Algorithmik Blatt 1 Aufgabe 2

Mtr.-Nr. 6329857

Universität Hamburg — 19. Oktober 2019
Zusammengearbeitet mit 7330980

Modulo Schleifeninvariante

Als Schleifeninvariante gilt $\big(r = x - q \cdot y\big) \wedge \big(q \geq 0\big)$.

Initialisierung

1. $q = 0, r = x \implies \big(x = x - 0 \cdot y\big) \wedge \big(q \geq 0\big)$

Fortsetzung

1. Angenommen zu Beginn einer Iteration gilt $r = x - q \cdot y$
2. Die Schleife setzt $r' = r - y$ und $q' = q + 1$
3. Umgeformt: $r = r' + y$ und $q = q' - 1$
4. Die Schleife ändert weder x noch y, also $x' = x$ und $y' = y$
5. Einsetzen in Annahme: $r' + y' = x' - \big(q' - 1\big) \cdot y' = x' - q' \cdot y' + y'$
6. Also gilt $r' = x' - q' \cdot y'$ und $q + 1 = q' \geq 0$ auch am Ende der Iteration.

Terminierung

Wenn die Schleife endet, ist $r < y$ und $x = q \cdot y + r$. Damit entspricht r genau der Definition von x mod y. Die Schleife terminiert wenn $y > 0$, weil r in jeder Iteration um y verringert wird.

Primzahl Schleifeninvarianz

Als Schleifeninvarianten gelten:

a) Bis i^2 sind nur Primzahlen markiert: $\forall p : \big(0 < p < i^2\big) \rightarrow \big(P[p] \rightarrow \forall z : (z \leq n) \rightarrow \neg P[p \cdot z]\big)$

b) Alle Primzahlen im bis i^2 sind markiert: $\forall p : \big(0 < p < i^2\big) \rightarrow \big(\text{isPrime}\big(p\big) \rightarrow P[p]\big)$

Sie ließen sich auch in einer gemeinsamen Formel ausdrücken, allerdings ist es übersichtlicher sie getrennt zu zeigen

Initialisierung (a)

Zu beginn ist $P[1] = \texttt{false}$ und $i = 2$. Als einziges p ist 1 zu prüfen. Das Antezedens des inneren Konditionals ($P[1]$) ist falsch, also gilt die Invariante.

Initialisierung gb)

Da das Array mit `true` initialisiert wird, gilt das Konditional in Invariante b zu Beginn für alle p.

Fortsetzung (a)

Die Schleife erhöht i in jedem Durchlauf um 1. Die Invariante muss also nach jedem Durchlauf für i weitere p gelten sowie auch für alle niedrigeren p, für die sie schon im Vorherigen Durchlauf galt. Bis auf das Inkrement wird i nicht verändert.

Die Invariante besteht im Kern aus einem Konditional. Sie könnte also nur dadurch verletzt werden, dass das Antezedens wahr, aber das Konsequenz falsch ist. Um die Invariante zu verletzen, müsste also gelten, dass es ein $p < i^2$ und ein $z \leq n$ gibt für die gilt $P[p] \wedge P[p \cdot z]$.

In jedem Durchlauf der äußeren Schleife wird aber nur $P[k], k \geq i^2$ beschrieben. Der linke Operand kann also durch die Operationen der Schleife nicht wahr werden. Außerdem wird das Array nur mit `false` Werten beschrieben, also kann auch der rechte Operand nicht wahr werden. Es ist also nicht möglich eine Belegung zu finden, die die Invariante verletzt.

Fortsetzung (b)

Der Algorithmus beschreibt nur Felder des Arrays, deren Index ein vielfaches (> 1) von i sind. Also wird das Konsequenz des Konditionals von Invariante b niemals mit $false$ belegt.

Terminierung (a)

Die Schleife terminiert mit $i^2 > n$, also gilt $\forall p : \big(0 < p < n\big) \rightarrow \big(P[p] -> \forall z : (z \leq n) \rightarrow \big(\neg P[p \cdot z]\big)\big)$. Der Algorithmus hat also alle Felder mit `false` belegt, deren Index ein vielfaches eines Index B ist, dessen zugehöriges Feld $P[B] = \texttt{true}$ ist.

Terminierung (b)

Da im Verlauf nur Felder mit ganzzahlig-vielfachen Indizes beschrieben wurden, sind alle Primzahlfelder immer noch mit der initialen Belegung von `true` belegt.

Terminierung

Also sind genau die Primzahlfelder im Array mit `true` und alle anderen mit `false` markiert.