Fakultät IV - Elektrotechnik und Informatik
Einführung in die Programmierung WS 2015/16
Feldmann / Semmler / Lichtblau / Streibelt / Pujol / Rost

Zusatzmaterial Korrektheitsbeweis Selection Sort

letzte Aktualisierung: 05. Januar, 22:15 Uhr

(45c9527d03b571f500a883392864e4fa4d739632)

Ausgabe: Mittwoch, 06.01.2016
Autoren: Johannes Wortmann, Matthias Rost, Niklas Semmler

Korrektheitsbeweis Selection Sort

Gegeben ist der Pseudocode des Selection Sort Algorithmus:

```
1  SelectionSort(Array A)
2      for i←1 to length(A)−1 do
3          min ←i
4          for j←i+1 to length(A) do
5              if A[j] < A[min] then
6                  min←j
7          tmp ← A[i]
8          A[i] ← A[min]
9          A[min] ← tmp
10     return A
```

Schleifeninvarianten

Für beide Schleifen wird jeweils eine Schleifeninvariante benötigt.

A-1 Für die äußere Schleife (Zeilen 2-9) gilt:

- **(a)** $A[1, \dots, i]$ ist aufsteigend sortiert sowie
- **(b)** $(i > 1 \Rightarrow (A[i-1] \leq A[k]))$ für alle $k \in \{i, i+1, \dots, \mathrm{length}(A)\}$.

B-1 Für die innere Schleife (Zeile 4-6) gilt:

- **(a)** $\min \geq i \wedge \min \leq \mathrm{length}(A)$ sowie
- **(b)** $A[\min] \leq A[k]$ für alle $k \in \{i, i+1, \dots, j-1\}$.

Beweis der inneren Schleifeninvariante

Die Aussage (a) gilt trivialerweise, da $\min$ entweder i oder j zugewiesen wird und diese Variablen auf natürliche Weise durch die Grenzen der for-Schleifen begrenzt sind. Wir wenden uns daher ausschließlich dem Beweis von (b) zu.

B-2: Initiale Gültigkeit

Beim Eintritt in die Schleife gilt $\min = i$ (Zeile 3) und $1 \leq i \leq \mathrm{length}(A) - 1$ (Zeile 2). Nach der Initialisierung von $j = i+1$ gilt die Aussage (b) der inneren Schleifeninvariante: es ist $A[\underbrace{\min}_{=i=1}] \leq A[k]$ für alle $k \in \{\underbrace{i}_{=1}, \dots, \underbrace{j-1}_{=i=1}\} = \{1\}$.

B-3: Erhaltung der Gültigkeit

Wir nehmen an, dass die Schleifeninvariante für ein festes $j' \in \{i+1, \dots, \mathrm{length}(A)\}$ in der Zeile 5 gilt. Sei $\min'$ der Wert, den die Variable $\min$ vor der Ausführung von Zeile 5 angenommen hat. Wir benutzen die folgende Fallunterscheidung:

1.Fall: $A[j'] \geq A[\min']$

- Gemäß der Annahme, dass die Schleifeninvariante für j' gilt, wissen wir, dass $A[\min'] \leq A[k]$ für alle $k \in \{i, i+1, \dots j'-1\}$ gilt.

- Gemäß des aktuellen Falles in der Fallunterscheidung gilt $A[j'] \geq A[\min']$ und Zeile 6 wird nicht ausgeführt. Somit gilt $A[\min] \leq A[k]$ für alle $k \in \{i, i+1, \ldots, \underbrace{(j'+1)-1}_{=j'}\}$.
- Dies beweist, dass die Aussage auch für $j'+1$ im Falle $A[\min] \leq A[j]$ gilt. Die Invariante bleibt somit nach Inkrementierung der Variable j erhalten.

2.Fall: $A[j'] < A[\min']$

- In diesem Fall wird in Zeile 6 $\min = j'$ gesetzt.
- Gemäß der Annahme, dass die Schleifeninvariante für j' gegolten hat, folgt $A[\min'] \leq A[k]$ für alle $k \in \{i, i+1, \ldots, j-1\}$.
- Im betrachteten Fall folgt mit $A[j'] < A[\min']$ nun, dass $A[j'] = A[\min] \leq A[k]$ für alle $k \in \{i, i+1, \ldots, \underbrace{(j'+1)-1}_{=j'}\}$ gilt.
- Dies beweist, dass die Aussage auch für $j'+1$ im Falle $A[j'] < A[\min']$ gilt.

Da obige Fallunterscheidung vollständig ist – also alle Fälle abdeckt – gilt die Schleifeninvariante zu jedem Zeitpunkt, insbesondere beim letzten Aufruf für $j = \text{length}(A) + 1$.

Beweis der äußeren Schleifeninvariante

A-2: Initiale Gültigkeit

Beim erstmaligen Eintritt in die Schleife – also für $i = 1$ – gilt (a), da $A[1, \ldots, 1]$ ein einelementiges Array ist und somit bereits aufsteigend sortiert ist. Weiterhin gilt (b), da die Prämisse ($i > 1$) nicht erfüllt ist.

A-3: Erhaltung der Gültigkeit

Wir nehmen an, dass die (äußere) Schleifeninvariante zu Beginn der i'-ten Iteration für ein festes $i' \in \{1, 2, \ldots, \text{length}(A) - 1\}$ gilt. Es bleibt zu zeigen, dass die Schleifeninvariante auch für $i' + 1$ gilt. Gemäß der Gültigkeit der inneren Schleifeninvariante gilt $A[\min] \leq A[k]$ für alle $k \in \{i', i'+1, \ldots, \text{length}(A)\}$ nach Ausführung der inneren Schleife. Dies folgt daraus, dass die innere Schleife bei $j = \text{length}(A) + 1$ verlassen wird und der Aussage (b) der inneren Schleifeninvariante für diesen spezifischen Wert von j.
In Zeile 7-9 wird der Wert von $A[\min]$ mit dem Wert von $A[i']$ getauscht. Im Folgenden betrachten wir den Zustand nach diesem Tausch.

Bzgl. (b) Da $A[i']$ den Wert von $A[\min]$ annimmt, und $A[\min]$ vor dem Tausch kleiner als alle Elemente $A[k]$ für $k \in \{i', i'+1, \ldots, \text{length}(A)\}$ war, gilt $A[(i'+1)-1] \leq A[k]$ insbesonders für $k \in \{i'+1, i'+2, \ldots, \text{length}(A)\}$. Somit gilt die Aussage (b) auch für $i'+1$.

Bzgl. (a) Sofern $i' = 1$ gilt, so nimmt gemäß obiger Argumentation $A[i']$ den Wert des Minimums des gesamten Arrays A an, da $A[\min] \leq A[k]$ für alle $k \in \{\underbrace{i'}_{=1}, \ldots, \text{length}(A)\}$ gilt und $A[\min]$ und $A[\underbrace{i'}_{=1}]$ getauscht werden. Somit sind die Elemente $A[1, \ldots, 2]$ sicherlich sortiert und die Aussage (a) gilt.
Andererseits folgt aus der Gültigkeit der äußeren Schleifeninvariante, sofern $i' > 1$, gilt folgendes: $A[i'-1] \leq A[k]$ für $k \in \{i', i'+1, \ldots, \text{length}(A)\}$. Da $i' \leq \min \leq \text{length}(A)$ gilt, und der Wert von $A[i']$ und $A[\min]$ getauscht werden, gilt $A[i'-1] \leq A[i']$. Da $A[1, \ldots, i'-1]$ gemäß der Annahme der Gültigkeit der äußeren Schleifeninvariante bereits sortiert ist, ist also auch $A[1, \ldots, i'-1, i']$ sortiert.

Beweis der Korrektheit (C-1)

Sollte $\text{length}(A) < 2$ sein, wird die äußere for-Schleife nicht ausgeführt. In diesen Fällen ist das Array A entweder leer – d.h. hat kein Element – oder es besteht nur aus einem Element. Das Array ist somit per Definition schon sortiert.
Betrachten wir nun, den Fall dass $\text{length}(A) \geq 2$ gilt. Wir betrachten den Zustand des Arrays A nach dem Verlassen der äußeren Schleife. Die äußere Schleife wird bei dem Wert $i = length(A)$ verlassen. Die äußere Schleifeninvariante besagt für $i = length(A)$, dass $A[1, \ldots, \text{length}(A)]$ aufsteigend sortiert ist, was zu beweisen war.