{a}[Z(i)]), i \mapsto Z(i) + 1 \} & \text{falls } Z(\text{max}) < Z(\text{a}[Z(i)]) \\ Z \oplus \{ i \mapsto Z(i) + 1 \} & \text{sonst} \end{cases}
\end{aligned}$$

Jetzt wird gezeigt, dass $\text{INV}[Z'/Z]$ gilt; und wegen der Fallunterscheidung in der Berechnung von Z' machen wir bei diesem Nachweis dieselbe Fallunterscheidung.

Fall 1: $Z(\text{max}) < Z(\text{a}[Z(i)])$. Wir berechnen

$$\begin{aligned}
\text{INV}[Z'/Z] &\equiv \text{MAXI}[Z'/Z] \wedge \text{MAXINAI}[Z'/Z] \wedge \\
&\quad \text{UNCHGDI}[Z'/Z] \wedge Z'(i) \leq Z'(\text{a.length}) \\
\text{MAXI}[Z'/Z] &\equiv \forall k \in \{0, \dots, Z'(i) - 1\} : Z'(\text{a}[k]) \leq Z'(\text{max}) \\
\text{MAXINAI}[Z'/Z] &\equiv \exists k \in \{0, \dots, Z'(i) - 1\} : Z'(\text{a}[k]) = Z'(\text{max}) \\
\text{UNCHGDI}[Z'/Z] &\equiv Z'(\text{a.length}) = Z_{\text{Pre}}(\text{a.length}) \wedge \\
&\quad (\forall k \in \{0, \dots, Z'(\text{a.length}) - 1\} : Z'(\text{a}[k]) = Z_{\text{Pre}}(\text{a}[k]))
\end{aligned}$$

Aus obiger Berechnung von Z' folgt, dass $Z'(i) = Z(i) + 1$. Wir können also

$$\mathbf{INV}[Z'/Z] \equiv \mathbf{INV}[Z'/Z][(Z(i) + 1)/Z'(i)]$$

folgern, und erhalten damit

$$\begin{aligned} \mathbf{INV}[Z'/Z] &\equiv \mathbf{MAXI}[Z'/Z] \wedge \mathbf{MAXINAI}[Z'/Z] \wedge \\ &\quad \mathbf{UNCHGDI}[Z'/Z] \wedge Z(i) + 1 \leq Z'(\mathbf{a.length}) \\ \mathbf{MAXI}[Z'/Z] &\equiv \forall k \in \{0, \dots, Z(i)\} : Z'(\mathbf{a}[k]) \leq Z'(\mathbf{max}) \\ \mathbf{MAXINAI}[Z'/Z] &\equiv \exists k \in \{0, \dots, Z(i)\} : Z'(\mathbf{a}[k]) = Z'(\mathbf{max}) \\ \mathbf{UNCHGDI}[Z'/Z] &\equiv Z'(\mathbf{a.length}) = Z_{\text{Pre}}(\mathbf{a.length}) \wedge \\ &\quad (\forall k \in \{0, \dots, Z(\mathbf{a.length}) - 1\} : Z'(\mathbf{a}[k]) = Z_{\text{Pre}}(\mathbf{a}[k])) \end{aligned}$$

Weiterhin folgt aus der Berechnung von Z' , dass Array $\mathbf{a}$ und seine Länge immer noch dieselben Werte wie im Zustand Z haben. Dies resultiert in

$$\begin{aligned} \mathbf{INV}[Z'/Z] &\equiv \mathbf{INV}[Z'/Z][(Z(i) + 1)/Z'(i)] \equiv \\ &\quad \mathbf{INV}[Z'/Z][(Z(i) + 1)/Z'(i)][Z(\mathbf{a}[k])/Z'(\mathbf{a}[k])][Z(\mathbf{a.length})/Z'(\mathbf{a.length})] \end{aligned}$$

Damit stellt sich die Invariante im Zustand Z' folgendermaßen dar:

$$\begin{aligned} \mathbf{INV}[Z'/Z] &\equiv \mathbf{MAXI}[Z'/Z] \wedge \mathbf{MAXINAI}[Z'/Z] \wedge \\ &\quad \mathbf{UNCHGDI}[Z'/Z] \wedge Z(i) + 1 \leq Z(\mathbf{a.length}) \\ \mathbf{MAXI}[Z'/Z] &\equiv \forall k \in \{0, \dots, Z(i)\} : Z(\mathbf{a}[k]) \leq Z'(\mathbf{max}) \\ \mathbf{MAXINAI}[Z'/Z] &\equiv \exists k \in \{0, \dots, Z(i)\} : Z(\mathbf{a}[k]) = Z'(\mathbf{max}) \\ \mathbf{UNCHGDI}[Z'/Z] &\equiv Z(\mathbf{a.length}) = Z_{\text{Pre}}(\mathbf{a.length}) \wedge \\ &\quad (\forall k \in \{0, \dots, Z(\mathbf{a.length}) - 1\} : Z(\mathbf{a}[k]) = Z_{\text{Pre}}(\mathbf{a}[k])) \end{aligned}$$

Schließlich lesen wir aus der Berechnung von Z' ab, dass $Z'(\mathbf{max}) = Z(\mathbf{a}[Z(i)])$ und die Ersetzung $[Z(\mathbf{a}[Z(i)])/Z'(\mathbf{max})]$ in $\mathbf{INV}[Z'/Z]$ führt auf

$$\begin{aligned} \mathbf{INV}[Z'/Z] &\equiv \mathbf{MAXI}[Z'/Z] \wedge \mathbf{MAXINAI}[Z'/Z] \wedge \\ &\quad \mathbf{UNCHGDI}[Z'/Z] \wedge Z(i) + 1 \leq Z(\mathbf{a.length}) \\ \mathbf{MAXI}[Z'/Z] &\equiv \forall k \in \{0, \dots, Z(i)\} : Z(\mathbf{a}[k]) \leq Z(\mathbf{a}[Z(i)]) \\ \mathbf{MAXINAI}[Z'/Z] &\equiv \exists k \in \{0, \dots, Z(i)\} : Z(\mathbf{a}[k]) = Z(\mathbf{a}[Z(i)]) \\ \mathbf{UNCHGDI}[Z'/Z] &\equiv Z(\mathbf{a.length}) = Z_{\text{Pre}}(\mathbf{a.length}) \wedge \\ &\quad (\forall k \in \{0, \dots, Z(\mathbf{a.length}) - 1\} : Z(\mathbf{a}[k]) = Z_{\text{Pre}}(\mathbf{a}[k])) \end{aligned}$$

Jetzt sind wir in der Lage, die Gültigkeit eines jeden Konjunktes zu zeigen:

- $Z(i) + 1 \leq Z(\mathbf{a.length})$ gilt, weil die `while`-Schleife durchlaufen wird, also zu Beginn des Durchlaufs die Schleifenbedingung $Z(i) < Z(\mathbf{a.length})$ galt.

- Um $\text{MAXI}[Z'/Z]$ zu zeigen, zerlegen wir die Allquantifikation in

$$\begin{aligned} \text{MAXI}[Z'/Z] &\equiv (\forall k \in \{0, \dots, Z(i) - 1\} : Z(a[k]) \leq Z(a[Z(i)])) \\ &\quad \wedge Z(a[Z(i)]) \leq Z(a[Z(i)]) \end{aligned}$$

Da die **INV** im Zustand Z gültig war, folgt $\forall k \in \{0, \dots, Z(i) - 1\} : Z(a[k]) \leq Z(\text{max})$. Für den aktuellen Fall 1 gilt $Z(\text{max}) < Z(a[Z(i)])$, also folgt $\forall k \in \{0, \dots, Z(i) - 1\} : Z(a[k]) \leq Z(a[Z(i)])$. Damit ist $\text{MAXI}[Z'/Z]$ gültig.

- $\text{MAXINAI}[Z'/Z]$ gilt trivialerweise mit $k = Z(i)$.
- $\text{UNCHGDI}[Z'/Z]$ gilt, weil die Bedingung in Z gültig war und das Array beim Übergang zu Z' nicht verändert wurde.

Fall 2: $Z(\text{max}) \geq Z(a[Z(i)])$. In diesem Fall dürfen wir in $\text{INV}[Z'/Z]$ zuerst $Z(i) + 1$ für $Z'(i)$ einsetzen. Weiterhin bemerken wir, dass Z' ansonsten gegenüber Z unverändert ist. Folglich hat $\text{INV}[Z'/Z]$ die Form

$$\begin{aligned} \text{INV}[Z'/Z] &\equiv \text{MAXI}[Z'/Z] \wedge \text{MAXINAI}[Z'/Z] \wedge \\ &\quad \text{UNCHGDI}[Z'/Z] \wedge Z(i) + 1 \leq Z(a.\text{length}) \\ \text{MAXI}[Z'/Z] &\equiv \forall k \in \{0, \dots, Z(i)\} : Z(a[k]) \leq Z(\text{max}) \\ \text{MAXINAI}[Z'/Z] &\equiv \exists k \in \{0, \dots, Z(i)\} : Z(a[k]) = Z(\text{max}) \\ \text{UNCHGDI}[Z'/Z] &\equiv Z(a.\text{length}) = Z_{\text{pre}}(a.\text{length}) \wedge \\ &\quad (\forall k \in \{0, \dots, Z(a.\text{length}) - 1\} : Z(a[k]) = Z_{\text{pre}}(a[k])) \end{aligned}$$

- $Z(i) + 1 \leq Z(a.\text{length})$ gilt mit demselben Argument wie in Fall 1.
- $\text{MAXI}[Z'/Z]$ gilt, weil es in Z bereits gültig war und im Fall 2 $Z(\text{max}) \geq Z(a[Z(i)])$ gültig ist.
- $\text{MAXINAI}[Z'/Z]$ gilt mit demselben $k \in \{0, \dots, Z(i) - 1\}$ wie im Zustand Z vor Beginn des Schleifendurchlaufs.
- $\text{UNCHGDI}[Z'/Z]$ gilt nach Voraussetzung über die Gültigkeit von **INV** in Z .

Damit ist das Lemma vollständig bewiesen. $\square$

Lemma 3 *Aus Gültigkeit der Invariante und aus der Schleifenbedingung folgt, dass die Nachbedingung **POST** eingehalten wird, sobald die Schleife terminiert.*

Beweis: Wenn die Schleife terminiert (oder gar nicht erst durchlaufen wird), gilt im resultierenden Zustand Z_{Post}

- $Z_{\text{Post}}(i) \geq Z_{\text{Post}}(a.\text{length})$, denn dies ist die Negation der Schleifenbedingung, und
- $\text{INV}[Z_{\text{Post}}/Z]$, denn nach Lemma 1 und 2 haben wir es mit einer Invarianten zu tun.

Wegen $\text{INV}[Z_{\text{Post}}/Z]$ gilt auch $Z_{\text{Post}}(i) \leq Z_{\text{Post}}(a.\text{length})$, es folgt also $Z_{\text{Post}}(i) = Z_{\text{Post}}(a.\text{length})$. Damit gilt also

$$\text{INV}[Z_{\text{Post}}/Z][Z_{\text{Post}}(a.\text{length})/Z_{\text{Post}}(i)]$$

und dies stellt sich dar als

$$\begin{aligned} \text{INV}[Z_{\text{Post}}/Z][Z_{\text{Post}}(a.\text{length})/Z_{\text{Post}}(i)] &\equiv \\ (\forall k \in \{0, \dots, Z_{\text{Post}}(a.\text{length}) - 1\} : Z_{\text{Post}}(a[k]) \leq Z_{\text{Post}}(\text{max})) \wedge \\ (\exists k \in \{0, \dots, Z_{\text{Post}}(a.\text{length}) - 1\} : Z_{\text{Post}}(a[k]) = Z_{\text{Post}}(\text{max})) \wedge \\ Z_{\text{Post}}(a.\text{length}) = Z_{\text{Pre}}(a.\text{length}) \wedge \\ (\forall k \in \{0, \dots, Z_{\text{Post}}(a.\text{length}) - 1\} : Z_{\text{Post}}(a[k]) = Z_{\text{Pre}}(a[k])) \\ &\equiv \text{POST} \end{aligned}$$

Damit ist die Behauptung gezeigt. □

Terminierung: Es ist zu beachten, dass die verifizierte Aussage nur unter der Bedingung gilt, dass die Schleife auch tatsächlich terminiert (sogenannte **partielle Korrektheit**). Die **totale Korrektheit** erfordert daher noch den zusätzlichen Nachweis, dass die Terminierung der obigen while-Schleife auch tatsächlich immer gesichert ist. Hierzu muss eine weitere Beweisstrategie eingesetzt werden. Das Strategieprinzip bei gegebener while-Schleife

```

1   while ( b(x_1, ..., x_n) ) {
2       B
3   }
```

lautet

1. Finde eine Variable $z \in \{x_1, \dots, x_n\}$ so dass für jeden beliebigen Zustand Z und eine Konstante c gilt

$$Z(z) > c \Rightarrow \neg b(Z(x_1), \dots, Z(x_n))$$

2. Zeige, dass wenn Z den Vorzustand von B zu Beginn eines Schleifendurchlaufs bezeichnet, der Nachzustand Z' nach Ausführung von B

$$Z'(z) = (\llbracket B \rrbracket(Z))(z) \geq Z(z) + \varepsilon$$

für eine Konstante $\varepsilon > 0$ erfüllt.

Wenn z eine ganzzahlige Variable ist, hat ε den Wert 1. Bei Gleitkommavariablen z garantiert die Existenz von ε , dass nach endlichen vielen Schleifendurchläufen mit wiederholter Ausführung von B tatsächlich $Z(z) > c$ erfüllt ist und die Schleife folglich terminieren muss.

In unserem Beispielalgorithmus wählen wir die Variable i für den Terminierungsbeweis, denn offensichtlich gilt

$$Z(i) > Z(a.length) \Rightarrow \neg(Z(i) < Z(a.length))$$

Die Konstante c ist also hier durch $c = Z(a.length)$ gegeben (aus der Postcondition **UNCHGD** wissen wir, dass $Z(a.length)$ tatsächlich konstant ist). Der zweite Schritt des Terminierungsbeweises folgt jetzt aus der Ableitung (B bezeichnet den gesamten while-Block, Zeilen 11 — 14 und IB den gesamten if-Block, Zeilen 11 — 13)

$$\begin{aligned} Z'(i) &= (\llbracket B \rrbracket(Z))(i) \\ &= (\llbracket i = i + 1 \rrbracket(\llbracket IB \rrbracket(Z)))(i) \\ &= (\llbracket i = i + 1 \rrbracket(Z))(i) \\ &= (Z \oplus \{i \mapsto Z(i) + 1\})(i) \\ &= Z(i) + 1 \end{aligned}$$

Dabei haben wir ausgenutzt, dass $(\llbracket IB \rrbracket(Z))(i) = Z(i)$, weil der if-Block IB keine Zuweisung auf i enthält.