EE552
(Extra Credit Project)

Introduction to Petri Nets

Mani Balakrishnan
Introduction to Petri Nets

**History**

Petri nets were introduced by C.A. Petri in the early 1960s as a mathematical tool for modeling distributed systems and, in particular, notions of concurrency, non-determinism, communication and synchronization.

Their further development was facilitated by the fact that Petri Nets easy model process synchronization, asynchronous events, concurrent operations, and conflicts or resource sharing.

Petri Nets have been successfully used for concurrent and parallel systems modeling and analysis, communication protocols, performance evaluation and fault-tolerant systems.

**A quick overview of Petri Nets**

There are many varieties of Petri nets from simple net (let’s call them black and white Petri nets), which are conceptually simple and straightforward to analyze, to more complex nets such as colored nets, timed nets and stochastic nets.

**Black and white Petri nets**

A simple (black and white) Petri net is a digraph with nodes that are places (circles) or transitions (rectangles). Nodes of different kind are connected together by means of arcs. Arcs are of two kinds:

- **Input arcs** that connect one place to one transition.
- **Output arcs** that connect one transition to one place.

A Petri net can be initialized by indicating the tokens which are contained in each place at starting time.

At any time the distribution of tokens among places defines the current state of the modeled system.

**Petri net can be executed by**:

1. establishing an initial marking,
2. choosing a set of eligible transitions,
3. firing a transition among the set of eligible ones,

4. Going back to step 2 until no more transition is eligible.

A transition is said to be eligible if all its input places contain (at least) one token. Then if it does fire, one token is removed from each of its input places and one token is added to each of its output places.

______________________________

Colored Nets

Colored nets are extended Petri nets in which tokens are differentiated by "COLORS".

Colored nets allow the modeling of complex systems and data flows.

Transition eligibility depends then on the availability of an appropriately colored token in all the input places of this transition.

Similarly, the output of a transition is not just a token but a specifically colored token.

The notation of colored nets is far more concise than black and white nets, and thereby avoids a lot of the duplication which is typical of black and white nets.

______________________________

Timed and Stochastic Nets

Timed nets are Petri nets that attach delays to transitions to give them the ability to model time.

Stochastic nets are Petri nets that attach delays to places.

The Basics

A Petri Net is a collection of directed arcs connecting places and transitions. Places may hold tokens. The state or marking of a net is its assignment of tokens to places. Here is a simple net containing all components of a Petri Net:
Arcs have capacity 1 by default; if other than 1, the capacity is marked on the arc. Places have infinite capacity by default, and transitions have no capacity, and cannot store tokens at all. With the rule that arcs can only connect places to transitions and vice versa, we have all we need to begin using Petri Nets. A few other features and considerations will be added as we need them.

A transition is enabled when the number of tokens in each of its input places is at least equal to the arc weight going from the place to the transition. An enabled transition may fire at any time. When fired, the tokens in the input places are moved to output places, according to arc weights and place capacities. This results in a new marking of the net, a state description of all places.

Definition for **Petri Net**:

Petri Net is a *bipartite directed graph* represented by a quadruple $PN = (P, T, I, O)$ where:

- $P = \{p_1,...,p_n\}$ is a finite set of places.
- $T = \{t_1,...,t_m\}$ is a finite set of transitions.
- $I(p,t)$ is a mapping $P \times T \rightarrow \{0,1\}$ corresponding to the set of directed arcs from places to transitions.
- $O(t,p)$ is a mapping $T \times P \rightarrow \{0,1\}$ corresponding to the set of directed arcs from transitions to places.

In the above figure, the two places have a token in it. This will result in firing of the transition, and the each places after the transition are filled with a token.
When arcs have different weights, we have what might at first seem confusing behavior. Here is a similar net, ready to fire:

![Net Diagram]

The bottom figure shows the state after firing:

![Net Diagram]

When a transition fires, it takes the tokens that enabled it from the input places; it then distributes tokens to output places according to arc weights. If the arc weights are all the same, it appears that tokens are moved across the transition. If they differ, however, it appears that tokens may disappear or be created. That, in fact, is what happens; think of the transition as removing its enabling tokens and producing output tokens according to arc weight.

A special kind of arc, the inhibitor arc, is used to reverse the logic of an input place. With an inhibitor arc, the absence of a token in the input place enables, not the presence:
Consider the above example. In this place P1 and P2 have a token in it. But from the figure we can see that the transition will fire only if P2 is inhibited i.e. only when P2 doesn’t have a token in it.

So in the above example, the transition will not fire. But if we remove the token from P2 then the transition can fire.

**Types of Petri nets:**

Here is a collection of primitive structures that occur in real systems, and thus we find in Petri Nets.

**Fig 1: Linear Petri net**

**Fig 2: Branching net**

**Fig 3: Concurrency net**

**Fig 4: Conflict**

**Fig 5: Merging**
Example 1:

At point T: The transition can be fired continuously (infinite number of times).

At point Trap: The transition can be fired many number of times but it cannot be infinite.

Consider N1 has a token in it. T1 will be fired. This will result in a token be placed in N2 and N1. Since the path covered by T will fire N1 continuously, at any time of the stage there will be a token in N1.

T2 will be fired when a token is placed in N2. Thus a token is taken and deposited at N3. Now if there is a token in N2, then T3 will be fired. This stage is like T but at this point it can fire many number of times but it cannot be infinite.
Adding enabling predicates variables

\[ X = \text{Program variable} \]

Due to monotonic property, ordinary Petri nets cannot achieve a test for zero i.e. only fires when it is zero or greater than zero. Therefore they are not computation universe.

Explanation:
Consider that there are three tokens in T1 and one token is always present in T4. N1 is fired and one token from taken from T1 and it is deposited in T2. Now T1 will have two tokens. N2 will be fired since both T2 and T4 has one token in it. Now the token in T2 is carried to T3. After T3 gets the token N3 will be fired and the token is deposited into T1 and one in T4. Thus the process will continue.
Example:

Consider that token is in N1. T1 can fire since there is a token in N1. Now the token from N1 is taken and put into N2 and N3. At this moment both transition can fire, since both N2 and N3 nodes have a token in them. Let us consider that T2 fires first. When this happens token will shift from node N2 to node N4. Now T5 cannot fire since only N4 has a node but not node N5.

So there cannot be any possible movement for the token from N4 till node N5 gets a token. Now consider that T3 fires. Token from node N3 is put into node N5. Since there is a token available in node N4 and N5, there is a possibility that any one transition T5 or T4 will fire. Let us consider that T4 fires. If this happens token from N5 is put into node N3. But now T5 cannot be fired because node N5 doesn’t have a token.

T5 can fire only if T4 is not fired. If T5 fires then a token is deleted from both N5 and N4, and a token is put into node N1. Even though there is totally two token available from node N5 and N4, only one token will be put into node N1. Also, since there is a token in node N5, T5 can fire.
Example: To find the power of a number. X power Y (Not taken from any reference)

Consider that P1 contains the power of the number. P5 contains the number to be powered. When we place a token in P2, T1 will be fired. A token is put in P3. T2 will be fired and a token is put in P4. This will result in firing of T3.

A token will be placed in P7 and P3. But T4 will not be fired, since it has a weight age of X. So the process will be repeated still P7 has a token of X. When P7 has X tokens, T4 will be fired and P2 will have one token since its weight age is one, P6 will get X tokens and P5 will be filled again with the term for which we need to find the power i.e. X.

T1 will be fired once again and the process gets repeated again. But the process goes through the cycle for Y number of times. This is because after Y term, P1 will not be having any token to fire T1. At this time P6 will be having a token which will be the required number i.e. X power Y.
Example

Here is an example of a Petri Net model, one for the control of a vending machine.

This machine stocks five items. When a coin is accepted, an item can be dispensed. When all items are dispensed, the request refill place gets a token, and the refill transition replenishes the machine.

Here is the initial state:
A coin is ready to drop. T1 firing leads to

where either the coin is rejected or accepted. A rejected coin simply takes us back to the initial marking. An accepted coin leads to dispensing:

Notice what, when T4 fires, a token will be placed in P2, to start another dispensing cycle, and a token will be removed from P4 and added to P6. After inserting several coins, we get to the following marking:
where a coin is ready, but there is nothing left to dispense. The inhibitor arc from P4 now enables T6; when it fires, T5 will be enabled, and when it fires, goods are removed from P6 and placed in P4, ready to start another complete cycle.

**Conclusion:**

Petri nets are a promising tool for describing and studying systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. As a graphical tool, Petri nets can be used as a visual-communication aid similar to flow charts, block diagrams, and networks. In addition, tokens are used in these nets to simulate the dynamic and concurrent activities of systems. As a mathematical tool, it is possible to set up state equations, algebraic equations, and other mathematical models governing the behavior of systems.

To study performance and dependability issues of systems it is necessary to include a timing concept into the model. There are several possibilities to do this for a Petri net; however, the most common way is to associate a firing delay with each transition. This delay specifies the time that the transition has to be enabled, before it can actually fire. If the delay is a random distribution function, the resulting net class is called stochastic Petri net. Different types of transitions can be distinguished depending on their associated delay, for instance immediate transitions (no delay), exponential transitions (delay is an exponential distribution), and deterministic transitions (delay is fixed). Thus, Petri Nets have proven useful for modeling, analyzing, and verifying protocols typically used in networks.

**Reference:**

Web site of The World of Petri Nets: [www.daimi.au.dk/PetriNets](http://www.daimi.au.dk/PetriNets)
[www.worldserver.oleane.com/adv/elstech/petrinet.htm](http://www.worldserver.oleane.com/adv/elstech/petrinet.htm)
[www.abo.fi/~atorn/Petri/PCont.html](http://www.abo.fi/~atorn/Petri/PCont.html)
[www.pdv.cs.tu-berlin.de/~azi/petri.html](http://www.pdv.cs.tu-berlin.de/~azi/petri.html)

Lecture notes from [www.taylor.eng.clemson.edu](http://www.taylor.eng.clemson.edu)
Declaration

I alone prepared and wrote this project. I received no help from anyone else. This material is not copied or paraphrased from any other source except where specifically indicated. I grant my permission for this project to be placed on the course homepage during future semesters. I understand that I could receive an F for the course retroactively, even after graduation, if this Work is later found to be plagiarized.

Mani Balakrishnan