

Discrete Event Systems

Exercise 9

1 Strukturelle Eigenschaften von PN und Token Game

Sei folgendes Petrinetz gegeben:

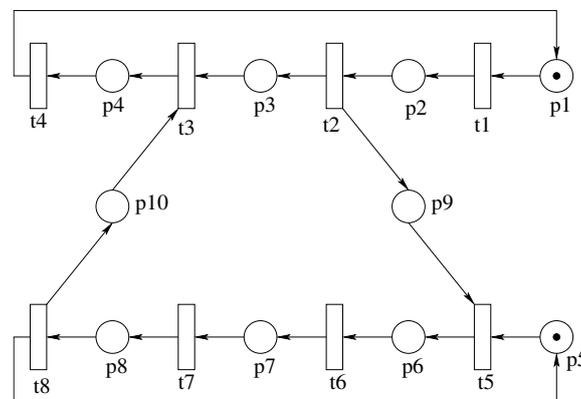


Abbildung 1: Petri-Netz N_1

- Benennen Sie die Prä- und Postmengen der Transitionen t_2, t_5, t_6 und der Plätze p_3, p_4, p_{10} in N_1 .
- Welche Transitionen sind nach dem Feuern von t_1 und t_2 aktiviert?
- Ermitteln Sie die Gesamtzahl der Tokens im Netz vor und nach dem Feuern von t_2 .
- Spielen Sie das Token-Spiel für N_1 . Beschreiben Sie in Worten, wie das Spiel abläuft, bauen Sie den Erreichbarkeitsgraphen auf und versuchen Sie herauszufinden, was das Petrinetz modellieren könnte.

2 Carsharing

Christoph, Michael und Johannes wohnen zusammen in einer WG. Sie haben ein gemeinsames Auto und zwei passende Autoschlüssel. Jeder von ihnen benötigt ab und zu das Auto: Michael fährt damit zum Hockeytraining, Christoph zu seiner Rollenspielgruppe und Johannes zum Fotoshooting. Wenn einer von ihnen losfahren will, braucht er natürlich einen der beiden Schlüssel und das Auto. Falls beide Schlüssel fehlen oder das Auto weg ist, muss er zu Hause bleiben. Sobald jemand von seinem Ausflug zurückkommt, stellt er das Auto ab und legt den Schlüssel wieder zurück an seinen Platz.

Modellieren Sie dieses Szenario mit einem Petri-Netz und erläutern Sie die Lösung.

3 Erreichbarkeitsanalyse

Benutzen Sie für diese Aufgabe folgendes Petri-Netz:

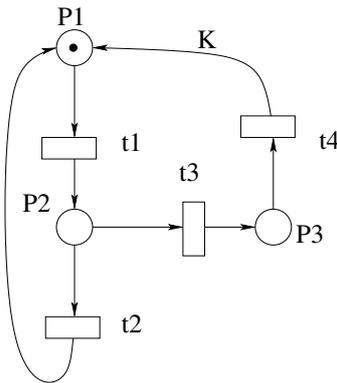


Abbildung 2: Petri-Netz N_2

- In der Vorlesung wurde ein Algorithmus zur Durchführung einer Erreichbarkeitsanalyse besprochen. Nehmen Sie an, die Zustandsmenge \mathbb{S}_{tmp} sei als eine FIFO-Liste (First-In-First-Out) implementiert und spielen Sie den Algorithmus für das Petri-Netz N_2 mit $K = 2$ durch bis Sie auf $\vec{s}_{test} = (2, 0, 1)$ stossen. Bauen Sie dabei den Erreichbarkeitsbaum auf (Knoten=Zustände, Kanten=Transitionen). Handelt es sich dabei um eine Tiefen- oder eine Breitensuche im Erreichbarkeitsgraphen?
- Wie wirkt es sich auf die Art der Suche (Breite oder Tiefe) aus, die der Erreichbarkeitsalgorithmus macht, wenn die Zustandsmenge \mathbb{S}_{tmp} anstatt als eine FIFO-Liste als eine LIFO-Liste (Last-In-First-Out) implementiert wird?
- Warum kann man mit einem Erreichbarkeitsalgorithmus im Allgemeinen nicht entscheiden, ob in einem PN eine Markierung erreichbar ist oder nicht?
- Ist in N_2 der Zustand $(P1, P2, P3) = (101, 99, 4)$ vom initialen Zustand $(1, 0, 0)$ für $K = 2$ erreichbar? Beweisen Sie Ihre Antwort.

Hinweis: Nutzen Sie die in der Vorlesung besprochene hinreichende Bedingung für Erreichbarkeit in einem gewichteten PN.

4 Gegenseitiger Ausschluss

Es soll ein System modelliert werden, in dem zwei Prozesse gemeinsam auf eine exklusive Ressource zugreifen wollen, d.h. die Prozesse müssen sich wechselseitig vom zeitgleichen Zugriff auf diese Ressource (bspw. einen kritischen Programmabschnitt) ausschliessen können. D.h. konkret: Ein Prozess arbeitet sein Programm ab. Bevor er den kritischen Abschnitt seines Programms betreten kann, muss eine gegebene Mutexvariable (oder Semaphore) $\neq 0$ sein. Ist dies der Fall, so setzt der Prozess die Semaphore auf 1 und arbeitet seinen kritischen Abschnitt ab, anschliessend setzt der Prozess die Semaphore wieder zurück und betritt einen unkritischen Abschnitt, der ebenfalls abgearbeitet wird. Nun beginnt der Prozess von vorne.

- Modellieren Sie dieses System als Petri-Netz.
- Welche Eigenschaft müssen alle Zustände im Erreichbarkeitsgraphen erfüllen, damit gewährleistet ist, dass die beiden Prozesse niemals gleichzeitig ihren kritischen Abschnitt betreten? Bitte formalisieren Sie diese Anforderung und markieren Sie die Zustände des dem PN zugrundeliegenden gelabelten Transitionssystems entsprechend.

5 Euklidscher Algorithmus

Folgender Algorithmus, bekannt als Euklids Algorithmus, berechnet den grössten gemeinsamen Teiler (ggT) zweier Zahlen $x, y \in \mathbb{N}$ mit $x \geq y$.

```
0: int ggT(int x, int y) {
1:   if(y == 0) return x;
2:   int z = 0;
3:   while(x > 0){
4:     x--; z++;
5:     if(z == y) z = 0;
6:   }
7:   return ggT(y, z); }
```

- a) Führen Sie obigen Code mit $x = 225, y = 60$ "manuell" aus. Notieren Sie dabei die jeweiligen Aufrufe der Funktion `ggT` sowie das Endresultat. Beschreiben Sie in Worten wie der Algorithmus funktioniert.
- b) Geben Sie analog zu Euklids Algorithmus ein PN an, das den ggT berechnet. Das PN soll zwei Eingabestellen P_x, P_y haben, die im Initialzustand x , bzw. y Tokens enthalten, sowie eine Ausgabestelle P_{ggT} die nach Ausführen des Token Games `ggT(x, y)` Tokens enthält.
Hinweis: Verwenden Sie Inhibitorkanten.

6 Boundedness und Deadlock Freeness

Erklären Sie anhand des Petrinetzes N_2 in Abb. 2 die Begriffe Beschränktheit (=Boundedness) und Deadlock-Freiheit. Für welche $K \in \mathbb{N}$ ist N_2 beschränkt bzw. unbeschränkt und für welche ist es (nicht) deadlockfrei?