

Lecture 17: Petri-Nets

Kenneth M. Anderson
Foundations of Software Engineering
CSCI 5828 - Spring Semester, 1999

Today's Lecture

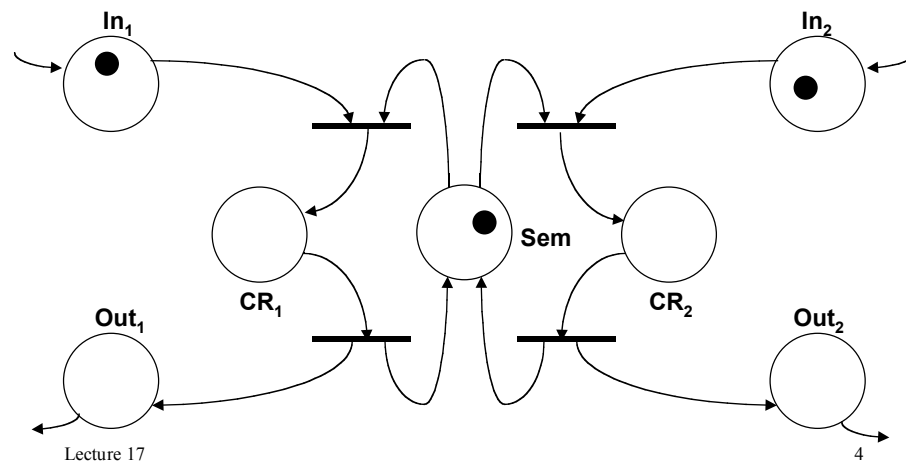
- Introduce the Petri Net Formalism
 - Present several examples

Petri Nets

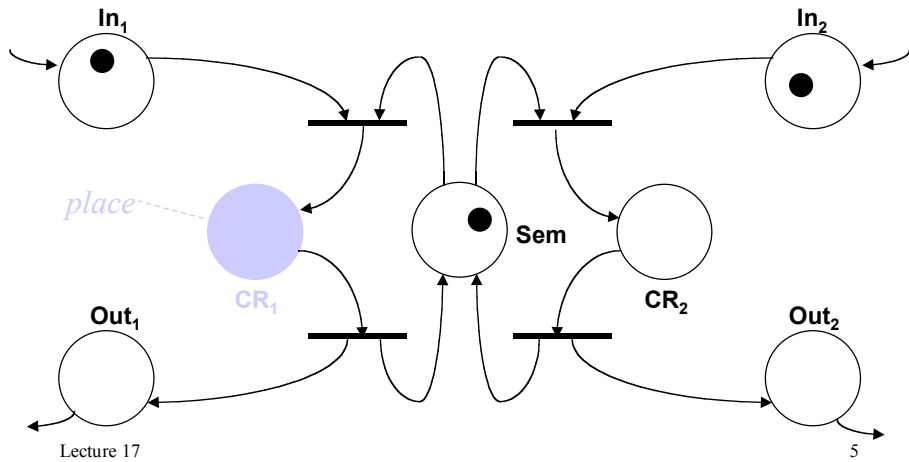
- Formal Definition

$N = \{P, T, A, M_0\}$, where
P is a finite set of *places*
T is a finite set of *transitions*
A is a finite set of *arcs (arrows)*
*M*₀ is the *initial marking* of *N*

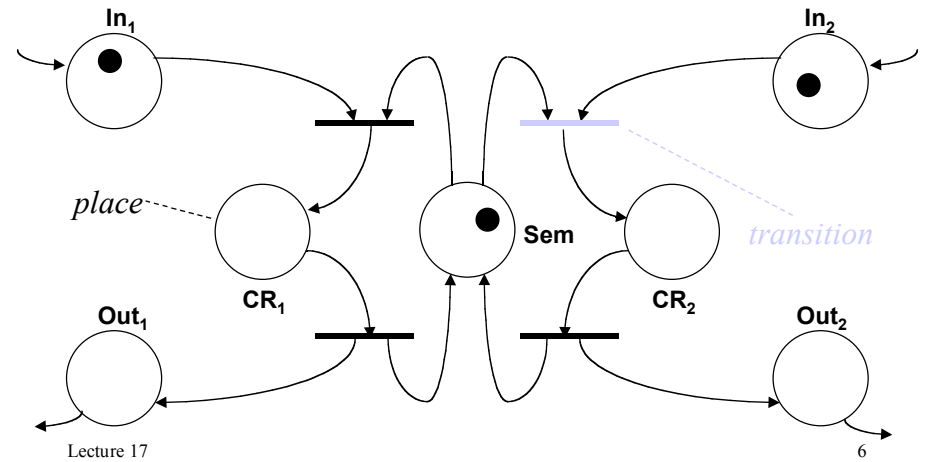
Graphical Representation



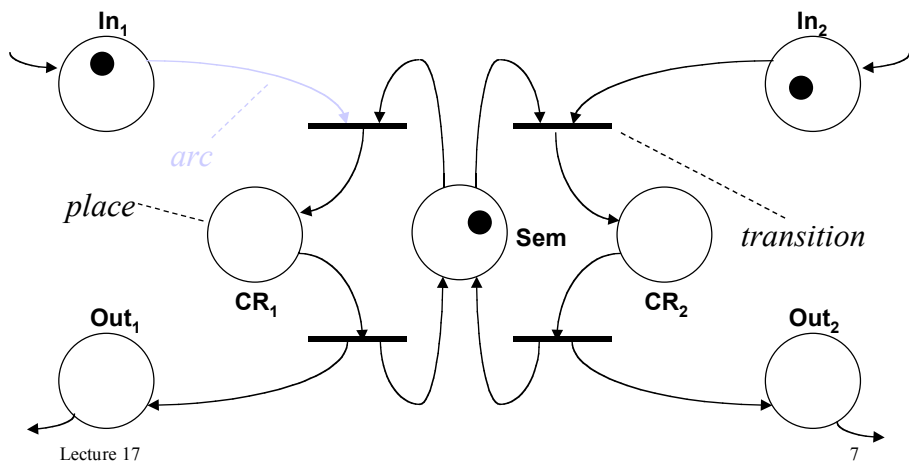
Graphical Representation



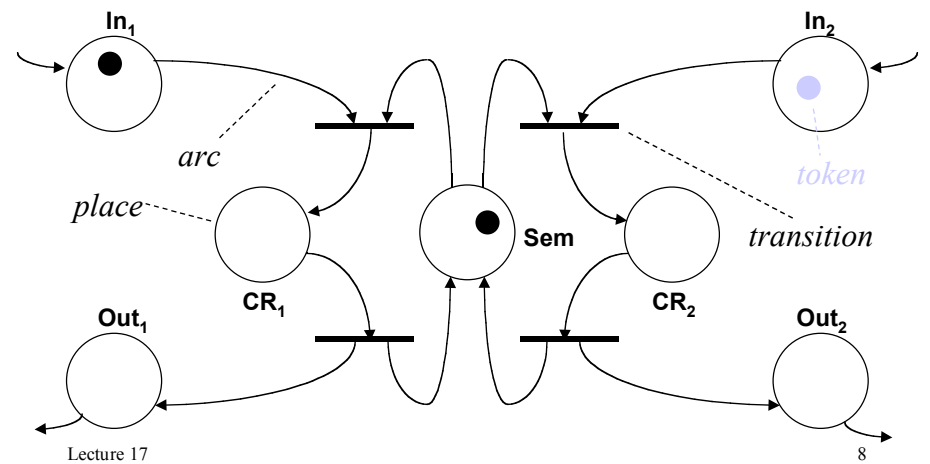
Graphical Representation



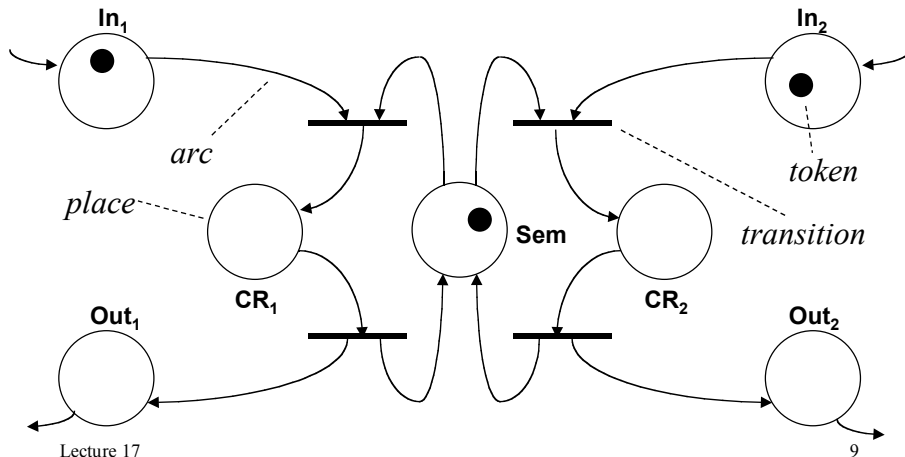
Graphical Representation



Graphical Representation



Graphical Representation



Lecture 17

9

Petri Nets

- Intuitive Meaning
 - A place holds *tokens*
 - A transition represents *activity*
 - An arc connects a place and a transition
 - A marking is an arrangement of tokens in places, representing *state*
 - An initial marking represents an initial state

Lecture 17

10

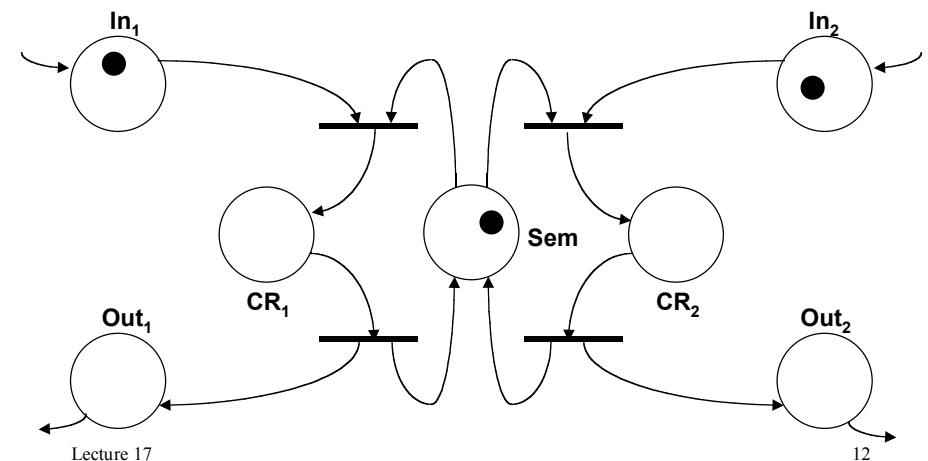
Execution Model

- Input and Output Places
 - Place P is an *input place* for transition T if there is an arc from P to T
 - Place P is an *output place* for transition T if there is an arc from T to P
- Enabled Transition
 - A transition is *enabled* if there is at least one token at each of its input places

Lecture 17

11

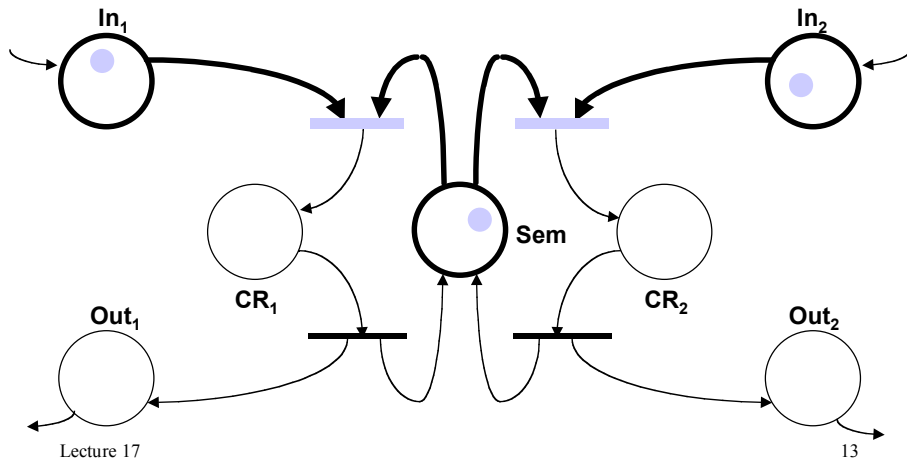
Petri Net Semaphore



Lecture 17

12

Enabled Transitions



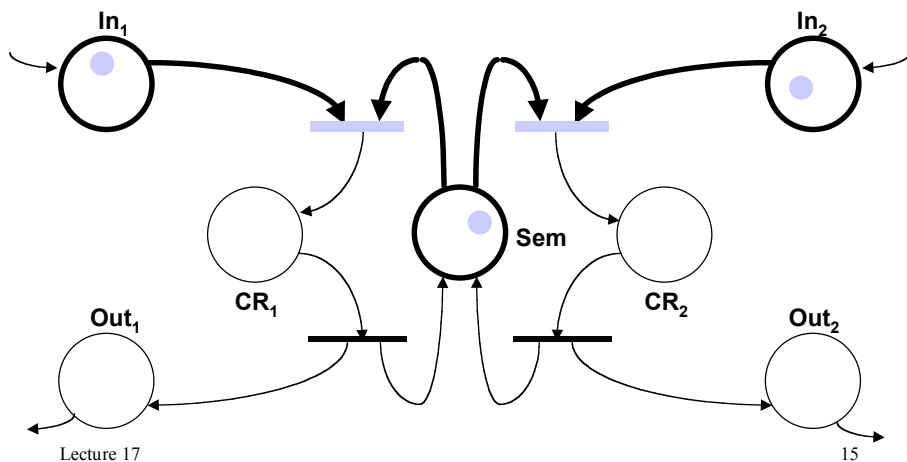
Execution Model

- Firing a Transition
 - An enabled transition is nondeterministically selected and *fired* by removing one token from each of its input places and depositing one token at each of its output places
- Firing Sequence
 - A *firing sequence* is a sequence $\langle t_0, t_1, \dots, t_n \rangle$ such that t_0 is enabled and fired in M_0 , t_1 is enabled and fired in M_1 , etc.

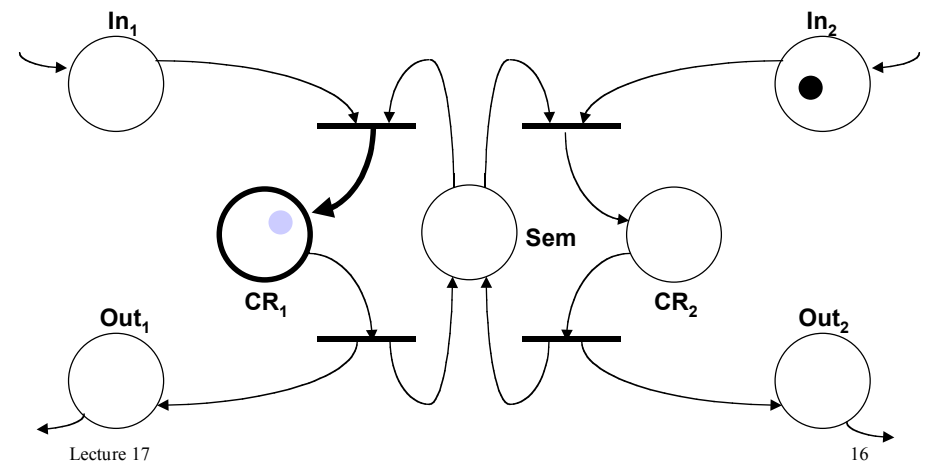
Lecture 17

14

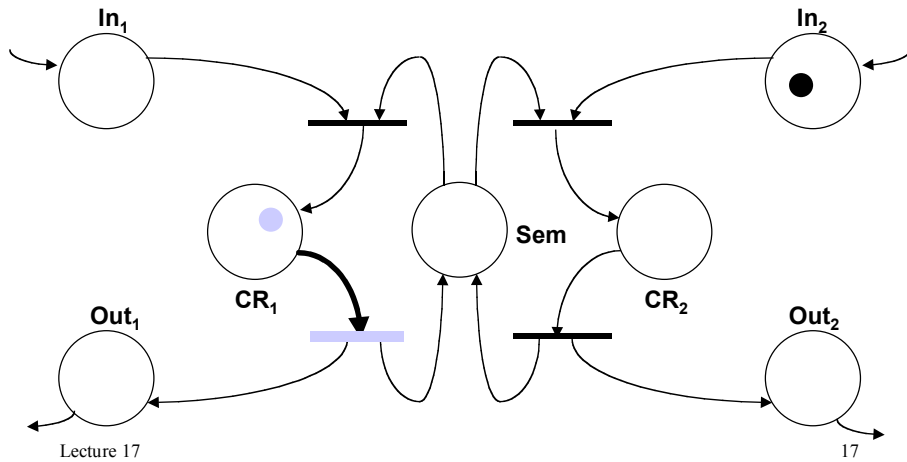
Enabled Transitions



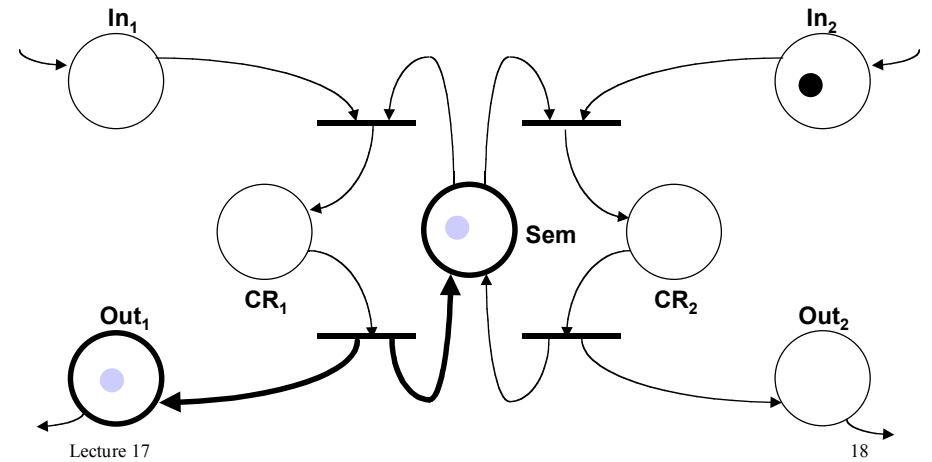
After Firing



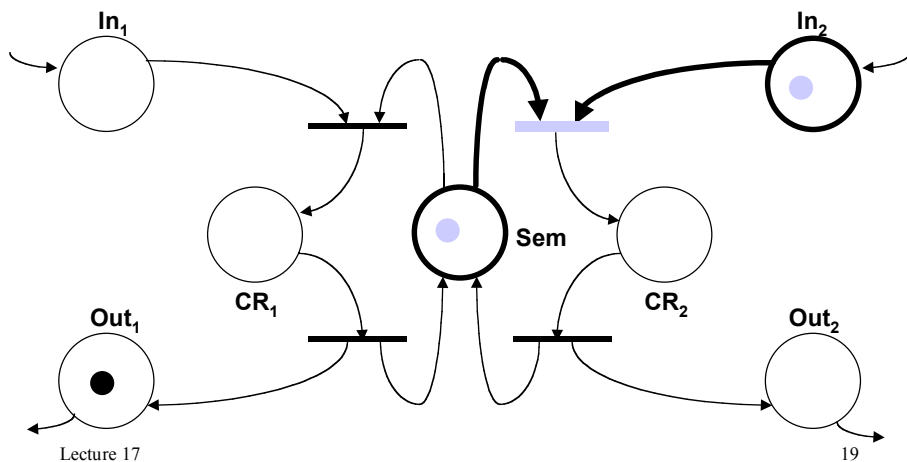
Enabled Transition



After Firing



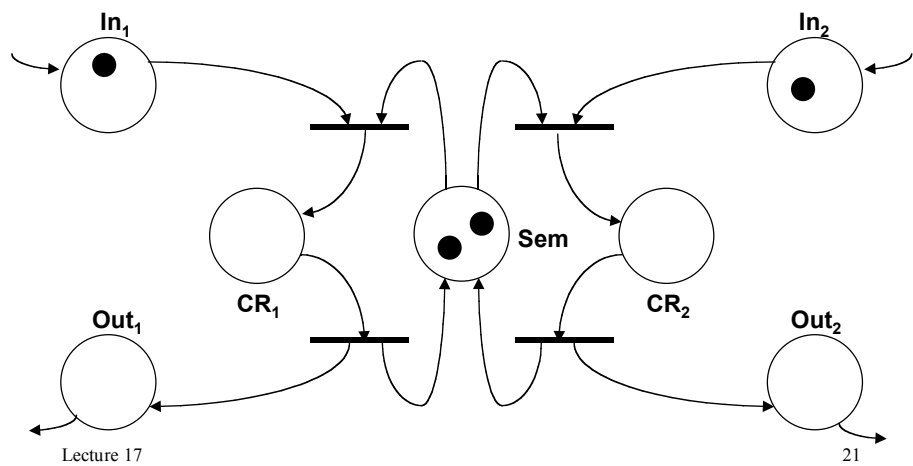
Enabled Transition



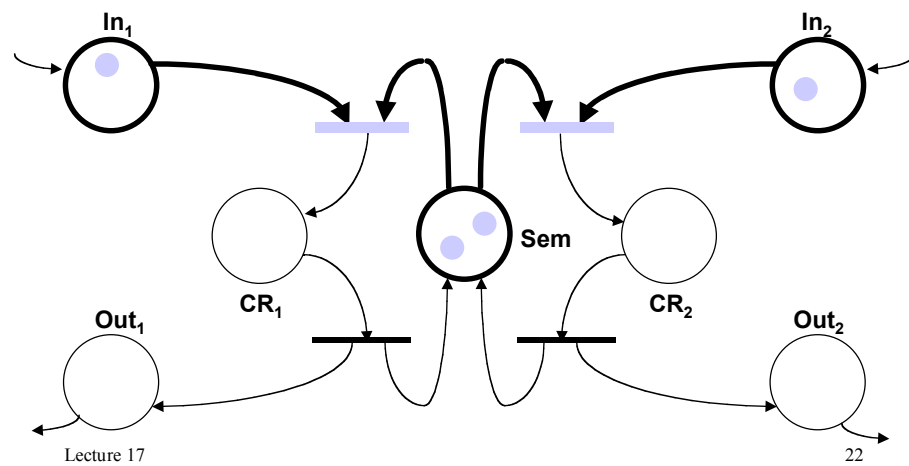
Breaking the Semaphore

- Lets look at the semaphore example again and see how a change to the initial marking will change the semantics of the Petri Net
 - In particular, we will break the semantics of the semaphore by adding *one* token

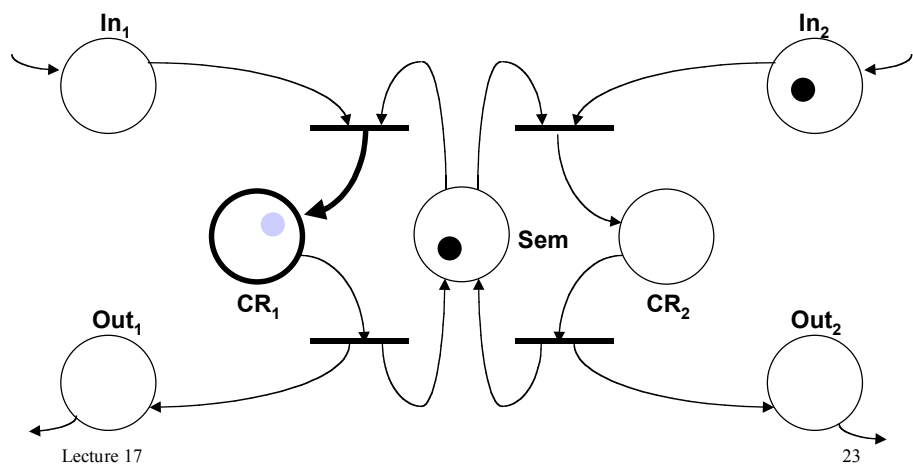
Petri Net Semaphore



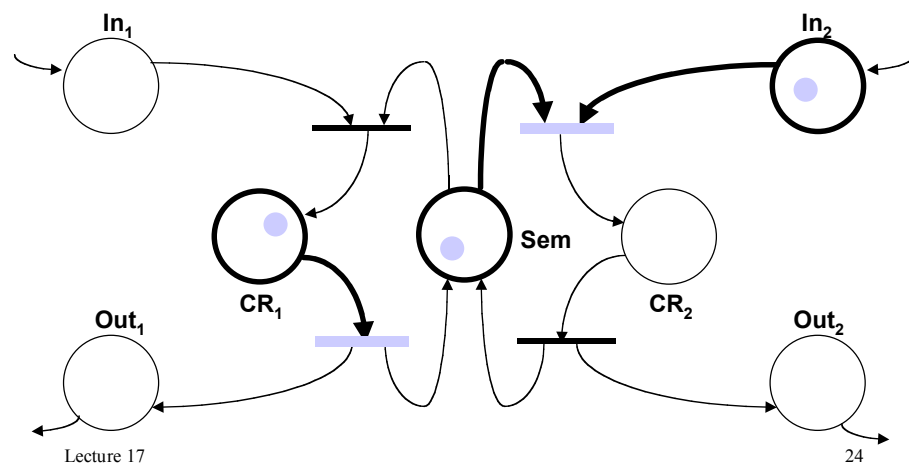
Enabled Transitions



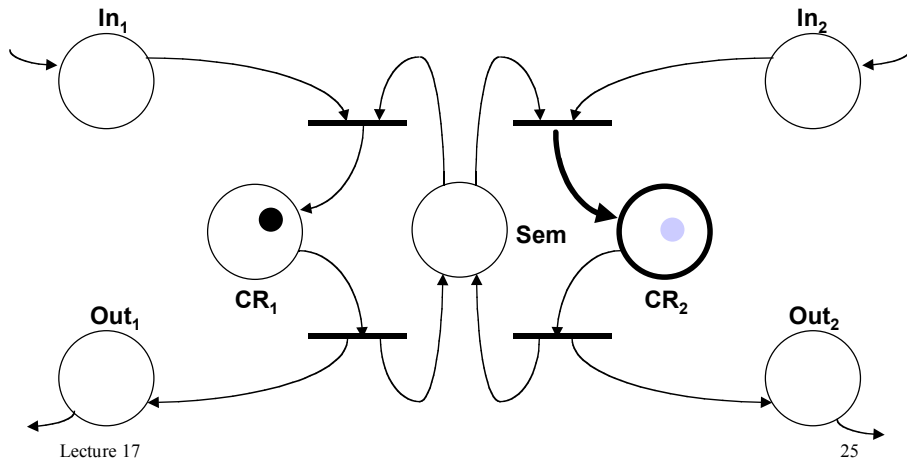
After Firing



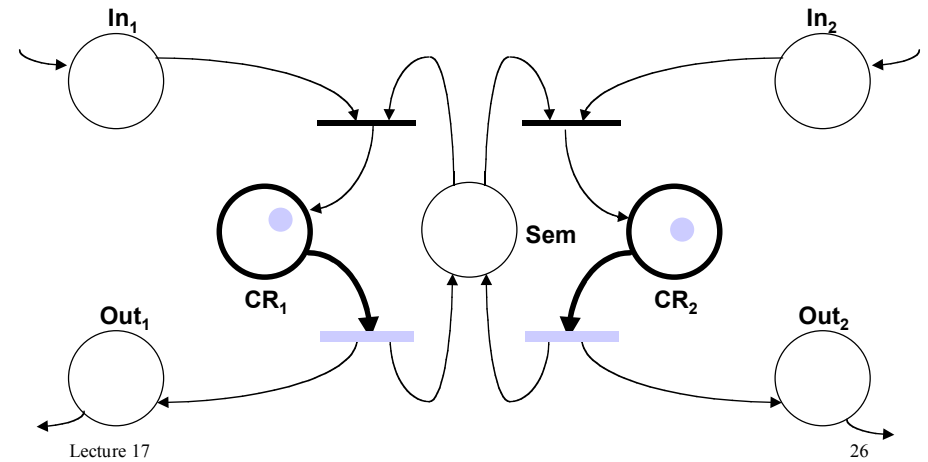
Enabled Transitions



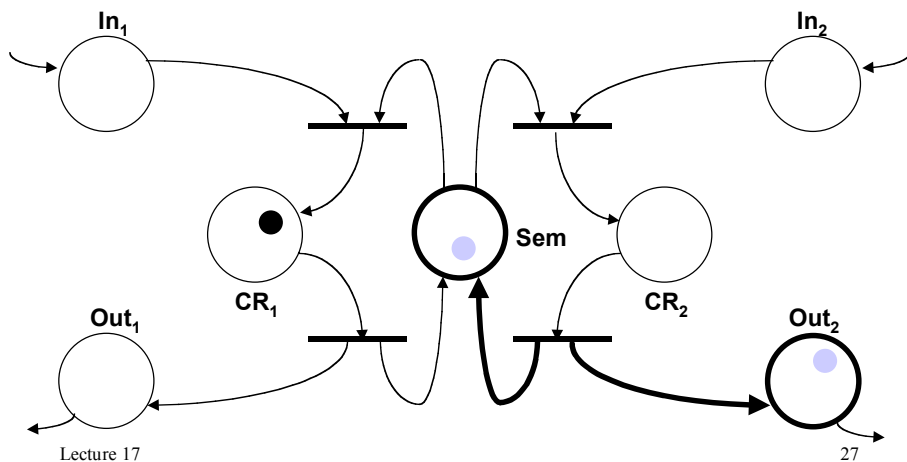
After Firing



Enable Transitions



After Firing



Filling Station Example

- Lets model the following situation
 - Fuel Pumps
 - Spaces next to Pumps
 - A cashier that takes payment
- Questions
 - What is the concurrency that we want modeled?
 - How do we handle the parameterization of the Petri net? (e.g. lets say I want to add a pump)