

Übung 7: Petri Netze (PN) & BDDs

Remark:

You can download the Moses package (approx. 8 MB) from:

www.tik.ee.ethz.ch/~klampka/downloads/moses.zip.

This high-level modelling framework allows you to solve some of the following exercises, but its use is not mandatory. Moses should simply serve as an example for tool-based modeling and analysis, so please feel free to check it out. Trying it out is as simple as:

1. unzip the file
2. change to the unzipped directory "moses"
3. run Moses using the following commands: `java -jar moses.jar -d moses.rep` (No classpath specification is necessary for the java calls, since it is contained in the manifest file contained in moses.jar.)

You will find a doc directory as well, in which the features of the editor (and how to use it) are described. A tutorial can also be found at

www.tik.ee.ethz.ch/~klampka/downloads/MosesTutorial.pdf

Once you have moses running, you must create a directory tmp (somewhere). Within moses it is now necessary, setting the variable tmp to this directory, full path name. Also it might be necessary to install other tools like dotty so you can plot state graphs etc. Unfortunately the SimplePetriNet-Formalism is not fully implemented. Instead you should make use of Timed PN. Within the models just leave the timing parameter of transitions empty, which gives you a PN. You may know use moses for solving some of the exercises.

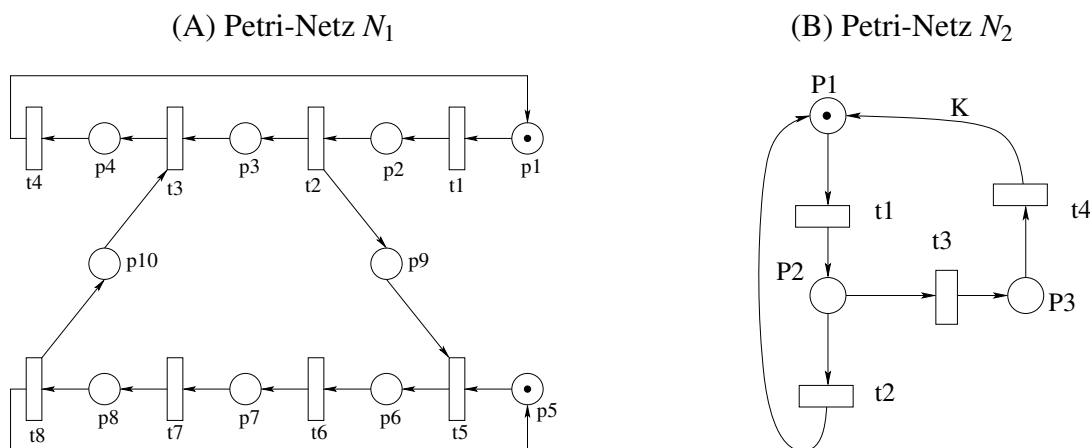


Abbildung 1: Zwei einfache Petri-Netze

Aufgabe 1: Strukturelle Eigenschaften von PN und Token game

Sei das in Abbildung 1.A gezeigte Petrinetz gegeben.

1. Benennen Sie die Prä- und Postmengen der Transitionen t_5, t_6, t_7, t_8 und der Plätze p_3, p_4, p_5, p_6 .
2. Welche Transitionen sind nach dem Feuern von t_1 und t_2 aktiviert, was wird hier modelliert?
3. Ermitteln Sie die Gesamtzahl der Marken im Netz vor und nach dem Feuern von t_2 .
4. Spielen Sie das Token-Spiel für N_1 und bauen Sie den Überdeckungsgraphen auf.

Aufgabe 2: Grundlegende Eigenschaften von PN

Sei das in Abbildung 1.B gezeigte Petrinetz gegeben.

1. Erklären Sie die Begriffe: Beschränktheit, Lebendigkeit und Verklemmungsfreiheit an diesem Beispiel.
2. Für welches $K \in \mathbb{N}$ ist das Netz N_2 aus Abb. 1 beschränkt bzw. unbeschränkt, lebendig und nicht deadlockfrei?

Aufgabe 3: Gegenseitiger Ausschluss

Es soll ein System modelliert werden, in dem 2 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 krit. Abschnitt ab, anschliessend setzt der Prozess die Semaphore wieder zurück und betritt einen unkritischen Abschnitt, der ebenfalls ab gearbeitet wird. Nun beginnt der Prozess von vorne.

1. Modellieren Sie dieses System als Petrinetz.
2. Welche Eigenschaften müssen alle möglichen Markierungen erfüllen, damit das System tatsächlich korrekt arbeitet?

Aufgabe 4: Erreichbarkeitsanalyse

In der Vorlesung wurde ganz allgemein ein Algorithmus zur Durchführung einer Erreichbarkeitsanalyse besprochen.

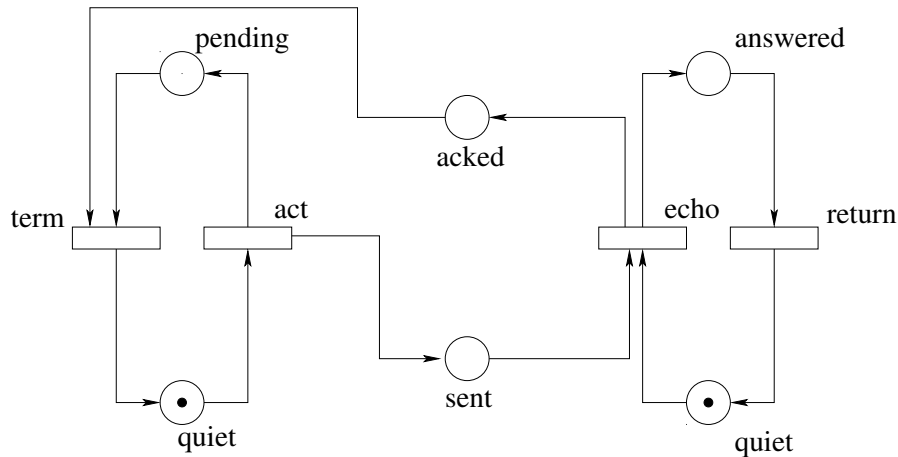


Abbildung 2: Basis-Crosstalk-Petri-Netz

1. Handelt es sich bei dem besprochenen Algorithmus um einen Tiefen- oder Breitensuche, –bitte erklären Sie die Begriffe?
2. Wie erzeuge ich entsprechend eine Version, die dem anderen Suchschemata entspricht?

Zusatzaufgabe :

Crosstalk In dieser Aufgabe soll ein Modell eines Crosstalk-Algorithmus mit Hilfe eines Petri-netzes erstellt werden.

Eine Crosstalk-Situation in Verteilten Systemen tritt dann auf, wenn zwei Clients gleichzeitig Nachrichten senden wollen. Dies ist beispielsweise bei einem Ethernet-Netz problematisch, da hier immer nur ein Client als Sender auftreten darf und nicht gleichzeitig weitere Sender aktiv sein dürfen.

Sei das in Abbildung 2 gezeigte Petri-Netz gegeben, in dem die linke Seite Sender und die rechte Seite Empfänger von Nachrichten ist.

1. Erweitern Sie das PN so, dass beide Seiten sowohl Sender als auch Empfänger sein können (Symmetrie).
2. Finden Sie die Probleme dieser einfachen “Lösung” durch Inspektion des Erreichbarkeitsgraphen.

Aufgabe 5: BDD-basierte Komposition von PN

Gegeben seien 2 Instanzen des PNs aus Abb. 1.B, mit $K=1$.

1. Kodieren Sie die Transitionsrelation der Subnetze als BDD über den Variablen a_1, a_2 (Transitionsidentifizier), p_i^j, p_i^j' (Zustandsbelegungen vor/nach Transitionsausführung) fuer die stelle P_i des Subnetzes j .
2. Kodieren Sie nun die Transitionsrelation der parallel geschalteten Subnetze mit Hilfe eines kompositionellen Schemata, wobei alle Transitionen nebenläufig sein sollen. Was ist das Problem, wenn Sie auf den Variablen des jeweiligen anderen Subnetzes die Identität verwenden?

3. Wie müssen Sie nun ihren Ansatz ändern, wenn Transition t_4 in beiden Netzen immer gemeinsam ausgeführt werden muss? (synchronization)

Aufgabe 6: Zero-suppressed BDDs

In der Vorlesung wurden RO-BDDs und der Apply-Algorithmus besprochen. Anstatt *don't care* Knoten bzw. Variablen zu überspringen, wollen wir nun mit 0 belegte Variablen auslassen. Dies gibt einen ein sog. zero-suppressed BDD [Minato'93], in welchem also gilt:

$$\nexists n \in \mathbf{K}_{NT} : \text{then}(n) = 0 - \text{node}$$

1. Wie muss nun der Apply-Algorithmus angepasst werden, so dass man binäre Operatoren auf ZBDDs anwenden kann.
2. Welche Voraussetzung müssen diese Operatoren erfüllen?