Elementary Petri Nets

SE 3BB4

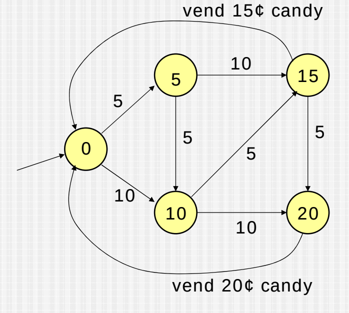
Ryszard Janicki

Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada

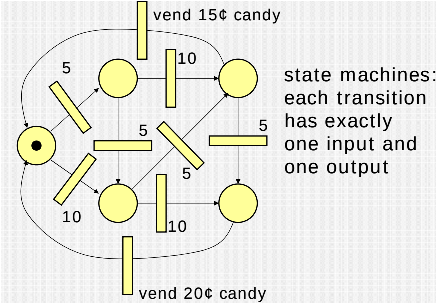
From State Machines to Elementary Petri Nets

Elementary Petri Nets are directed bipartite graphs with

- *places*, represented by circles or ovals (represent some type of resource)
- *transitions*, represented by rectangles or lines (consume and produce resources)
- *arcs* (from places to transitions or transitions to places)
- *tokens*, placed in places
- *initial marking*, tokens is some places



Finite State Machine

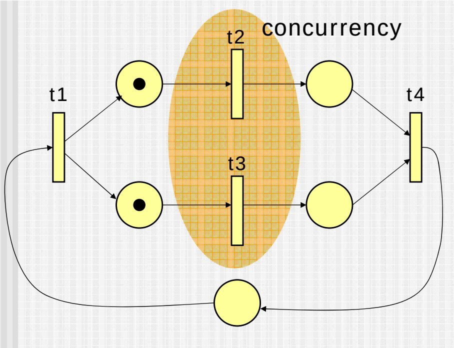


Petri Net

Modeling Concurrency

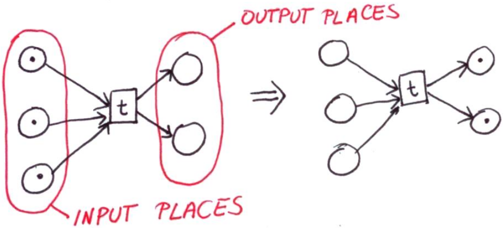
Elementary Petri Nets are directed bipartite graphs with

- *places*, represented by circles or ovals (represent some type of resource)
- *transitions*, represented by rectangles or lines (consume and produce resources)
- *arcs* (from places to transitions or transitions to places)
- *tokens*, placed in places
- *initial marking*, tokens is some places

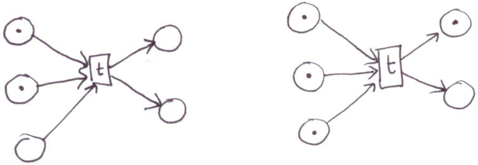


Firing Rules for Elementary Petri Nets

- A transition t can be *fired* if and only if it has tokens in all its input places and all output places are empty.
- After firing t, all its input places become empty and all its output places contain tokens.



Before fining t After firing t

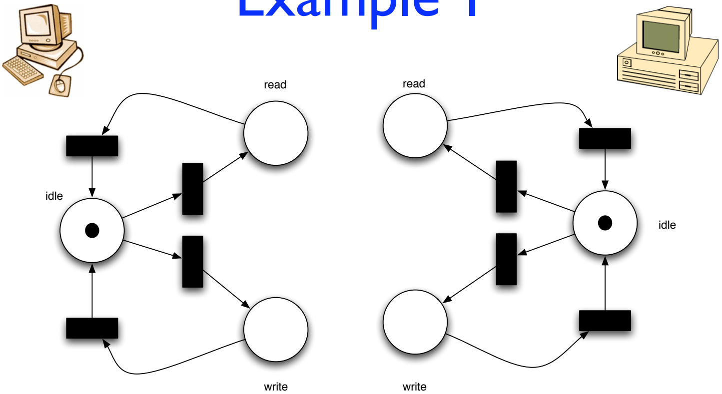


t can not be fired

Modeling Mutual Exclusion

- Two computers, one printer/data base, etc.
- Without any synchronization, individual viewpoints of computers.

Example I

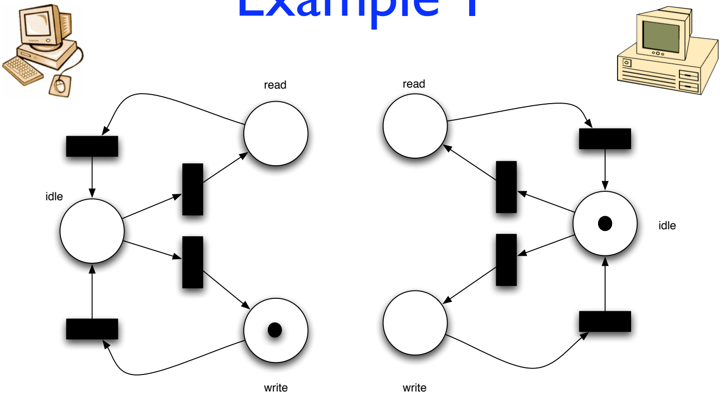


The token tells us the state of the process

Modeling Mutual Exclusion

- Two computers, one printer/data base, etc.
- Without any synchronization, individual viewpoints of computers.

Example I

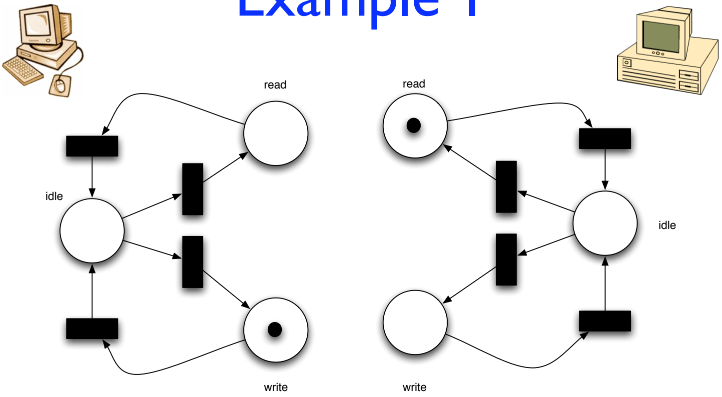


The token tells us the state of the process

Modeling Mutual Exclusion

- Two computers, one printer/data base, etc.
- Without any synchronization, individual viewpoints of computers.

Example I

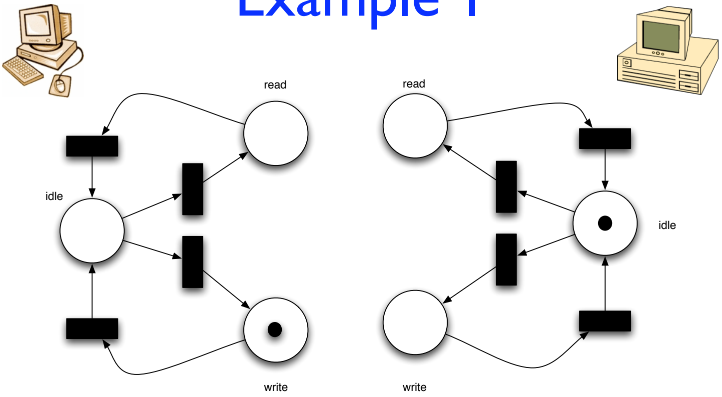


The token tells us the state of the process

Modeling Mutual Exclusion

- Two computers, one printer/data base, etc.
- Without any synchronization, individual viewpoints of computers.

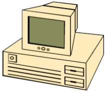
Example I

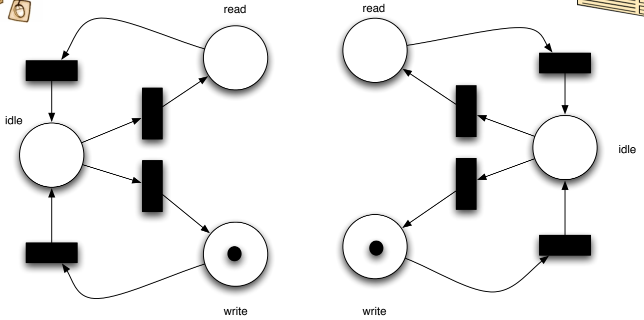


The token tells us the state of the process

- Two computers, one printer/data base, etc.
- Without any synchronization, individual viewpoints of computers.
- PROBLEM, both computers want to write but there is only one printer/data base, etc.

Example I



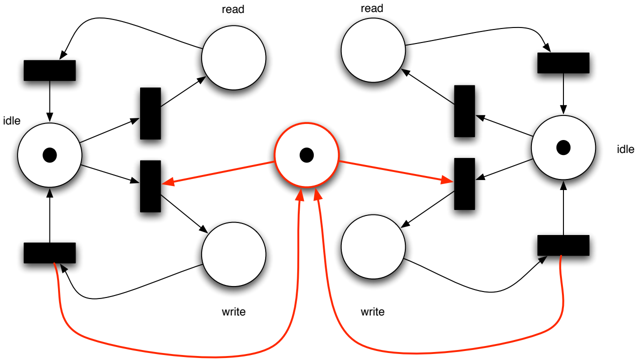


The token tells us the state of the process

Modeling Mutual Exclusion

- Two computers, one printer/data base, etc.
- Synchronization is added.

Example I

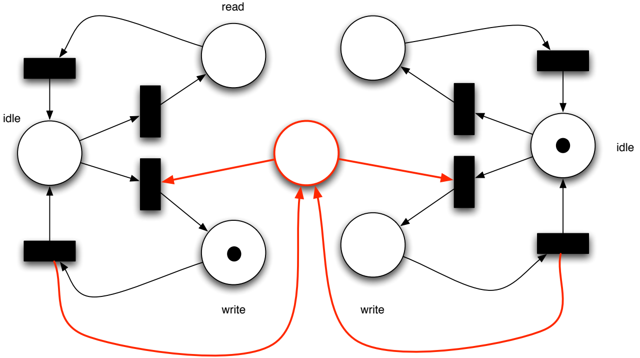


Add a lock to ensure mutual exclusion

Modeling Mutual Exclusion

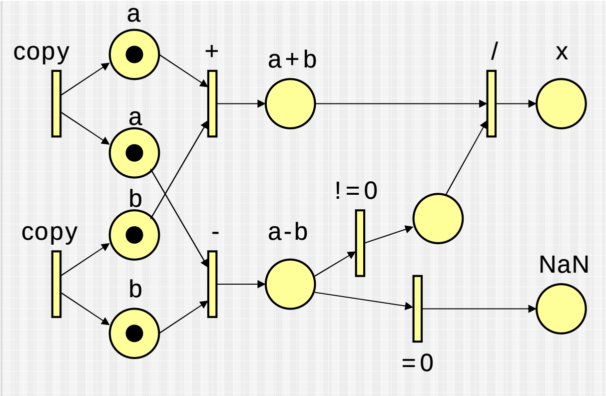
- Two computers, one printer/data base, etc.
- Synchronization is added.

Example I



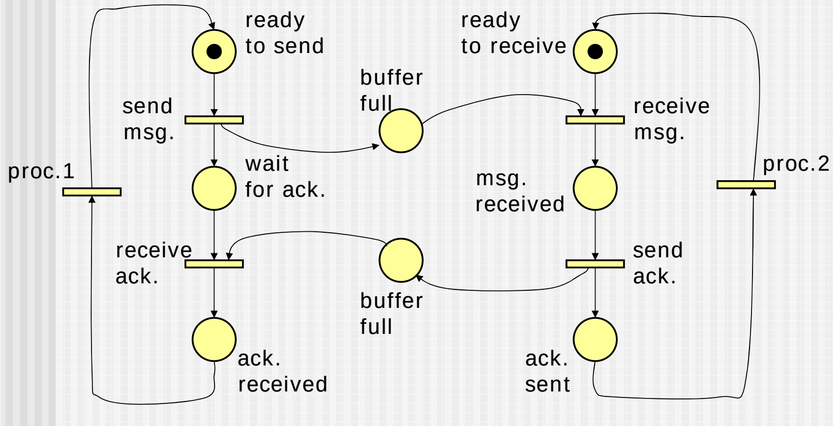
Modeling Dataflow Computation

- Petri nets allow modeling **without** decomposing the whole system into sequential component!
- $x = (a + b)/(a - b)$



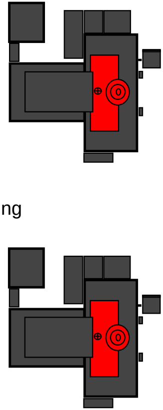
- The two *copy* transitions can be removed, they represent inputs from environment.

Modeling Communication Protocols



Loading and Unloading Two Machines

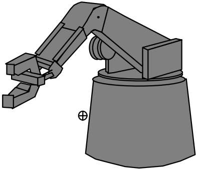
Machine 1



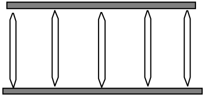
Machine 2

Machine States:
Loading
Processing
Waiting for unloading
Unloading

Robot

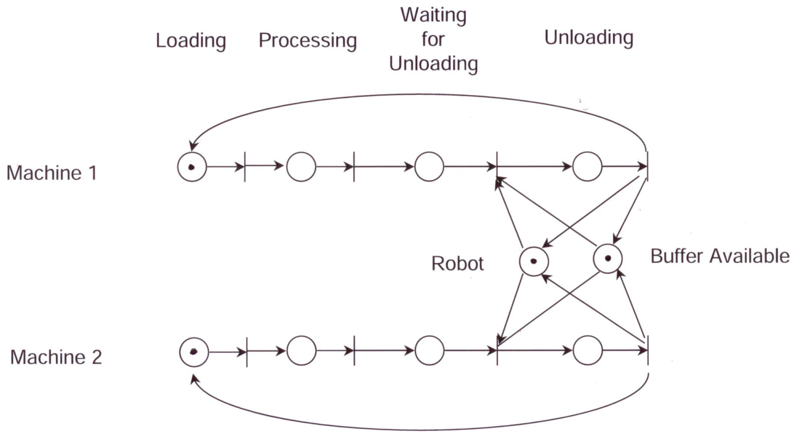


Buffer



Buffer State:
Space availability

Loading and Unloading Two Machines

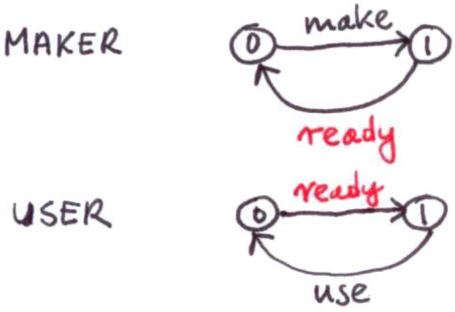


Composition of LTS (Maker-User Example)

$$MAKER = make \rightarrow ready \rightarrow MAKER$$

$$USER = ready \rightarrow use \rightarrow USER$$

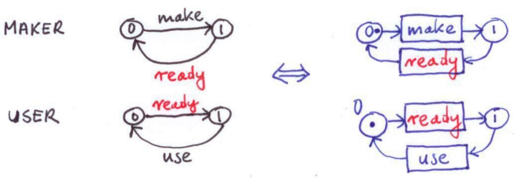
$$\| \; MAKER\_USER = Maker \parallel USER$$



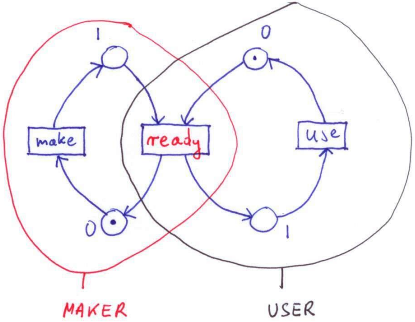
LTS:

LTS via Elementary Petri Nets

1. Represent each LTS as a Petri Net

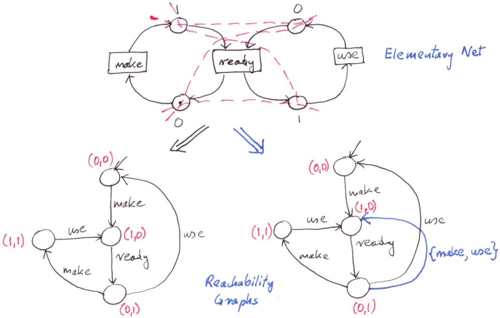


2. 'Glue' together both nets through the common transition *ready*.



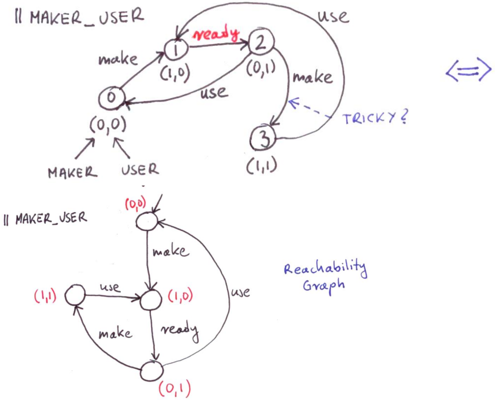
Reachability Graphs

- Reachability graphs are finite state machines that represent the behaviours of Petri nets.
- Each state of the reachability graph represent a 'marking' of Petri net.
- Simultaneous executions (*steps*), like $\{make, use\}$, may be allowed.



Reachability Graphs and $\|$ operator

- Reachability graphs are the same as final LTS obtained via operator '$\|$'!
- Getting final LTS vis Petri nets is more natural than via standard procedure!



Formal Definition (1)

Definition

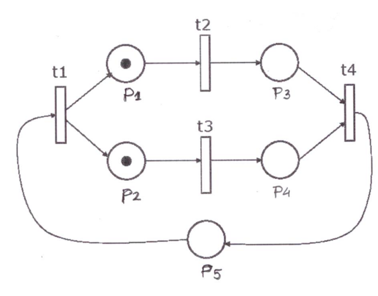
An **Elementary Net** is a tuple

$$\mathtt{N} = (P, T, F, C_{init})$$

such that

1. P and T are finite and disjoint sets of places and transitions represented, respectively, as circles and rectangles;
2. $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation of $\mathtt{N}$ - represented as directed arcs between places and transitions;
3. $C_{init} \subseteq P$ is the initial marking (or initial configuration) of $\mathtt{N}$.

Example

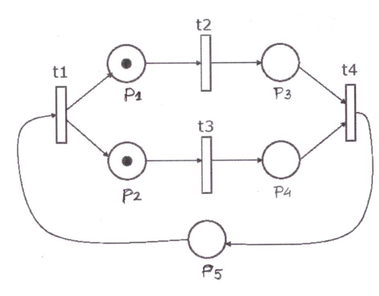


$$
\begin{aligned}
P &= \{p_1, p_2, p_3, p_4, p_5\},\\
T &= \{t_1, t_2, t_3, t_4\},\\
F &= \{(p_1, t_2), (p_2, t_3), (p_3, t_4), (p_4, t_4), (p_5, t_1),\\
&\qquad (t_1, p_1), (t_1, p_2), (t_2, p_3), (t_3, p_4), (t_4, p_5)\},\\
C_{init} &= \{p_1, p_2\}.
\end{aligned}
$$

Formal Definition (2)

- For every $x \in P \cup T$, the set ${}^\bullet x = \{y \mid (y,x) \in F\}$ denotes the *input* nodes of x, and
- the set $x^\bullet = \{y \mid (x,y) \in F\}$ denotes the *output* nodes of x.
- The dot-notation extends to sets in the natural way, e.g. the set $X^\bullet$ comprises all outputs of the nodes in X.
- We often (but not always) assume that for every $t \in T$, both ${}^\bullet t$ and $t^\bullet$ are non-empty and disjoint.

Example



$^\bullet p_1 = \{t_1\}$, $^\bullet p_2 = \{t_1\}$, $^\bullet p_3 = \{t_2\}$, $^\bullet p_4 = \{t_3\}$, $^\bullet p_5 = \{t_4\}$,
$p_1^\bullet = \{t_2\}$, $p_2^\bullet = \{t_3\}$, $p_3^\bullet = \{t_4\}$, $p_4^\bullet = \{t_4\}$, $p_2^\bullet = \{t_1\}$,
$^\bullet t_1 = \{p_5\}$, $^\bullet t_2 = \{p_1\}$, $^\bullet t_3 = \{p_2\}$, $^\bullet t_4 = \{p_3, p_4\}$,
$t_1^\bullet = \{p_1, p_2\}$, $t_2^\bullet = \{p_3\}$, $t_3^\bullet = \{p_4\}$, $t_4^\bullet = \{p_5\}$,
$^\bullet\{p_1, p_4\} = \{t_1, t_3\}$, $^\bullet\{p_1, p_2\} = \{t_1\}$,
$\{p_1, p_4\}^\bullet = \{t_2, t_4\}$, $\{p_1, p_2\}^\bullet = \{t_2, t_3\}$,
$^\bullet\{t_1, t_3\} = \{p_2, p_5\}$, $^\bullet\{t_2, t_3\} = \{p_1, p_2\}$,
$\{t_1, t_3\}^\bullet = \{p_1, p_2, p_4\}$, $\{t_2, t_3\}^\bullet = \{p_3, p_4\}$.

- A transition t is enabled at a configuration C if ${}^\bullet t \subseteq C$ and $t^\bullet \cap C = \emptyset$.
- An enabled transition t can fire leading to a new configuration $C' = (C \setminus {}^\bullet t) \cup t^\bullet$.
- We denote this by $C[t\rangle C'$, or by $C[t\rangle_{\mathrm{N}} C'$, if C, C' and t may belong to different nets.
- We will also write $C[t_1 \dots t_n\rangle C'$ if $C[t_1\rangle C_1 \dots C_{n-1}[t_n\rangle C'$ for some configurations $C_1, \dots, C_{n-1}$.

Definition

A **firing sequence** of an Elementary Petri Net is any sequence of transitions $t_1, \dots, t_n$ for which there are markings $C_1, \dots, C_n$ satisfying:

$$C_{init}[t_1\rangle C_1[t_2\rangle C_2 \dots [t_n\rangle C_n.$$

Step-sequence Semantics

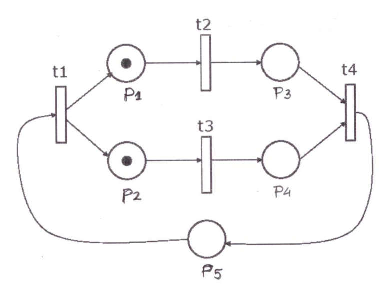
- Let $A \subseteq T$ be a non-empty set such that for all distinct $t_1, t_2 \in A$:
$$(t_1^\bullet \cup {}^\bullet t_1) \cap (t_2^\bullet \cup {}^\bullet t_2) = \emptyset.$$
- Then A is enabled at a marking C if ${}^\bullet A \subseteq C$ and $A^\bullet \cap C = \emptyset$.
- We also denote this by $C[A\rangle C'$, or $C[A\rangle_{\mathrm{N}} C'$ when C, C' and A may belong to different nets, where $C' = (C \setminus {}^\bullet A) \cup A^\bullet$.

Definition

A **firing step sequence** is a sequence of sets (or steps) $A_1, \ldots, A_n$ for which there are markings $C_1, \ldots, C_n$ satisfying:

$$C_{init}[A_1\rangle C_1[A_2\rangle C_2 \ldots [A_n\rangle C_n.$$

Example



- Some firing sequences: $t_2 t_3 t_4 t_1$ since $\{p_1, p_2\}[t_2\rangle\{p_2, p_3\}[t_3\rangle\{p_3, p_4\}[t_4\rangle\{p_5\}[t_1\rangle\{p_1, p_2\}$, $t_3 t_2 t_4 t_1$ since $\{p_1, p_2\}[t_3\rangle\{p_1, p_4\}[t_2\rangle\{p_3, p_4\}[t_4\rangle\{p_5\}[t_1\rangle\{p_1, p_2\}$.
- A firing step-sequence: $\{t_2, t_3\}\{t_4\}\{t_1\}$ since $\{p_1, p_2\}[\{t_2, t_3\}\rangle\{p_3, p_4\}[\{t_4\}\rangle\{p_5\}[\{t_1\}\rangle\{p_1, p_2\}$.