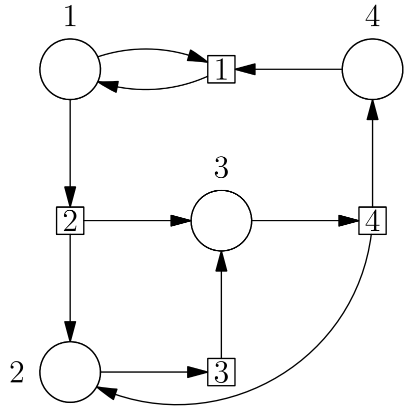
Datum: 29.06.2020

Übungsblatt mit Lösungen 12

Aufgabe T33

Gegeben ist folgendes Petrinetz:



Beantworten Sie die folgenden Fragen. Begründen Sie jeweils Ihre Antwort.

1. Ist das Netz mit Startmarkierung $(1, 0, 0, 0)$ beschränkt?
2. Kann das Netz von der Startmarkierung $(0, 0, 1, 0)$ aus verklemmen?
3. Ist von der Startmarkierung $(1, 0, 0, 1)$ die Markierung $(0, 1, 1, 3)$ erreichbar?

Lösungsvorschlag

1. Nein, das Netz ist von dieser Startmarkierung nicht beschränkt. Es ist möglich mit Hilfe der ausgehenden Transitionen von 2 und 3 unbeschränkt viele Marken auf Position 4 zu erzeugen.

2. Nein, das Netz kann von dieser Startmarkierung aus nicht verklemmen, da die einzig möglichen Transitionen die ausgehenden Positionen zwischen den Positionen 2 und 3 sind und diese immer abwechselnd feuern können.

3. Ja. Wir betrachten hierzu das Matrixverfahren: Nach Vorlesung stellen wir die Transitionsmatrix auf:

$$D = \begin{pmatrix} 0 & 0 & 0 & -1 \\ -1 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

   Wir wollen nun folgendes Gleichungssystem lösen:

   $\begin{pmatrix} 0 & 1 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} + xD$

   Falls $x \notin \mathbf{N}$ wissen wir, dass $(0, 1, 1, 3)$ keine Folgemarkierung von $(1, 0, 0, 1)$ ist. Das Gleichungssystem hat folgende Lösung:

   $x_2 = 1, x_3 = x_1 + 2, x_4 = x_1 + 2$

Für jedes $x_1 \in \mathbf{N}$ hat das Gleichungssystem demnach eine ganzzahlige Lösung. Wir wählen $x_1 = 0$. Wir müssen nun prüfen, ob die errechneten Transitionen $x = \begin{pmatrix} 0 & 1 & 2 & 2 \end{pmatrix}$ auch tatsächlich anwendbar sind. Dies ist tatsächlich der Fall: Wir können zuerst Transition 2 schalten, anschließend 3, zwei mal 4 und zuletzt wieder Transition 3.

Aufgabe T34

Wir betrachten ein weiteres mal die Programme P_1 und P_2 aus T31:

```
x := 1;                                          x := 0;
if(x=0) print;                                   if(x=1) print;
```

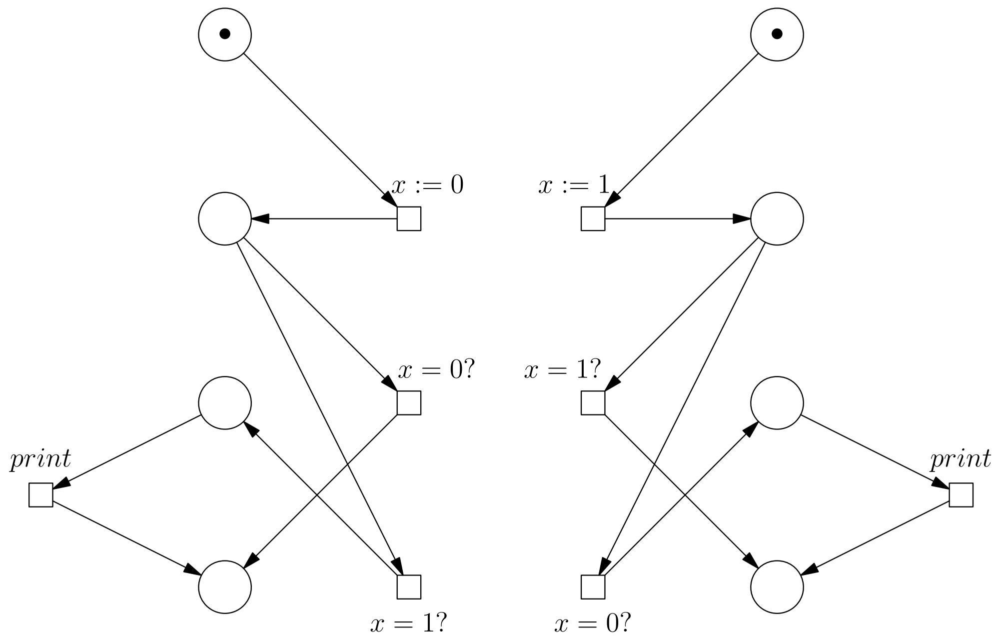
a) Bilden Sie Petrinetze für P_1, P_2 und die Variable, und modellieren Sie die nebenläufige Ausführung mit Hilfe von (un-) synchronisierten Produkten.

b) Begründen Sie, dass im resultierenden Petrinetz von a) keine Markierung erreichbar ist, in der beide Vorbereiche der `print`-Transitionen besetzt sind.

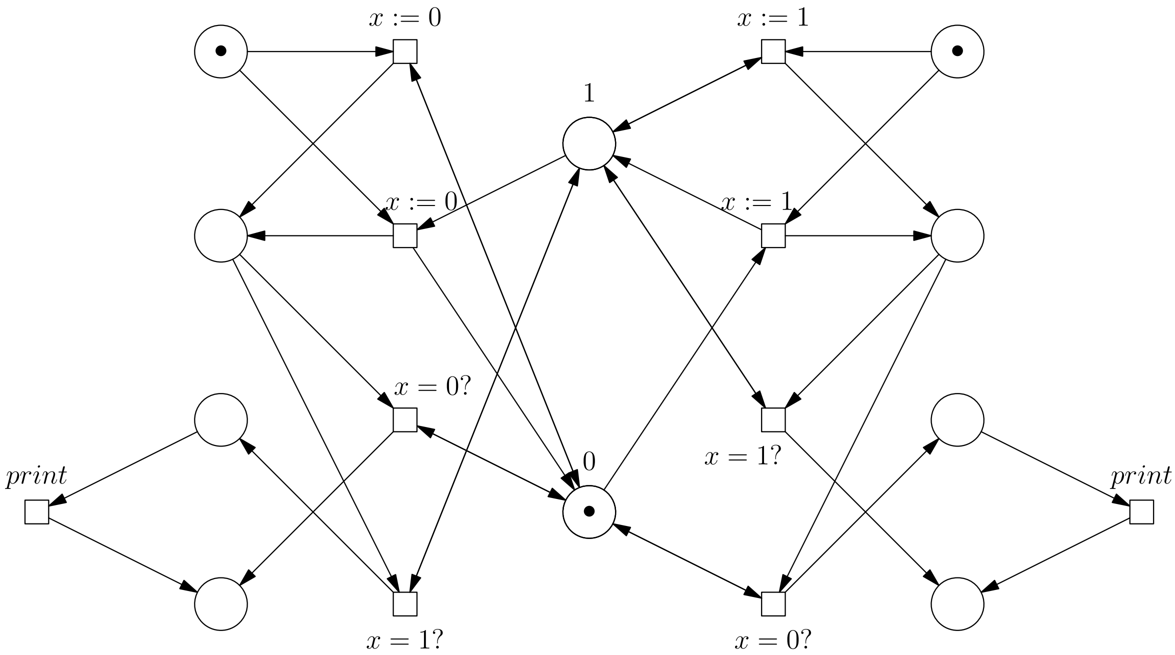
Lösungsvorschlag

a)

Petrinetze für P_1 und P_2 und ihr unsynchronisiertes Produkt:



Wir fügen ein Petrinetz für die Variable ein und vereinigen die Transitionen mit den gleichen Anweisungen jeweils paarweise wie in Aufgabe T31.



b) Angenommen es liegt eine Markierung in den beiden `print`-Vorbereichen die wir im folgenden c und d nennen werden. Dann muss sowohl die Transition $x = 1?$ von P_1 und $x = 0?$ von P_2 ausgeführt worden sein. Dies ist nur möglich wenn P_1 $x := 0$ und P_2 $x := 1$ ausgeführt hat. Da $x = 1?$ von P_1 und $x = 0?$ von P_2 aber einen Marker in Zustand 1 bzw. Zustand 0 benötigen, muss folgen, dass die entsprechenden $x := 1$ und $x := 0$ Transitionen vorher ausgeführt wurden. Wir geben den Transitionen namen: Sei $A = [x := 0]$, $B = [x = 1?]$, $C = [x := 1]$ und $D = [x = 0?]$ und sei "$<$" eine Relation, wo $A < B$ ausdrückt, dass A vor B geschehen muss. Wir erhalten die Bedingungen $A < B$, $C < D$, $C < B$ und $A < D$. Die einzigen Reihenfolgen die dies Erlauben sind die, in denen A und C vor B und D ausgeführt werden. Nachdem A und C beide in beliebiger Reinfolge ausgeführt wurden, liegt entweder ein Marker in 1 oder in 0 und kann nicht wieder bewegt werden. Daraus folgt, dass nicht in c *und* in d eine Makierung liegen kann.

Aufgabe T35

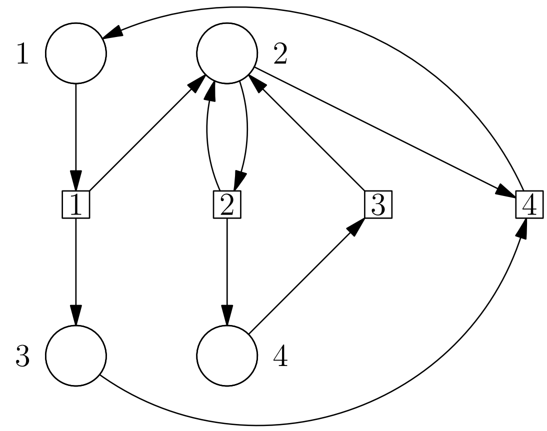
Wie groß ist das (un-) synchronisierte Produkt zweier Petrinetze bezüglich seiner Anzahl an Stellen und Transitionen?

Lösungsvorschlag

Bei (un-) synchronisierten Produkt zweier Petrinetze addiert sich die Anzahl der Stellen immer. Im folgenden betrachten wir die Anzahl der Transitionsknoten. Beim unsynchronisierten Produkt addiert sich auch die Anzahl der Transitionen, beim synchronisierten hingegen müssen die synchronisierten Transitionen paarweise verknüpft werden, d.h. wenn zwei Petrinetze jeweils s_1 und s_2 synchronisierende Transitionen und u_1 und u_2 unsynchronisierende Transitionen haben hat das entstehende Petrinetz danach $u_1 + u_2 + s_1 s_2 - s_1 - s_2$ Transitionen.

Aufgabe H39 (10 Punkte)

Gegeben ist folgendes Petrinetz:



Beantworten Sie die folgenden Fragen. Beweisen Sie jeweils Ihre Antwort.

1. Gibt es eine Startmarkierung, von der aus es möglich ist, nach genau 5 Transitionen eine Verklemmung des Netzes zu erzeugen?

2. Ist von der Startmarkierung $(0, 1, 1, 0)$ die Markierung $(2, 0, 0, 0)$ erreichbar?

Lösungsvorschlag

1.) Nein, eine solche Startmarkierung kann es nicht geben. Die einzige Menge an Markierungen, von denen das Netz aus verklemmen kann, ist gegeben durch $\{ (0, 0, k, 0) \mid k \in \mathbf{N} \}$, da der Vorbereich jeder Transition bis auf Transition 4 aus jeweils genau einem der verbleibenden Plätze besteht.

Es ist jedoch auch auch keine verklemmende Markierung von einer anderen Startmarkierung aus erreichbar. Nehmen wir zwecks eines Widerspruchsbeweises an, dass $(0, 0, k, 0)$ für ein $k \in \mathbf{N}$ nach fünf Schritten erreichbar ist. Wir betrachten die Markierung (a, b, c, d) vor dem Schalten der fünften Transition. Wir wissen, dass $a, b, c, d \in \mathbf{N}$, da nur dies legale Markierungen nach Definition sind. Nun führen wir eine Fallunterscheidung darüber, welche Transition zuletzt schaltet:

1. Die Folgemarkierung lautet $(a - 1, b + 1, c + 1, d)$, was einen Widerspruch darstellt, da $b + 1 \neq 0$.

2. Die Folgemarkierung lautet $(a, b+1, c, d+1)$, welches zu dem gleichen Widerspruch führt.

3. Die Folgemarkierung lautet $(a, b+1, c, d-1)$, welches zu dem gleichen Widerspruch führt.

4. Die Folgemarkierung lautet $(a + 1, b - 1, c - 1, d)$, was einen Widerspruch darstellt, da $a + 1 \neq 0$.

In jedem der Fälle können wir also keine verklemmende Markierung nach 5 Schritten erreichen.

2.) Nein. Wir betrachten hierzu das Matrixverfahren: Nach Vorlesung stellen wir die Transitionsmatrix auf:

$$D = \begin{pmatrix} -1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 0 \end{pmatrix}$$

Wir wollen nun folgendes Gleichungssystem lösen:

$\begin{pmatrix} 2 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 0 \end{pmatrix} + xD$

Falls $x \notin \mathbf{N}$ wissen wir, dass $(2, 0, 0, 0)$ keine Folgemarkierung von $(0, 1, 1, 0)$ ist.

Tatsächlich hat obiges Gleichungssystem keine Lösung. Damit ist sichergestellt, dass $(2, 0, 0, 0)$ nicht von der genannten Startmarkierung aus erreichbar ist.

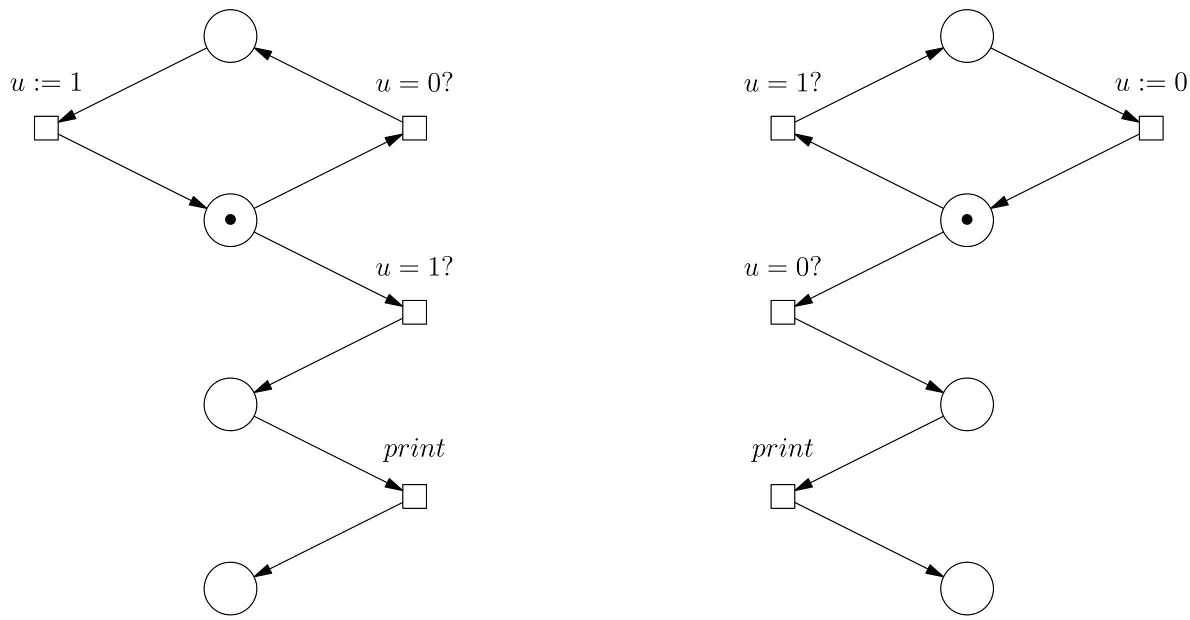
Aufgabe H40 (10 Punkte)
Gegeben sind die folgenden Programme P_1 und P_2.

```
while(u=0){
    u := 1;
}
print;
```

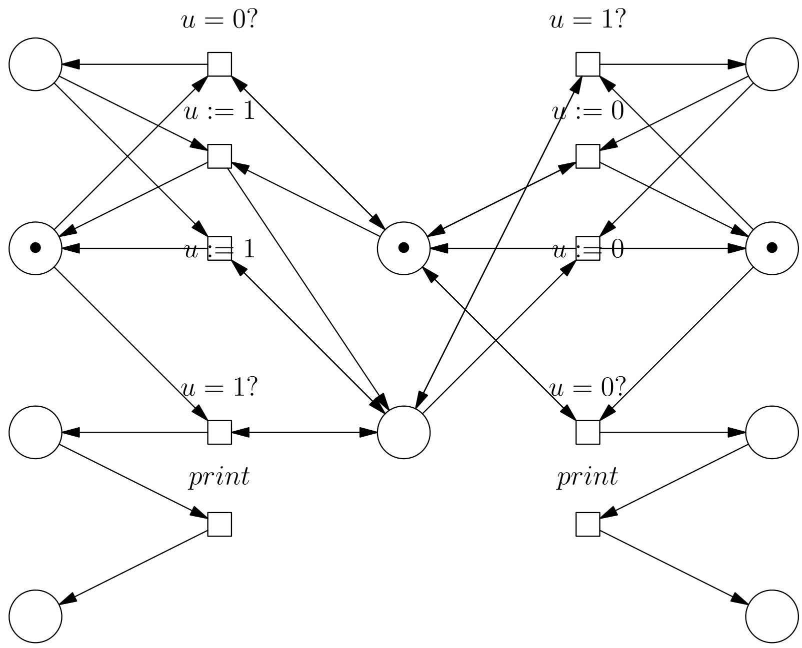
```
while(u=1){
    u := 0;
}
print;
```

Wie oft kann `print` ausgeführt werden, falls P_1 und P_2 gleichzeitig laufen? Geben Sie alle Möglichkeiten an. Modellieren Sie das Problem mit Hilfe von synchronisierten Petrinetzen. Begründen Sie Ihre Antwort mit Hilfe des entstandenen Netzes.

Lösungsvorschlag
Petrinetze für P_1 und P_2 und ihr unsynchronisiertes Produkt:



Wir fügen ein Petrinetz für die Variable ein und vereinigen die Transitionen mit den gleichen Anweisungen jeweils paarweise.

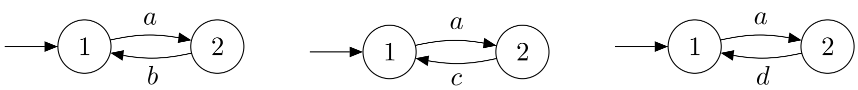


Wir behaupten dass die print Anweisung keinmal oder genau zweimal ausgeführt wird.

Um einen (unendlichen) Lauf mit null print-Anweisungen zu erhalten, muss P_1 erst $u = 0?$ und dann $u := 1$ ausführen, danach P_2 $u = 1?$ und $u := 0$. Damit erhält man wieder die Startmarkierung und man wiederholt den Prozess. Um zwei print-Anweisungen aufführen zu können, kann P_1 einmal komplett durchlaufen, also $u = 0?$, $u := 1$, $u = 1?$, *print* ausführen und P_2 danach komplett durchlaufen. Offensichtlich kann nicht mehr als zweimal geprinted werden, da der Nachbereich der print-Aktionen kein Vorbereich irgendeiner Aktion ist. Nun zeigen wir, dass nicht genau einmal print gefeuert werden kann: Zuerst beobachten wir, dass die Anzahl der Marken immer drei ist und je eine Marke in Prozess P_1, P_2 und der boolschen Variable ist. Damit print genau einmal ausgeführt wird, muss genau ein Prozess (o.B.d.A P_2) schon print gefeuert haben und die Marke ist im Nachbereich dieser Transition und der andere Prozess (P_1) ist verklemmt oder läuft unendlich lange. Wenn eine Marke in der Startmarkierung von P_1 liegt, kann man mittels $u = 0?$, $u := 1$, $u = 1?$, *print* ausführen, wenn eine Marke in Zustand 0 von der boolschen Variable liegt. Wenn es in Zustand 1 der Variable liegt, kann man direkt $u = 1?$ und dann *print* ausführen. Wenn die Marke in der obersten Stelle liegt, kann man direkt $u := 1$ ausführen und kommt zum vorherigen Fall, unabhängig in welcher Stelle die Marke in der boolschen Variable liegt. Damit haben wir gezeigt, dass genau eine Print-Anweisung nicht möglich ist.

Aufgabe H41 (10 Punkte)

Bestimmen Sie das synchronisierte Produkt folgender drei NFA.



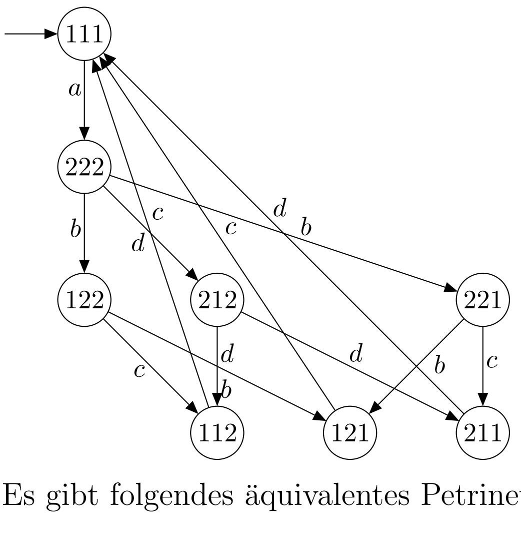
Wie viele Zustände hat Ihr synchronisiertes Produkt?

Geben Sie nun ein äquivalentes Petrinetz an, indem Sie zuerst für jeden NFA ein äquivalentes Petrinetz angeben und diese drei Netze dann nach dem Verfahren der Vorlesung synchronisieren. Geben Sie außerdem eine Startmarkierung an, sodass ihr konstruiertes Petrinetz äquivalent zu Ihrem zuvor gebildeten synchronisierten Produkt ist und beschriften Sie die Transitionen der Petrinetze mit den entsprechenden Buchstaben.

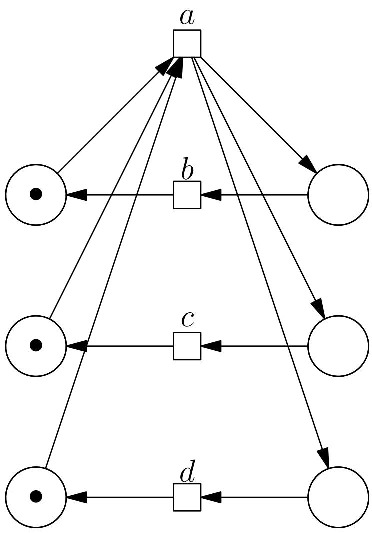
Wie viele Stellen hat Ihr Petrinetz?

Lösungsvorschlag

Synchronisieren der drei Automaten ergibt einen Automaten mit 8 Zuständen:



Es gibt folgendes äquivalentes Petrinetz mit nur 6 Stellen und 4 Transitionen.


a
b
c
d