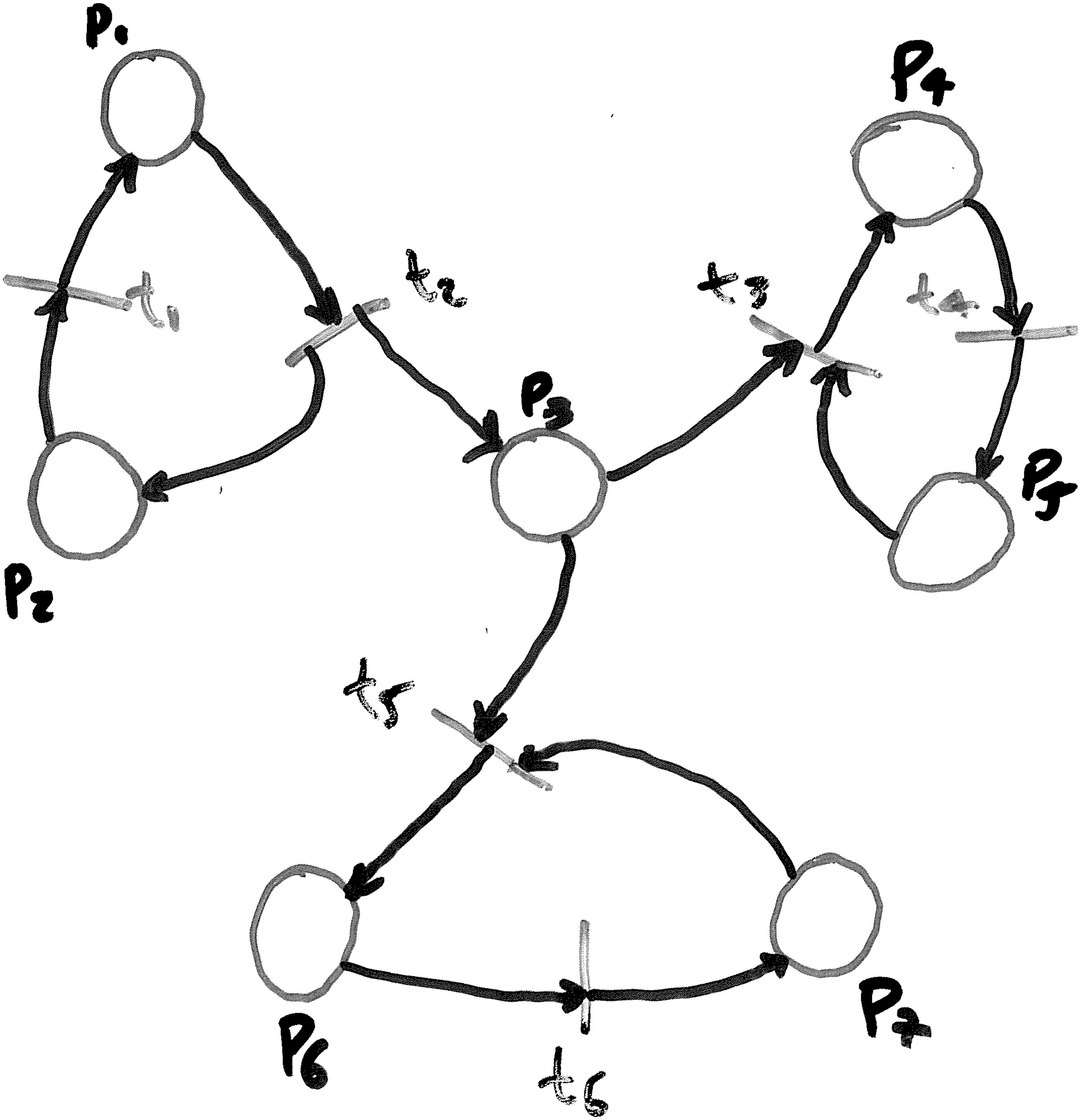
PETRI NETS

- 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



PETRI NET STRUCTURE
-- EXAMPLE

$$C = (P, T, I, O)$$

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$$

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

$I(t_1) = \{p_2\}$ $\quad$ $O(t_1) = \{p_1\}$

$I(t_2) = \{p_1\}$ $\quad$ $O(t_2) = \{p_2, p_3\}$

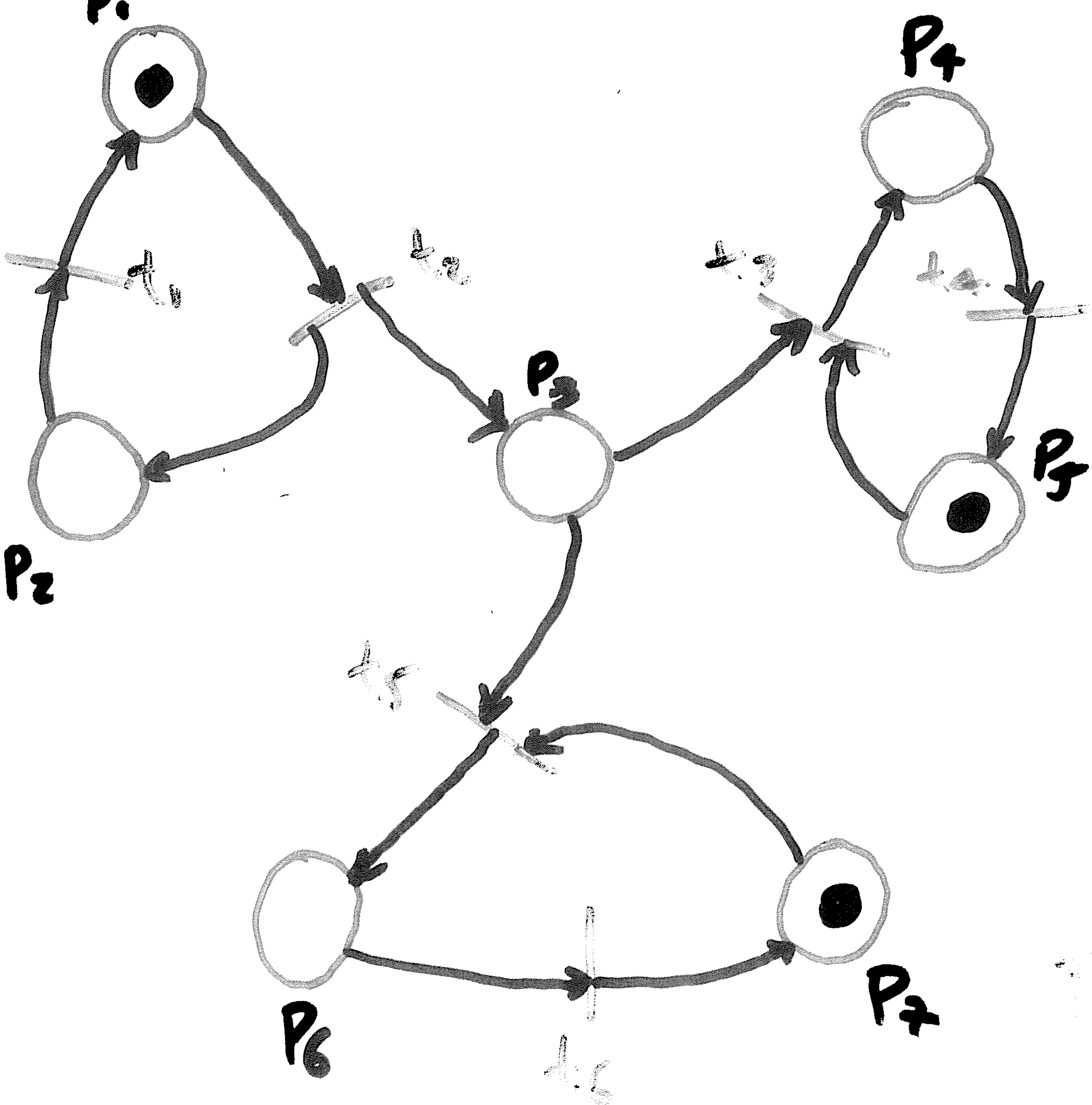
$I(t_3) = \{p_3, p_5\}$ $\quad$ $O(t_3) = \{p_4\}$

$I(t_4) = \{p_4\}$ $\quad$ $O(t_4) = \{p_5\}$

$I(t_5) = \{p_3, p_7\}$ $\quad$ $O(t_5) = \{p_6\}$

$I(t_6) = \{p_6\}$ $\quad$ $O(t_6) = \{p_7\}$

MARKED PETRI NETS -- AN EXAMPLE



P_1, P_5, P_7 each has a token.

Petri Net Computation

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

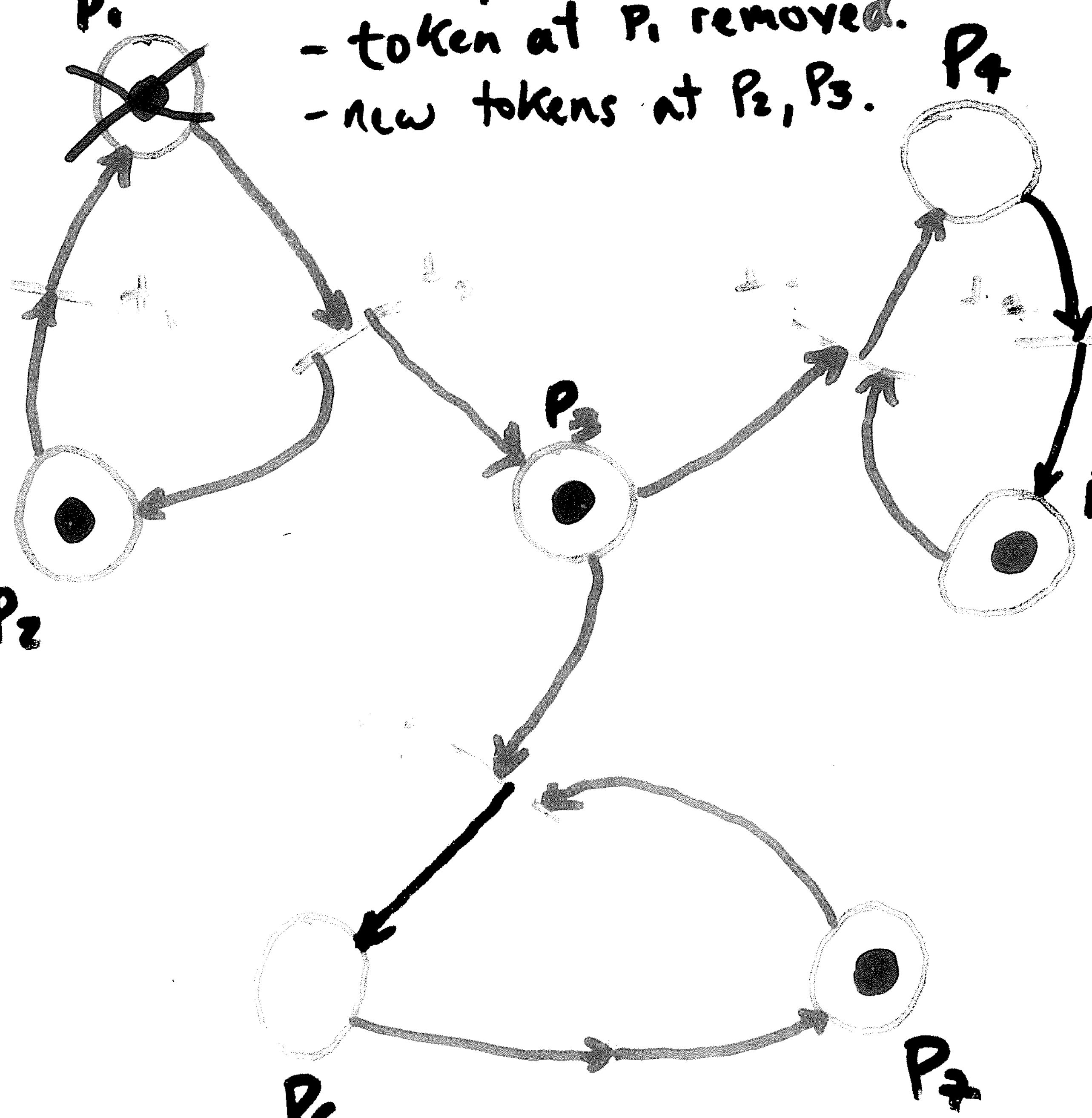
- A transition must be enabled in order to fire.

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

- The transition fires by removing tokens from input places & generating tokens in output places.

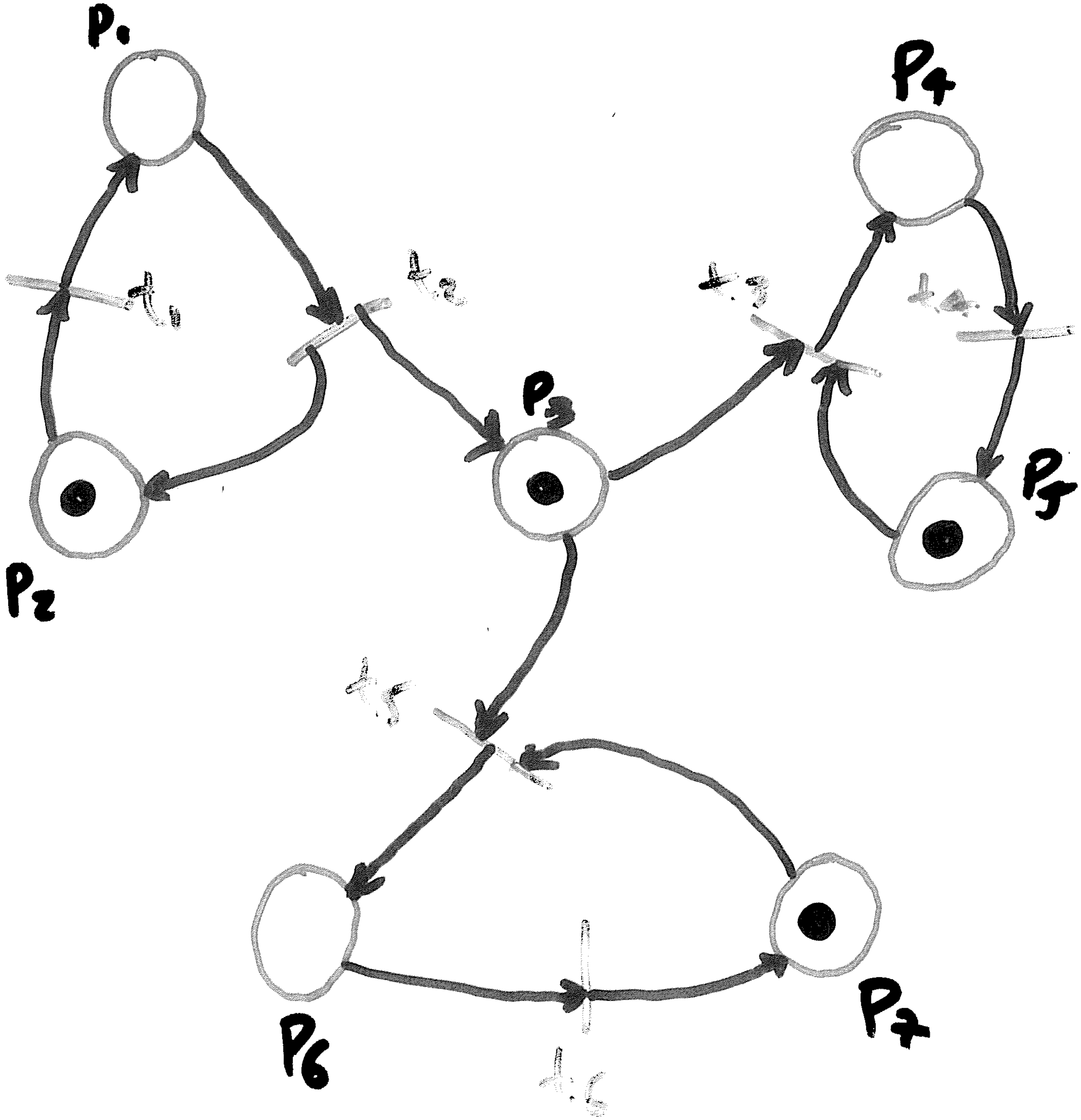
MARKED PETRI NETS -- AN EXAMPLE COMPUTATION

- 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 μ 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.

μ can be seen as a function, s.t. $\mu(P_i) = \mu_i$

EXAMPLE

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

$\mu' = (0, 1, 1, 0, 1, 0, 1)$ After transition.

MARKED PETRI NET STRUCTURE

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

$$M = (P, T, I, O, \mu)$$

Since number of tokens is unbounded, there is a denumerable infinite number of markings for a Petri net.

STATE SPACE OF A PETRI NET

For a Petri net 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 δ, defined for enabled transitions t_j in a marking μ, s.t.

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

where μ' is the marking resolting 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

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

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

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

$\mu^2 = (1, 0, 1, 0, 1, 0, 1)$

$\mu^3 = (1, 0, 0, 1, 0, 0, 1)$

$j_0 = 2,\ j_1 = 1,\ j_3 = 3, \ldots$

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

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

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

and so on.

REACHABILITY SET OF A PETRI NET

- μ' is immediately reachable from μ if
  $$\exists t \in T, \text{ s.t. } \delta(\mu, t) = \mu'.$$
  (if we can fire some enabled transition in μ resulting in μ'.)

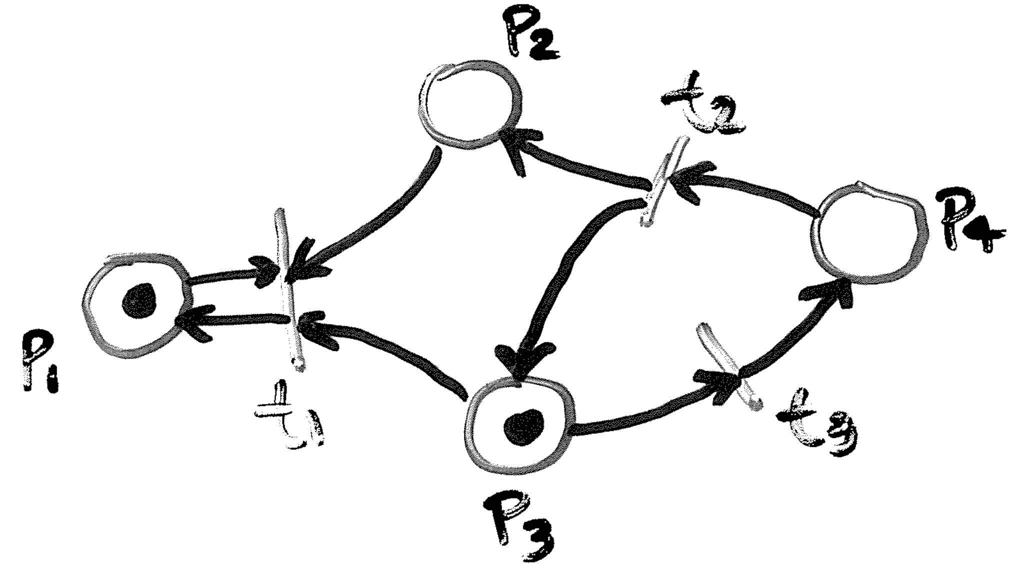
- μ' is reachable from μ if it is immediately reachable from μ or it is reachable from any marking immediately reachable from μ.
  (reflexive transitive closure of immediately reachable)

- $R(M)$ is set of all reachable markings

REACHABILITY TREE OF PETRI NETS

- Since often reachability set is infinite, the symbol ω is used to represent an arbitrarily large number of tokens.
- The reachability tree contains nodes representing markings, and links representing transitions.

REACHABILITY TREE EXAMPLE



$$
\begin{tikzcd}[column sep=small]
 & (1,0,1,0) \arrow[d, "t_3"] & \\
 & (1,0,0,1) \arrow[d, "t_2"] & \\
 & (1,\omega,1,0) \arrow[dl, "t_1"'] \arrow[dr, "t_3"] & \\
(1,\omega,0,0) & & (1,\omega,0,1) \arrow[d, "t_2"] \\
 & & (1,\omega,1,0)
\end{tikzcd}
$$

A SEMAPHORE MODELING EXAMPLE

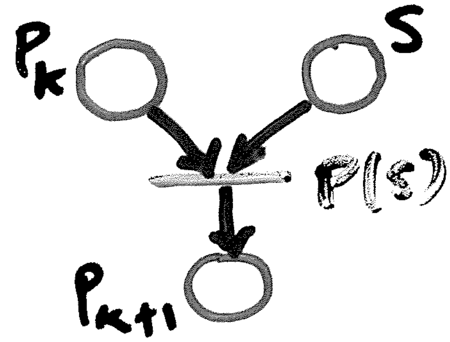
Process 1

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

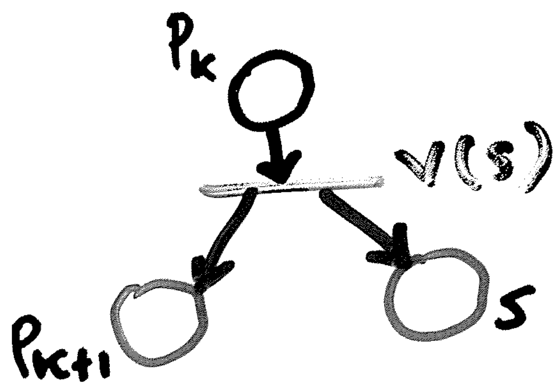
Process 2

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

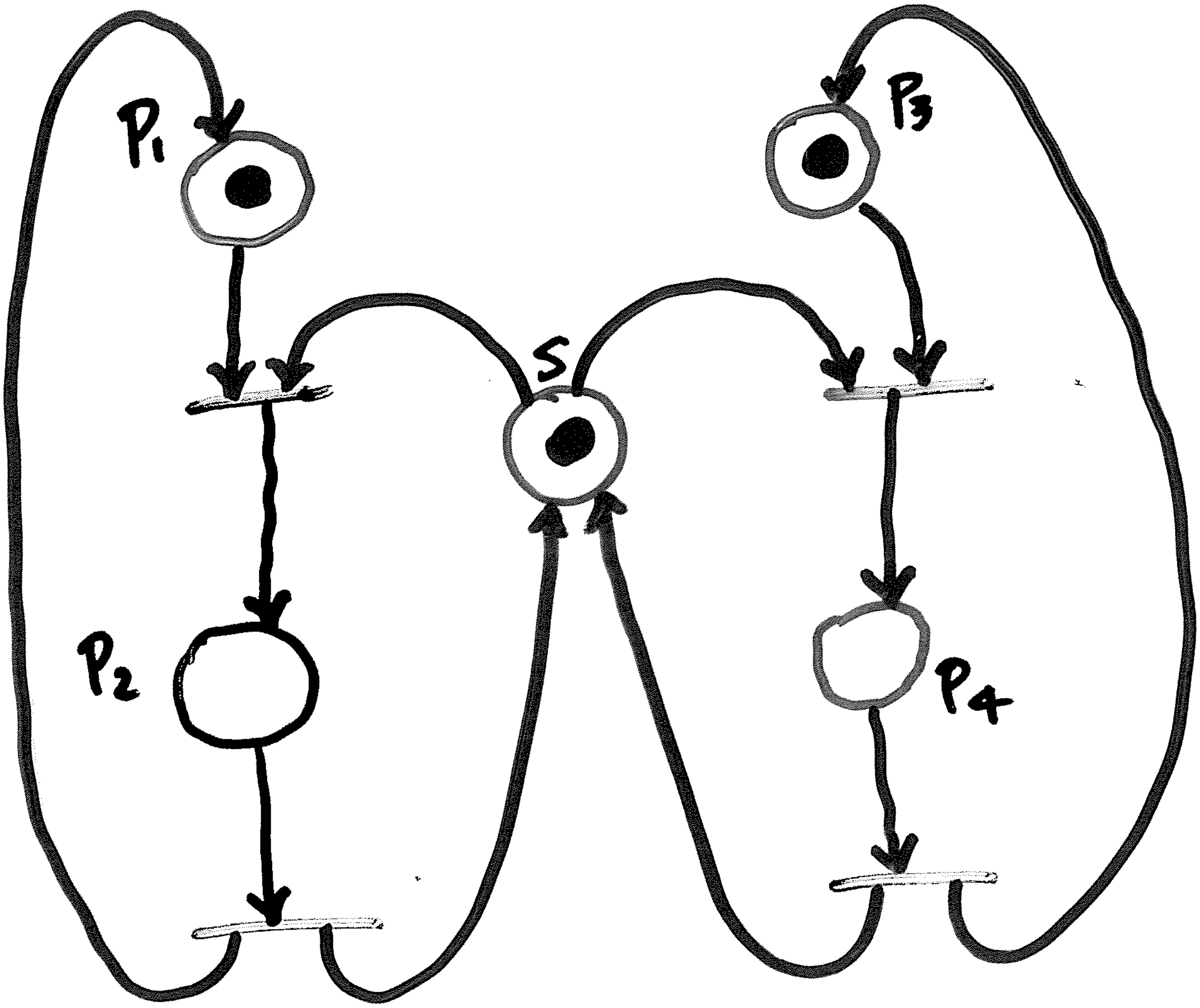
P operation



V operation



MUTUAL EXCLUSION USING A SEMAPHORE



PETRI NETS AS SYSTEM DESCRIPTIONS

- Atomic (local) states/places P
  conditions
- Atomic (local) transitions T
  events

$$P \cap T = \emptyset \quad \text{(disjoint sets)}$$

- 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