Slides contain material from www.yawlfoundation.org
Developed by Wil van der Aalst, Michael Adams, Arthur ter Hofstede, Nick Russel

PETRI NETS

Business 2710 – Class 9

Learning Objectives

- Be able to use Petri Nets to model simple business processes
- Understand the process execution semantics of simple Plate/Transition Petri Nets

Reisig, W. and Rozenberg, G. (1998) Information introduction to Petri Nets. In: Reisig, W. And Rozenberg, G. (eds.) *Lectures on Petri Nets – Advances in Petri Nets. Lecture Notes on Computer Science,* Volume 1491, Springer Verlag.

Petri Nets

- Technique for the description and analysis of concurrent systems
- Benefits:
  - Graphical notation
  - Formal
  - Based of a few simple concepts, yet expressive
  - Many analysis techniques exist
- Many extensions and variants have been defined over the years

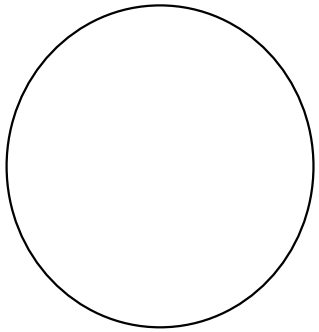
Applications for Workflow Management

- There are two main uses of Petri nets for workflows:
  - Specifications of workflows
  - Formal analysis of workflows (semantics, analysis of important properties)

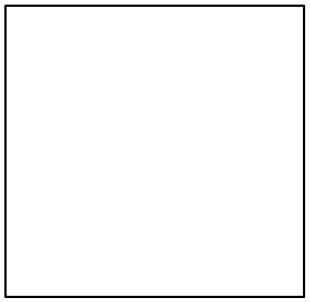
Petri Nets: Definitions

- Petri Nets consist of **places** and **transitions**
- Places and transitions are connected by arrows (arcs) and **must alternate each other**
- Places are represented by circles, transitions by thick bars or boxes
- Formally a Petri net N is a triple (P, T, F) where
  - P is a finite set of places
  - T is a finite set of transitions
  - $F \subseteq (P \times T \cup T \times P)$ is the flow relation ("the arrows")

Petri Nets: Graphical Symbols



Place



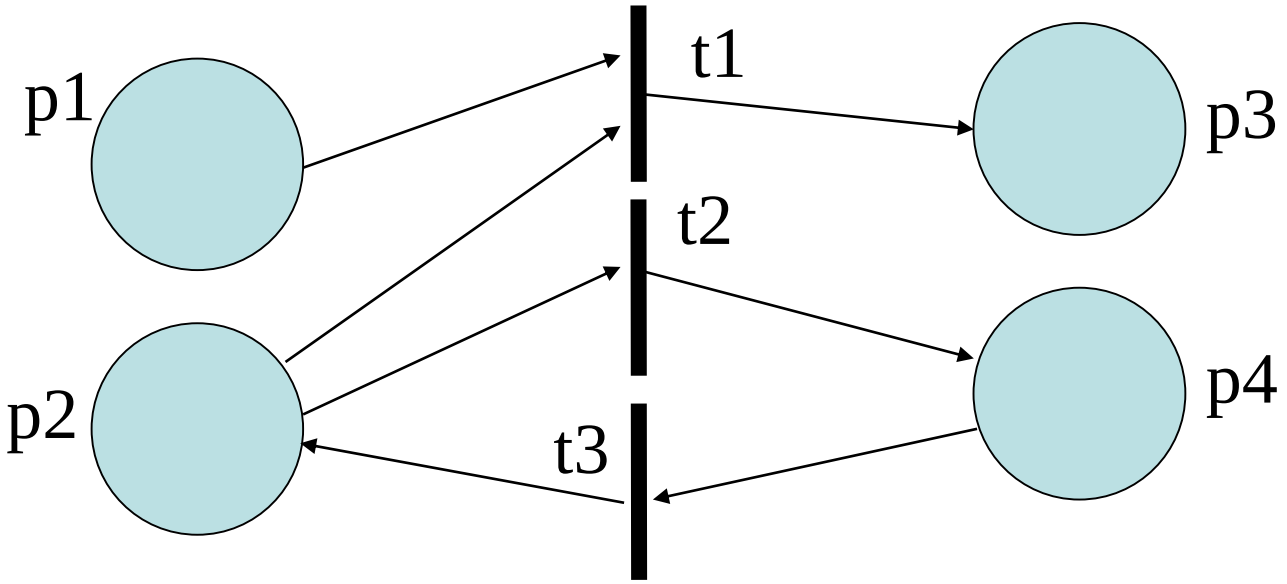
Transition

Arc

Petri Nets: Definitions

- Pre- and post-sets of places are defined as:
  - $p\bullet = \{t \in T \mid (p, t) \in F\}$, $\bullet p = \{t \in T \mid (t, p) \in F\}$
- Pre- and post-sets of transitions are defined as:
  - $t\bullet = \{p \in P \mid (t, p) \in F\}$, $\bullet t = \{p \in P \mid (p, t) \in F\}$

Petri Net: Example



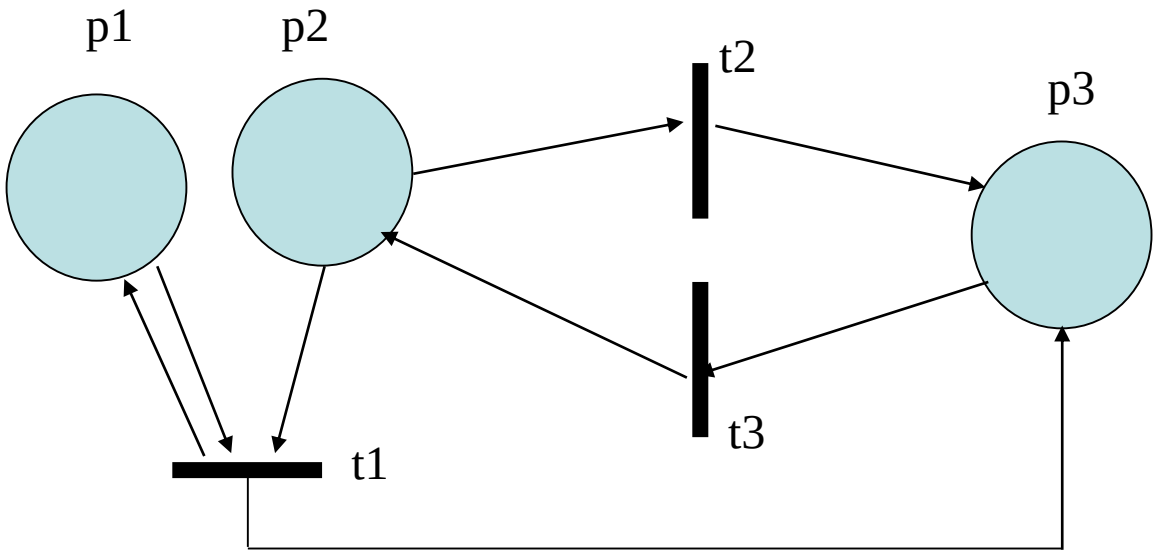
$P = \{p_1, p_2, p_3, p_4\}$

$T = \{t_1, t_2, t_3\}$

$F = \{(p_1, t_1), (p_2, t_1), (t_1, p_3), (p_2, t_2), (t_2, p_4), (p_4, t_3), (t_3, p_2)\}$

$t_1 \bullet = \{p_3\}$; $\bullet t_1 = \{p_1, p_2\}$; $\bullet p_2 = \{t_3\}$; $\bullet p_1 = \{\}$; $p_2 \bullet = \{t_1, t_2\}$

Petri Nets: Example



P = ...

T = …

F = ...

$t_1 \bullet$ = … ; $\bullet\, t_1$ = … ; $\bullet\, p_2$ = … ; $p_2 \bullet$ = ...

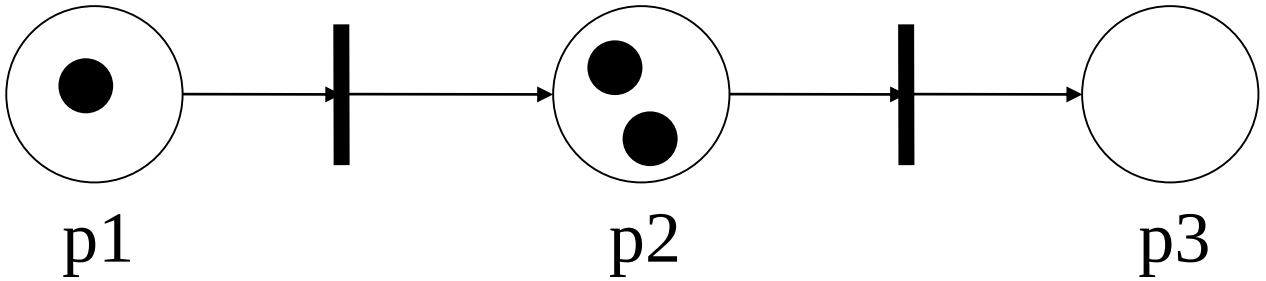
Markings

- Markings assign tokens (graphically represented as black dots) to places; they represent the state of the system

- Formally a marking M of a Petri net N=(P, T, F) is the assignments of the number of tokens for each place in the Petri net.

Markings

- The marking below is $\{(p_1,1),(p_2,2),(p_3,0)\}$. This can be shortened to $p_1 + 2p_2$.

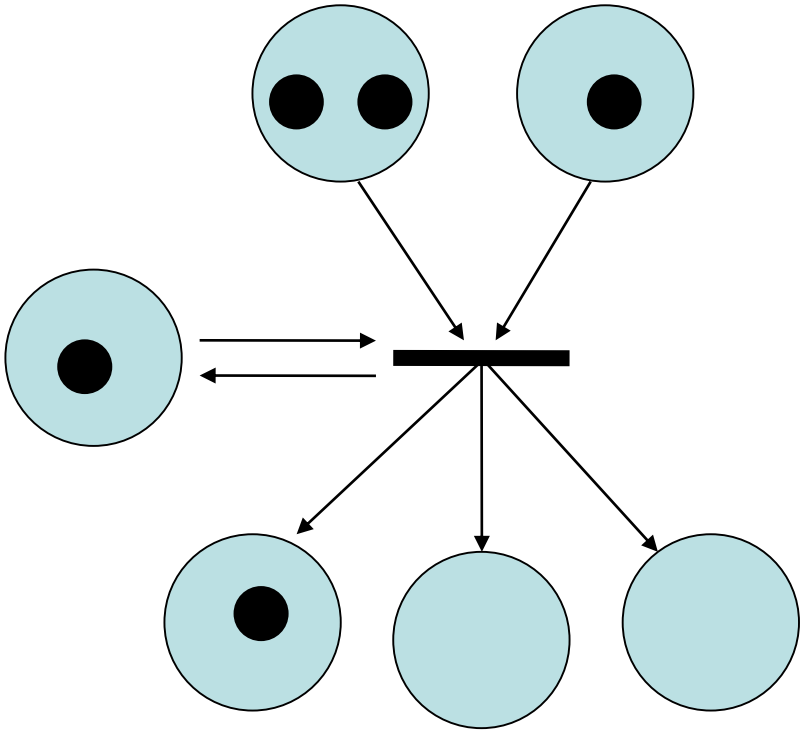


- A marking M is “greater or equal” than a marking M' (M≥M') if, for all places, the number of tokens on a place in M is greater or equal than the number of tokens on that place in M' (Analogous for “greater”, M>M')

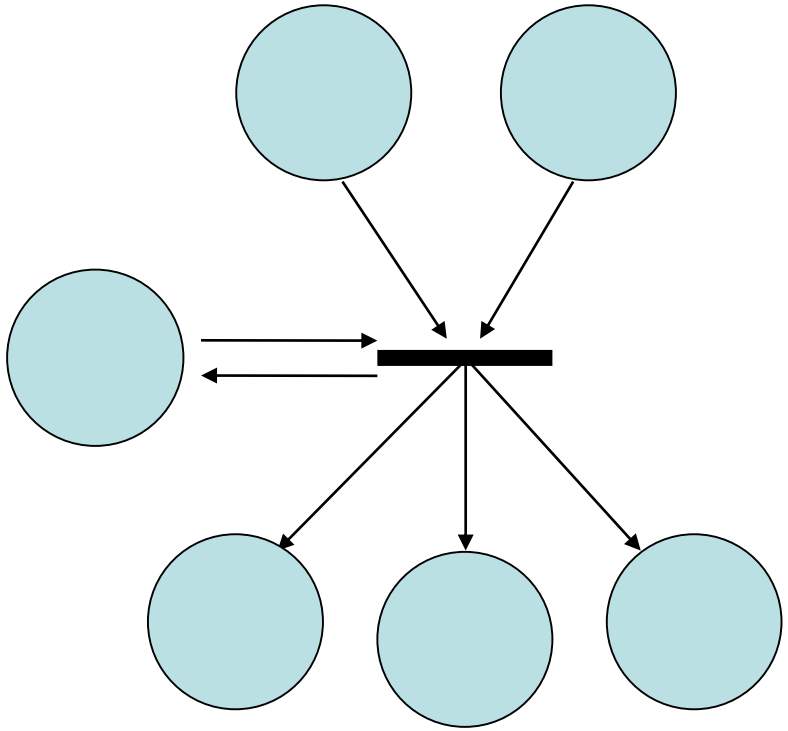
Enabled Transitions

- Transitions can change a marking by **firing**
- Only **enabled** transitions may **fire**.
  - A transition is **enabled** if each of its input places contains at least one token
- In a marking, any enabled transition may **fire**
  - Randomly chosen, or user may choose
- When a transition fires, a token is removed from each of its input places and a token is produced for each of its output places
  - The number of tokens in the Petri Net can increase or decrease!

Firing a Transition: Example

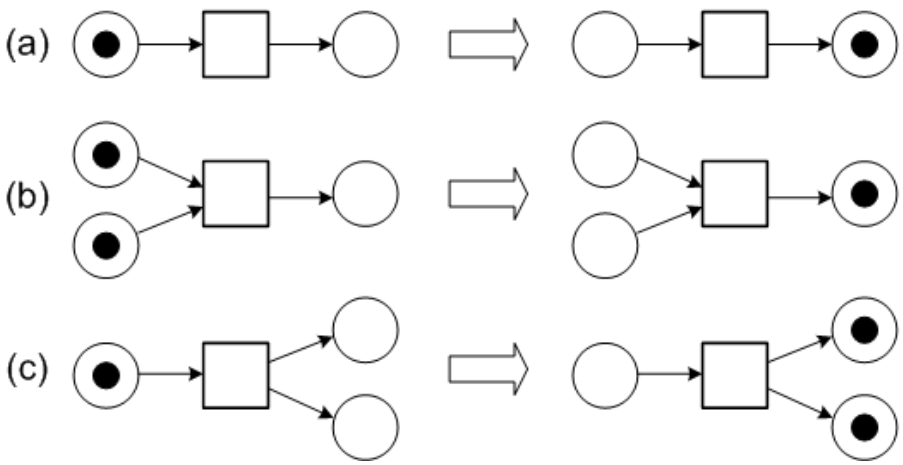


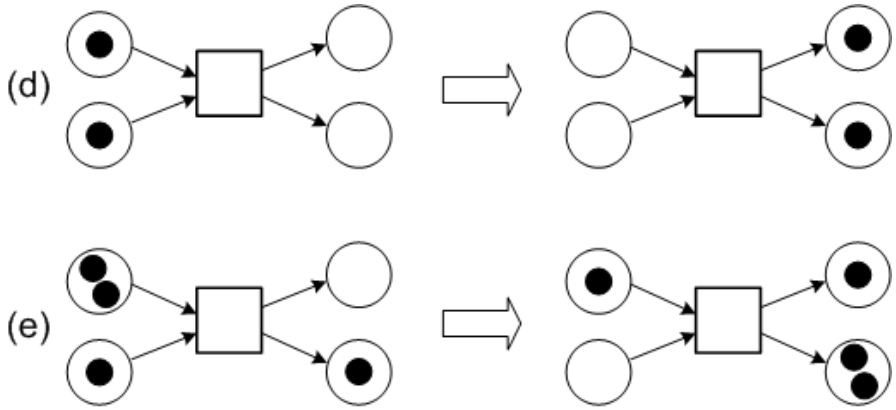
Before



After??

Firing Transitions: Further Examples



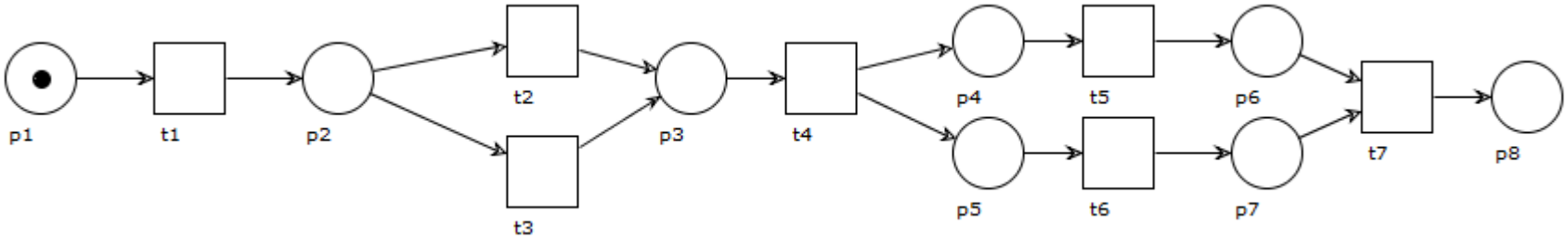


Behaviour

- Petri nets describe the **behaviour** of a system (e.g. that of a business process)
- **Behaviour** is the set of possible sequences of transitions (e.g. activities in a business process)
- When modeling a business process
  - Focus on the transitions
  - Interpreting (e.g. naming) the places and tokens is not important. They are only there to ensure the desired sequence of transitions

Trace

- A trace is a sequence of transition firings
- Example:



- The sequence t1 t2 t4 t5 t6 t7 is a valid trace for this model
- The sequence t1 t2 t3 t7 t7 t6 is not a valid trace for this model

Behaviour

- A Petri net model *correctly describes* the intended system behaviour if and only if
  1. All valid traces for the model are really possible behaviour for the intended system
  2. All invalid traces for the model are really impossible behaviour for the intended system
  3. All really possible behaviour for the intended system are valid traces in the model
  4. All really impossible behaviour for the intended system are invalid traces in the model

Example Petri Net

Candy Vending Machine