Aufgabenblatt 1

Systemkonzepte und -programmierung

2017

# Blatt 1

## Aufgabe 1| Kinokarten-Problem

1. Anfänglich steht in der zweiten Spalte überall 0 da alle Plätze noch frei sind.
2. Alle (200) Plätze sind nach 7826ms belegt.
3. Die parallele Reservierung dauert 5495ms und ist somit schneller als die sequentielle. Jedoch werden dabei insgesamt 203 Sitze reserviert.
4. i. nicht erfüllt, da 203 Sitze vergeben wurden

ii. nicht erfüllt, da mehr als 200 Sitze reserviert wurden, kann es nicht sein dass jeder Sitz höchstens einmal vergeben wird.

iii. erfüllt, durch betrachten der „seats“-Datei erkennt man, dass jeder Sitz vergeben wurde

iv. erfüllt, die Konsolenausgabe zeigt dass das Ablehnen nur am Ende stattfindet wenn es keine freien Karten mehr gibt.

## Aufgabe 2 | Atomarität

|  |  |
| --- | --- |
| Programm π : Registermaschine | |
| STORE 5,n | |
| p | **q** |
| *p1:* LOAD R1, n | *q1*: LOAD R1, n |
| *p2:* ADD R1, 2 | *q2:* LOAD R2, 2 |
| *p3:* STORE R1, n | *q3*: MULT R2, R1 |
|  | *q4*: STORE R2, n |

1. i.

|  |  |
| --- | --- |
| Programm π : Stapelmaschine | |
| PUSH 5 | |
| POP n | |
| p | **q** |
| *p1:* PUSH n | *q1*: PUSH n |
| *p2:* PUSH 2 | *q2*: PUSH n |
| *p3:* ADD | *q3:* ADD |
| *p4:* POP n | *q4:* POP n |

ii. Ausführungssequenz mit Endzustand n = 7

Für Registermaschine: p1 – q1 – q2 – q3 – q4– p2 – p3

Für Stapelmaschine: p1 – p2 – q1 – q2 – q3 – q4 – p3 – p4

iii. Ja, es gibt neun mögliche Ausführungssequenzen für die Registermaschine für die ebenfalls n = 7 als Endzustand erreicht wird. Die Bedingungen dafür sind dass die interne Reihenfolge von p und q jeweils eingehalten wird, dass p3 nach q3 und p1 vor q3 vorkommt.

Durch ausprobieren kommt man somit auf 9 verschiedene Möglichkeiten.

iv. Lost Update Problem:

Reg: p1 q1 p2 q2 p3 q3

Stap: p1 q1 q2 q3 q4 p2 p3 p4

Kein Lost Update Problem:

Reg: p1 p2 p3 q1 q2 q3

Stap: p1 p2 p3 p4 q1 q2 q3 q4

v. Die ganze Sequenz muss atomar ausgeführt werden, da das Laden und Speichern atomar geschehen muss um das Lost Update zu verhindern.

1. i. **4**: q1 p1 p2 q2 q3 p3 q4 p4

**3**: q1 p1 p2 q2 p3 p4 q3 q4

**2**: p1 p2 q1 q2 q3 q4 p3 p4

andere z.B.:

**6**: q1 q2 q3 q4 p1 p2 p3 p4

ii. **2**: p1 p2 q1 q2 q3 q4 p3 p4

**3**: q1 q2 p1 p2 p3 p4 q3 q4

**3**: q1 q2 p1 p2 q3 p3 q4 p4

iii. **6**: q123 q4 p123 p4

**2**: p123 p4 q123 q4

**3**: q123 p123 p4 q4

## Aufgabe 3 | Korrektheitsanalyse

1. i. p1 p2 q1 q3 q1 q2

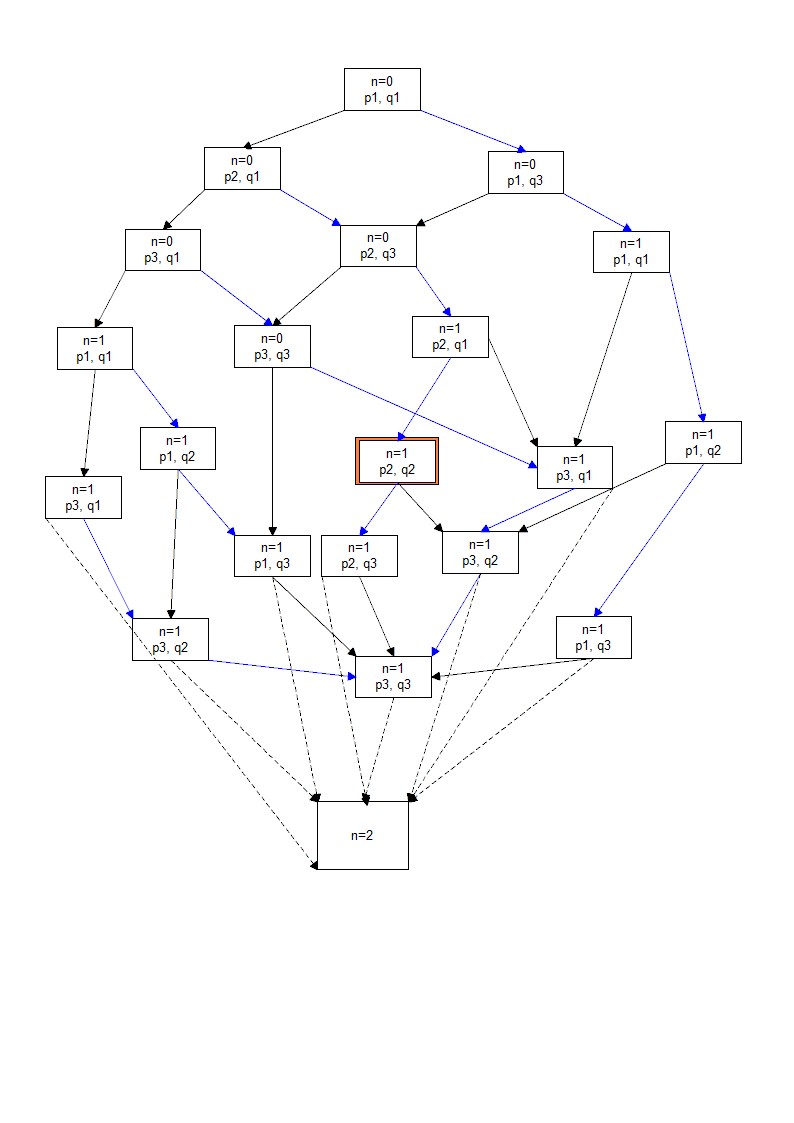
ii. 1 ist eine Lebendigkeitseigenschaft da, bei Verletzung in endlich vielen Schritten ein Zustand erreicht werden kann in dem die Eigenschaft erfüllt ist

und 2 eine Sicherheitseigenschaft da es, wenn sie verletzt ist in jeder möglichen Fortführung immer verletzt ist (einmal verletzt, immer verletzt)

1. Zustandsdiagramm:

P: Schwarzer Pfeil

Q: Blauer Pfeil

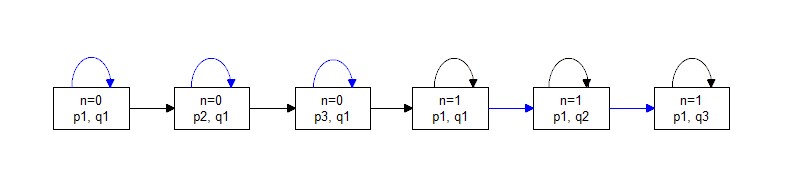


Eigenschaft 1: nicht erfüllt. Im orange umrandeten Zustand sind gleichzeitig p2 und q2, dh beide Prozesse greifen auf R zu.

Eigenschaft 2: erfüllt, zumindest für n<2. Da jeder Zustand eine Fortführung sowohl für p als auch für q hat (inkl gestrichelter Linien) und von jedem Zustand in mindestens 3 Schritten ein Zustand erreicht werden **kann**, in dem entweder p2 oder q2 enthalten ist.

1. Zustandsdiagramm:

*Übergänge wie oben.*



Eigenschaft 1 ist erfüllt. Offensichtlich kann der Zustand „p2,q2“ nicht erreicht werden, da nur die Zustände „pα,q1“ und „p1,qα“ für α in {1,2,3} erlaubt sind.

Eigenschaft 2 ist (analog zu 3b) auch erfüllt, da von jedem Zustand in mind 3 Schritten ein Zustand erreicht wird, der p2 oder q2 enthält und von jedem Zustand eine Fortführung für p und q möglich ist.