

Übungen zur Vorlesung
„Grundlagen der Programm- und Systementwicklung“

Aufgabe 12.1 Programmentwicklung vermittelt Zusicherungen

In Situationen, die zu einem zu entwickelnden Programm auch einen Korrektheitsbeweis verlangen, sollte beides, Programm und Beweis, simultan entwickelt werden. Möglicherweise stellt man sogar das Beweisen in den Fordergrund: Man beginnt mit einer Zielaussage über das gesuchte Programm und detailliert dieses dann schrittweise nach den Beweiserfordernissen. Dies funktioniert auch für die Zusicherungsmethode. E. W. Dijkstra hat wiederholt demonstriert, daß Algorithmen auf diese Weise verständlich gemacht werden können. Als Beispiel behandelte er auch *Batcher's Baffler*, einen verzwickten Sortieralgorithmus, den K. E. Batcher 1968 entworfen hat:

Ein Feld soll dadurch sortiert werden, daß Paare von Elementen geordnet werden. Die Grundidee dabei ist, Gruppen von disjunkten Paaren parallel zu behandeln.

Um uns die Beachtung der oberen Feldgrenze zu vereinfachen, nehmen wir an, daß das Feld

```
var [0:N-1] array nat f
```

über die Feldgrenze N (einschließlich) ins Unendliche mit Werten ∞ fortgesetzt ist. Wir verwenden die Funktion

```
fact OK = (nat i, j) bool: f[i] ≤ f[j].
```

Das Feld soll einzig verändert werden durch die Prozedur

```
proc Ord = (nat i, j):  
  if OK(i, j) then nop else swap(f, i, j) fi ,
```

wobei *swap* die Werte von $f[i]$ und $f[j]$ vertauscht.

Man beachte, daß wegen unserer Feldkomplettierung

(*) $\forall i: OK(i, i+t)$

gilt für $N \leq t$. Geht man also von einem hinreichend großen Wert von t aus und verkleinert diesen schrittweise durch einen Abschnitt S – unter Invarianz der Eigenschaft (*) – bis zum Wert 1, so kommt man offenbar zu einer Lösung der Aufgabe. Denn nach der while-Regel und der Abschwächungsregel folgt

$$\frac{\{t \neq 1 \wedge \forall i: OK(i, i+t)\} \quad S \quad \{\forall i: OK(i, i+t)\}}{\{\forall i: OK(i, i+t)\} \text{ while } t \neq 1 \text{ do } S \text{ od } \{t = 1 \wedge \forall i: OK(i, i+t)\}}$$
$$\frac{\{\forall i: OK(i, i+t)\} \text{ while } t \neq 1 \text{ do } S \text{ od } \{\forall i: OK(i, i+t)\}}{\{\forall i: OK(i, i+t)\} \text{ while } t \neq 1 \text{ do } S \text{ od } \{\forall i: OK(i, i+1)\}}$$

Batcher's Vorschlag lautet:

Man verkleinere t durch Halbieren, starte also mit einer hinreichend großen Zweierpotenz. und Partitioniere die Indexwerte durch

```
fact e = (nat i) bool: (i mod 2*t) < t
```

Dann kann man jedenfalls parallel behandeln:

```
||i if e(i) then Ord(i, i+t) else nop fi
```

Entwickeln Sie hieraus den Beweis und die Spezifikation von Batcher's Baffler. Die Frage ist insbesondere, was für diejenigen Elemente mit $\neg e(i)$ in der while-Schleife zu tun ist.