ALP4 SoSe 2013, Di. 16-18

Lösung Übungsblatt 2

Christoph van Heteren-Frese (Matr.-Nr.: 4465677),
Sven Wildermann (Matr.-Nr.: 4567553)
Tutor: Alexander Steen, eingereicht am 25. April 2013

Aufgabe 1

a)

Bei diese Implementierung ist es möglich, dass zunächst $a[0] = a[1] = false$ gilt (das Prinzip der Unteilbarkeit der Abfrage eines Zustandes und seiner Veränderung ist nicht eingehalten [vgl. 1, S. 39]). Im anshcließenden Schleifendurchlauf kann durch die gleiche Ablaufreihenfolge $a[0] = a[1] = true$ gelten, wodurch beide Prozesse gleichzeitig ihren kritischen Bereich betreten würden. Zwar ist es warscheinlich, dass sich bei einem der nächsten Schleifendurchläufe eine andere Situation einstellt, aber grundsätlich ist so eine Implementierung zu verwerfen.

b)

Wenn beide Prozesse den kritischen Abschnit betreten möchten, wird einer dem andren immer den Vortitt lassen.

c)

Aufgabe 2

a)

Gegenseitiger Ausschluss:

Behauptung: Der Algorithmus von Dekker erfüllt den wechselseitigen Ausschluss.
Beweis: Angenommen, die Aussage ist falsch. Dann gibt es eine Ablaufreihenfolge bei der sich beide Prozesse in ihrem kritischen Abschnitt befinden. Daraus folgt, dass beide Prozesse die Austrittsbedingung der äußeren for-Schleife erfüllt haben. Drei Fälle können unterschieden werden, die dazu führen könnten:

1. **Keiner der Beiden Prozesse läuft durch den Schleifenkörper.** Beiden Prozesse müssten dann die jeweilige Schleifenabbruchbedingung von Anfang an erfüllen ($\neg i_1$ für $P0$ und $\neg i_0$ für $P1$). Das ist aber nicht möglich. Sei $P1$ der zweite Prozess, der die for-Bedingung prüft. Dann hat $P0$ vorher durch die Anweisung `interested[p]` dafür gesorgt, dass $P1$ in die Schleife eintreten muss. Aus Symetriegründen gilt dies auch für die andere Reihenfolge.

2. **Beide Prozess durchlaufen den Schleifenkörper.** Dann muss einer der beiden Prozesse in der inneren *for*-Schleife hängen bleiben, da favoured erst nach Durchlaufen des kritischen Abschnitts (durch *Unlock*) geändert wird.

3. **Einer der Prozesse hat den Schleifenkörper durchlaufen während sich der andere bereits im kritischen Abschnitt befindet.** Dass geht auch nicht, denn Dann lässt sich leicht nachprüfen, dass er den kritischen Abschnitt nicht mehr betreten kann.

Diese Fälle zeigen, dass die Annahme falsch ist. Somit garantiert der Dekker-Algorithmus den wechselseitigen Ausschluss. □

b)

Aufgabe 3

Aufgabe 4

a)

b)

Literatur

[1] Christian Maurer. *Nichtsequentille Programmierung mit Go 1 Kompakt.* Springer Vieweg, 2012. ISBN 978-3642299681.