PETRI NETS

- A formal model of information flow.
- Used for modeling systems of parallel or concurrent activities.
- Created by Carl A. Petri in 1962.
- Graphs with two types of nodes: places and transitions.
PETRI NETS -- AN EXAMPLE

\[ P_0 \xrightarrow{t_1} P_2 \xrightarrow{t_2} P_3 \xrightarrow{t_3} P_4 \xrightarrow{t_4} P_5 \]

\[ P_2 \xrightarrow{t_5} P_6 \xrightarrow{t_6} P_7 \]
\textbf{Petri Net Structure -- Example}

\[
\mathcal{C} = (P, T, I, O)
\]

\[
P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7\}
\]

\[
T = \{t_1, t_2, t_3, t_4, t_5, t_6\}
\]

\[
I(t_1) = \{P_3\}
\]

\[
O(t_1) = \{P_1\}
\]

\[
I(t_2) = \{P_1\}
\]

\[
O(t_2) = \{P_2, P_3\}
\]

\[
I(t_3) = \{P_3, P_5\}
\]

\[
O(t_3) = \{P_4\}
\]

\[
I(t_4) = \{P_4\}
\]

\[
O(t_4) = \{P_5\}
\]

\[
I(t_5) = \{P_3, P_7\}
\]

\[
O(t_5) = \{P_6\}
\]

\[
I(t_6) = \{P_6\}
\]

\[
O(t_6) = \{P_7\}
\]
MARKED PETRI NETS -- AN EXAMPLE

P₀, P₅, P₇ each has a token.
PETRI NET COMPUTATION

- Tokens are moved by the firing of the transitions of the net.

- A transition must be enabled in order to fire.

A transition is enabled when all of its input places have a token in them.

- The transition fires by removing tokens from input places and generating tokens in output places.
MARKED PETRI NETS -- AN EXAMPLE COMPUTATION

- $t_2$ fires.
- Token at $P_1$ removed.
- New tokens at $P_2$, $P_3$.

$P_1$, $P_5$, $P_7$ each has a token.
PETRI NETS -- AN EXAMPLE COMPUTATION CONTINUED.
Petri Net Markings

A marking $\mu$ of a Petri net is an assignment of tokens to the places in that net.

The vector $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ gives for each place in the Petri net, the number of tokens in that place. $\mu$ can be seen as a function, s.t. $\mu(P_i) = \mu_i$.

Example

$\mu = (1, 0, 0, 0, 1, 0, 1)$

$\mu' = (0, 1, 1, 0, 1, 1, 0, 1)$

Before

After transition.
MARKED PETRI NET STRUCTURE

A Petri net \( C = (P, T, I, O) \) with a marking \( \mu \) becomes the marked Petri net

\[
M = (P, T, I, O, \mu)
\]

Since number of tokens is unbounded, there is a denumerable infinity number of markings for a Petri net.
STATE SPACE OF A PETRI NET

For a petri net $\mathcal{N}$ with $n$ places, the state space is the set of all possible markings, i.e. $\mathbb{N}^n$.

The next-state function is a partial function $\delta$, defined for enabled transitions $t_j$ in a marking $\mu$, s.t.

$$\delta(\mu, t_j) = \mu'$$

where $\mu'$ is the marking resulting from firing the transition.
Transition Sequences

- To record a Petri net execution, we use a sequence of markings:

\((\mu^0, \mu^1, \mu^2, \ldots)\)

and a sequence of transitions:

\((t_{j_0}, t_{j_1}, t_{j_2}, \ldots)\)

such that:

\[ \delta(\mu^k, t_{j_k}) = \mu^{k+1} \text{ for } k = 0, 1, 2, \ldots \]
Example of Transition Sequence

$(t_2, t_1, t_3, \ldots)$

$m^0 = (1, 0, 0, 0, 1, 0, 1)$

$m^1 = (0, 1, 1, 0, 1, 0, 1)$

$m^2 = (1, 0, 1, 0, 1, 0, 1)$

$m^3 = (1, 0, 0, 1, 0, 0, 1)$

$n_0 = 2$, $n_1 = 1$, $n_3 = 3$, $\ldots$

$\delta (m^0, t_2) = m^1$

$\delta (m^1, t_1) = m^2$

$\delta (m^2, t_3) = m^3$

and so on.
Reachability Set of a Petri Net

- $M'$ is immediately reachable from $M$ if
  $\exists t \in T, s.t. s(M, t) = M'$.
  (If we can fire some enabled transition in $M$ resulting in $M'$.)

- $M'$ is reachable from $M$ if it is immediately reachable from $M$ or it is reachable from any marking immediately reachable from $M$.
  (Reflexive transitive closure of immediately reachable)

- $R(M)$ is set of all reachable markings.
REACHABILITY TREE OF PETRI NETS

- Since often reachability set is infinite, the symbol \( w \) is used to represent an arbitrarily large number of tokens.

- The reachability tree contains nodes representing markings, and links representing transitions.
Reachability Tree Example

(1, 0, 1, 0)
  ↓ t3
(1, 0, 0, 1)
  ↓ t2
(1, w, 1, 0)
  ↓ t2
(1, w, 0, 1)
  ↓ t2
(1, w, 1, 0)
A Semaphore Modeling Example

Process 1
P(mutex);
"Critical Section";
V(mutex);

Process 2
P(mutex);
"Critical Section";
V(mutex);

P operation

V operation

Diagram:

- Pk
- P(s)
- V(s)
- Pkti
- S
Mutual Exclusion using a Semaphore
Petri Nets as System Descriptions

- Atomic (local) states/places \( P \)
- Atomic (local) transitions \( T \)

\[ P \cap T = \emptyset \] (disjoint sets)

- Distributed (global) state case
  set of conditions holding concurrently
- Distributed (global) transition step
  set of events occurring concurrently

Transition relation
specifies how cases are transformed into cases by the occurrence of steps