

Betriebssysteme (WS19/20)

Übungsblatt 6

Yudong Sun
12141043

2. Dezember 2019

Aufgabe H31 Sie können den Quellcode auch in der Datei `Server.java.txt` sehen.

(a)

```
7 public Server(int maxClients)
8 {
9     this.maxClients = maxClients;
10    this.anzahlClients = 0;
11    this.sicherungswunsch = false;
12 }
```

(b)

```
13 public synchronized void daten_ablegen(Client c) throws InterruptedException
14 {
15     System.out.println("Client " + c.ID + " will Daten ablegen");
16
17     // ----- kritischer Bereich -----
18     while(this.anzahlClients >= this.maxClients || this.sicherungswunsch)
19     {
20         try { wait(); }
21         catch(InterruptedException e) {}
22     }
23     anzahlClients ++;
24     System.out.println(anzahlClients + " Clients legen Daten ab.");
25 }
26 public synchronized void daten_ablegen_beenden()
27 {
28     anzahlClients --;
29     // ----- kritischer Bereich verlassen -----
30     notifyAll();
31 }
```

Es gibt bei beider Methoden kein „Sanity Check“, da wir davon ausgehen, dass die Methoden immer in einer sinnvollen Reihenfolge aufgerufen werden.

(c)

```

32 public synchronized void sicherungAktivieren() throws InterruptedException
33 {
34     this.sicherungswunsch = true;
35     System.out.println("Sicherungswunsch angemeldet!");
36
37     notifyAll();
38
39     // ----- kritischer Bereich -----
40     while(this.anzahlClients > 0)
41     {
42         try { wait(); }
43         catch(InterruptedException e) {}
44     }
45     // Zum Sichern bereit.
46 }
47 public synchronized void sicherungDeaktivieren()
48 {
49     this.sicherungswunsch = false;
50     // ----- kritischer Bereich verlassen -----
51     System.out.println("Sicherungswunsch deaktiviert.");
52     notifyAll();
53 }

```

(d) Die 2 kritische Bereichen sind:

- Als ein Client Daten in den Server ablegen möchte, und
- Als das Sicherungsskript eine Sicherung machen möchte.

Die beide Bereichen beziehen sich auf das Schreiben bzw. Lesen von Daten. Es wird hier mithilfe der folgenden Schritten sichergestellt, dass die Bedingungen der wechselseitigen Anschluss erfüllt ist:

- Die Methoden sind alle synchronisiert.
- Die Betriebsstatus des Systems wird durch die Variablen `Server.anzahlClients` und `Server.sicherungswunsch` gespeichert. Sie werden bei jedem Methodenaufruf aktualisiert.
- Das Sicherungsskript wartet darauf, dass `Server.anzahlClients` null ist. Das stellt es sicher, dass kein Client legt Daten auf dem Server.
- Die Clients warten, wenn `Server.sicherungswunsch` wahr ist.

Server.`sicherungswunsch` wird nur wieder auf falsch gestellt, wenn das Sicherungsskript fertig ist. Das stellt es sicher, dass kein Client legt mehr Daten auf dem Server, wenn eine Sicherung durchgeführt wird.

Dabei entsteht wechselseitige Ausschluss.

Aufgabe H32 Alle Stellen haben die Kapazität ∞ und alle Kanten haben das Gewicht 1. Die Stellen für die Semaphore existiert nur, weil die Kanten alle das Gewicht 1 haben müssen. Die Anfangsmarkierung beschreibt die in dem Übungsblatt gezeichneten Situation:

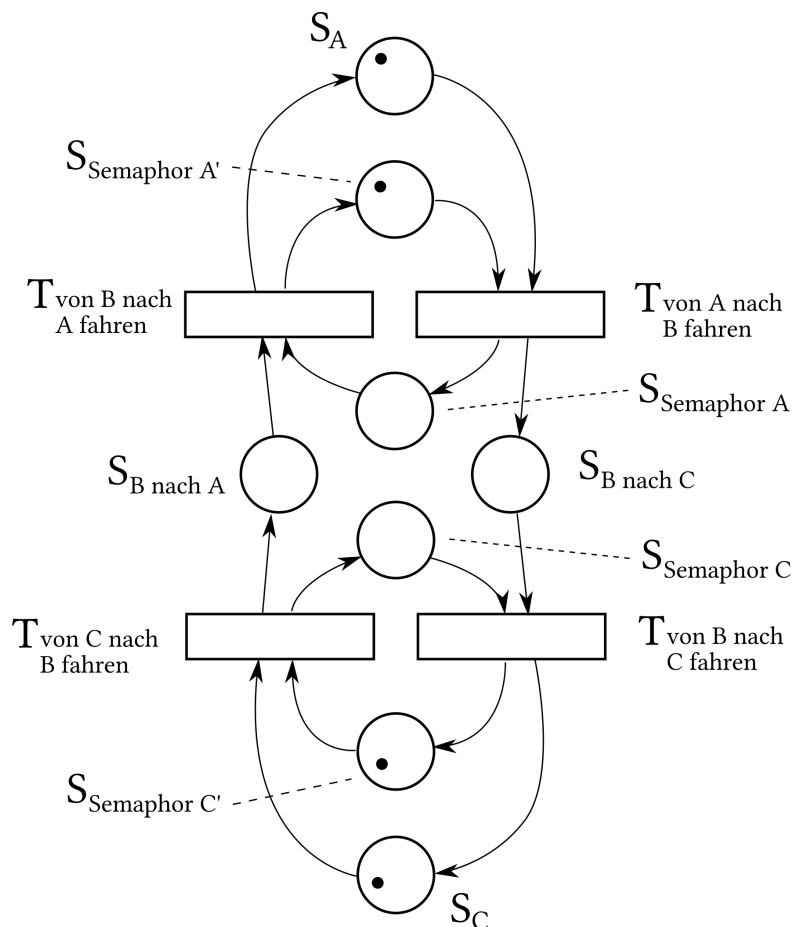


Abbildung 1: Petri-Netz Aufgabe 32

Es gibt 2 Stellen für den Abschnitt B, um die Bedingung zu modellieren, dass kein Richtungswechsel in B möglich ist. Die Sicherheitsbedingung wird durch die Semaphore $S_{\text{Semaphor } A}$ und $S_{\text{Semaphor } C}$ bzw. $S_{\text{Semaphor } A'}$ und $S_{\text{Semaphor } C'}$ sicher gestellt.

Aufgabe H33 Sehen Sie bitte u06-h33.txt