Petri Nets

- Part of this discussion is based on the paper
  - *Petri Nets: Properties, Analysis and Applications*

- Petri Nets
  - graphical and mathematical modeling tool
  - tool for describing systems characterized as being:
    - concurrent, asynchronous, distributed, parallel, nondeterministic and/or stochastic
Petri Nets

- **History**
  - **1962:** Carl Adam Petri’s submitted his dissertation at the Uni. Darmstadt, Germany
  - **1970:** early development was published by A.W. Host and in the records of the 1970 Project MAC Conference on Concurrent Systems and Parallel Computation
  - **1970-75:** Computation Structure Group and MIT was most active
  - **1975:** conference on Petri Nets and Related Methods at MIT
  - **1979:** 135 researchers assembled in Hamburg, Germany, for 2-week advanced course on General Net Theory of Processes and Systems
  - **1980:** first European Workshop on Applications and Theory of Petri Nets, Strasbourg, France.
  - check out Murata’s paper for the extensive literature discussion
Petri Nets

- General:
  - directed, weighted, bipartite graph
  - two kinds of notes (Places P, Transitions T)
  - arcs from P to T or from T to P
  - arcs have integer weights
  - non-negative Place weights are called tokens
Petri Nets

- A Petri Net is a 5-touple PN={P,T,A,W,M0}
- Place Set P = \{p_1, p_2, ..., p_m\}
  - finite set of places
  - condition = place
  - one condition or set of atomic conditions
  - symbol

- Transition Set T = \{t_1, t_2, ..., t_n\}
  - finite set of transitions
  - action = transition
  - one action or set of atomic transitions
  - symbol
Petri Nets

- Arc Set \( A \subseteq (\text{\textit{P}} \times \text{\textit{T}}) \cup (\text{\textit{T}} \times \text{\textit{P}}) \)
  - set of directed arcs
  - edge of graph = arc
  - symbol \( \rightarrow \)

- Weight Function \( W = A \rightarrow \{1, 2, 3, \ldots \} \)
  - weights are associated with arcs

- Initial Marking \( M_0 = \text{\textit{P}} \rightarrow \{0, 1, 2, \ldots \} \)
  - the initial assignment of tokens to places
Petri Nets

- example
**Petri Nets**

- **Dynamic Behavior**
  - during simulation of a petri net the state of the net may change
  - change of state:
    » transitions can be enabled
    » enabled transitions may fire
    » firing transition changes the marking of the net
    » the marking is the “snap-shot” of all the tokens
**Petri Nets**

- **Firing rules**
  - A transition $T$ is said to be *enabled* if each input place $P$ is marked with at least $W(P,T)$ tokens
    - $W(P,T)$ is the weight of the arc from $P$ to $T$
  - An enabled transition may or may not fire (depending on whether or not the event actually takes place).
  - A *firing* of an enabled transition $T$ removes $W(P,T)$ tokens from each input place $P$ of $T$, and adds $W(T,P)$ tokens to each output place $P$ of $T$
    - $W(T,P)$ is the weight of the arc from $T$ to $P$
  - Common misconception: When a transition fires, it does **not** move tokens
    - i.e. the number of tokens in the system is not necessarily constant
Petri Nets

- Example: assume the following initial marking
  - Only one transition is enabled, i.e. $t_2$
Petri Nets

- Now several transitions are enabled, i.e. $t_1$, $t_3$, and $t_5$
- if $t_1$ fires first
Petri Nets

- if $t_3$ fires first
Petri Nets

- if $t_5$ fires first
- $t_3$ and $t_5$ are said to be in conflict
Petri Nets

- what could this Petri net represent?
Petri Nets

- Marking: Number and placement of tokens
  - let \( m_i \) = # of tokens in place \( p_i \)
  - then marking
    \[
    M = \{ m_1, m_2, ..., m_n \}
    \]
  - marking -- system state
  - Advantage: economy of model
    » e.g. assume net with 6 places
      ■ we limit each place to maximal 1 token
      ■ then there are \( 2^6 \) possible markings
      ■ \( \Rightarrow \) 64 states
      ■ thus Petri Nets are a lot smaller than state diagrams, i.e. Markov chains
Petri Nets

- Firing rules
  - transition 1, 3 and 4 are enabled
**Petri Nets**

- **Firing rules**
  - transition 4 fires

```
  p1 -> p2 -> t2 -> p3 -> p4 -> t3 -> t4
  p1 -> t1
  p4 -> t2
  p5 -> t4
```

```
  p1 -> p2 -> t2 -> p3 -> p4
  p1 -> t1
  p4 -> t2
  p5 -> t4
```
Petri Nets

- Firing rules
  - transition 1 fires
Petri Nets

- Firing rules
  - transition 3 fires