ECE/CS 541
Computer System Analysis:
Stochastic Activity Networks

Mohammad A. Noureddine
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Fall 2018

Announcements and Reminders

- Project presentations on December 15
  - Will try to start at 5:00 pm to finish early

- Homework 4 is out and due next week

- Submit papers on the 17^{th} via EasyChair
  - Will send the link soon
  - You will get 3 *anonymous* reviews
    - I wonder who the reviewers are!

Outline for the next 2 Weeks

- **Today**
  - Stochastic PetriNets
    - "Syntax" and semantics
    - Example
  - Stochastic Activity Networks Intro
- **Thursday**
  - SAN case studies
  - Intro to output analysis
- **Tuesday**
  - Output analysis wrap up
- **Thursday**
  - TBA (More output analysis or Introduction to Game Theory)

Last Day of Classes

- **Course wrap-up**
  - Or, what have been doing for the past 3 months?

- **ICES forms!!!**
  - Please show and fill up the ICES forms
  - Get 1 point of the participation credits

Learning Objectives

- Or what is this course about?

- At the start of the semester, you should have
  - Basic programming skills (C++, Python, etc.)
  - Basic understanding of probability theory (ECE313 or equivalent)

- At the end of the semester, you should be able to
  - Understand different system modeling approaches
    - Combinatorial methods, state-space methods, etc.
  - Understand different model analysis methods
    - Analytic/numeric methods, simulation
  - Understand the basics of discrete event simulation
  - **Design simulation experiments and analyze their results**
  - Gain hands-on experience with different modeling and analysis tools

Project

Today's Lecture

- Stochastic Petri nets
  - Definitions and semantics
  - Reader/Writers example

- Stochastic Activity Networks
  - Definitions and semantics
  - Dependent behavior, well-specified, general distributions

- Guest "chat" with Professor William H. Sanders
  - Aka my advisor

Introduction

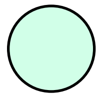
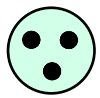
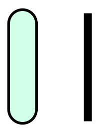
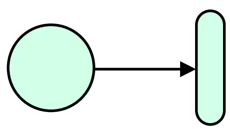
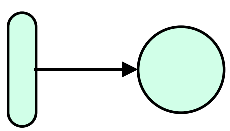
- Developing Markov models by hand can be quite tedious
  - Remember that car wash problem!
- Some small models can be very hard to solve
  - Need a lot of assumptions
    - Independence, exponential distributions, etc.

- Stochastic Activity Networks (SANs) to the rescue
  - Convenient, high-level, graphical, language for describing the behavior of systems
  - Can be solved both analytically or by simulation
  - Allow for complex and interleaved reward structures to obtain meaningful results and insights

- We'll start by studying a subset of SANs: Stochastic Petri Nets (SPNs)

Stochastic Petri Net Overview

One of the simplest high-level modeling formalisms is called *stochastic Petri nets*. A stochastic Petri net is composed of the following components:

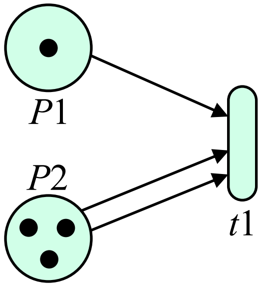
- Places:  which contain tokens, and are like variables
- tokens:  which are the "value" or "state" of a place
- transitions:  (timed, untimed) change the #tokens in places
- input arcs:  which connect places to transitions
- output arcs:  which connect transitions to places

Semantics: Firing Rules for SPNs

A stochastic Petri net (SPN) executes according to the following rules:

- A transition is said to be *enabled* if for each place connected by input arcs, the number of tokens in the place is ≥ the number of input arcs connecting the place and the transition.

Example:

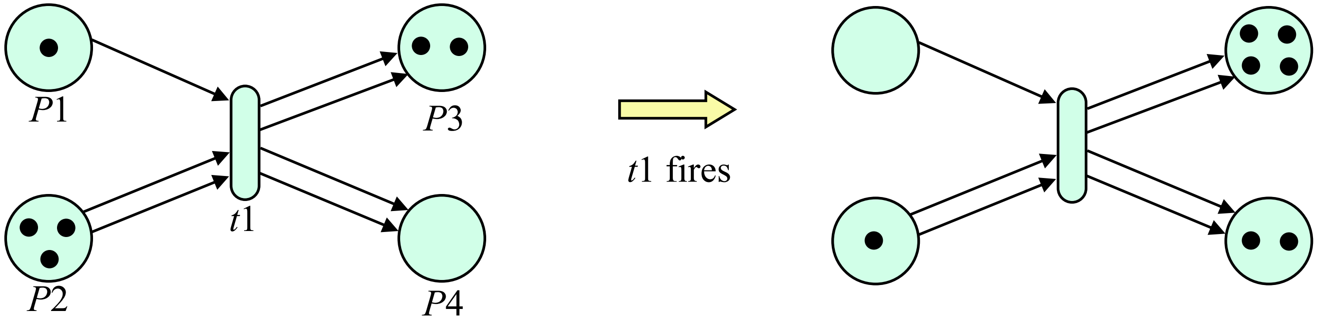


Transition $t1$ is enabled.

Semantics: Firing Rules

- A transition may *fire* if it is enabled.
- If a transition fires, for each input arc, a token is removed from the corresponding place, and for each output arc, a token is added to the corresponding place.

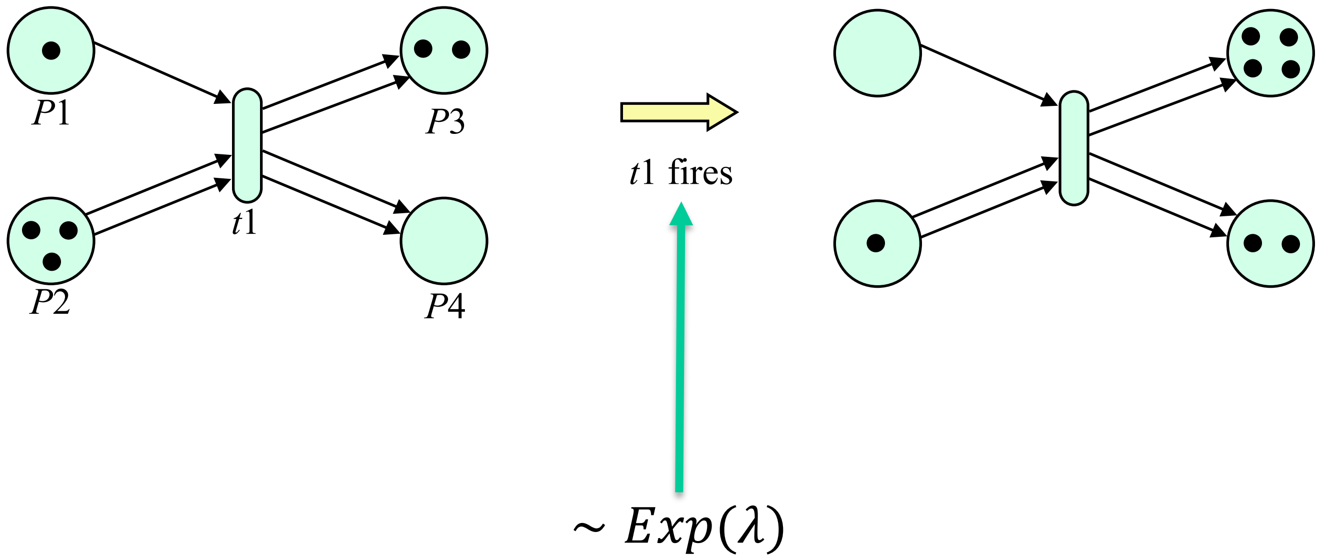
Example:



Note: tokens are not necessarily conserved when a transition fires.

Stochastic Behavior

- Where does stochastic come to play?
  - Assign an exponentially distributed time to all transitions.
  - Time represents the "delay" between enabling and firing of a timed transition.
  - Transitions "execute" in parallel with independent delay distributions.



Stochastic Behavior

- Since the minimum of multiple independent exponentials is itself exponential, time between transition firings is?

- If a transition *t* becomes enabled, and before *t* fires,
  - some other transition fires and changes the state of the SPN such that *t* is no longer enabled,
  - then *t* *aborts*, that is, *t* will not fire.

- Enabled immediate transitions are transient, state changes non-deterministically

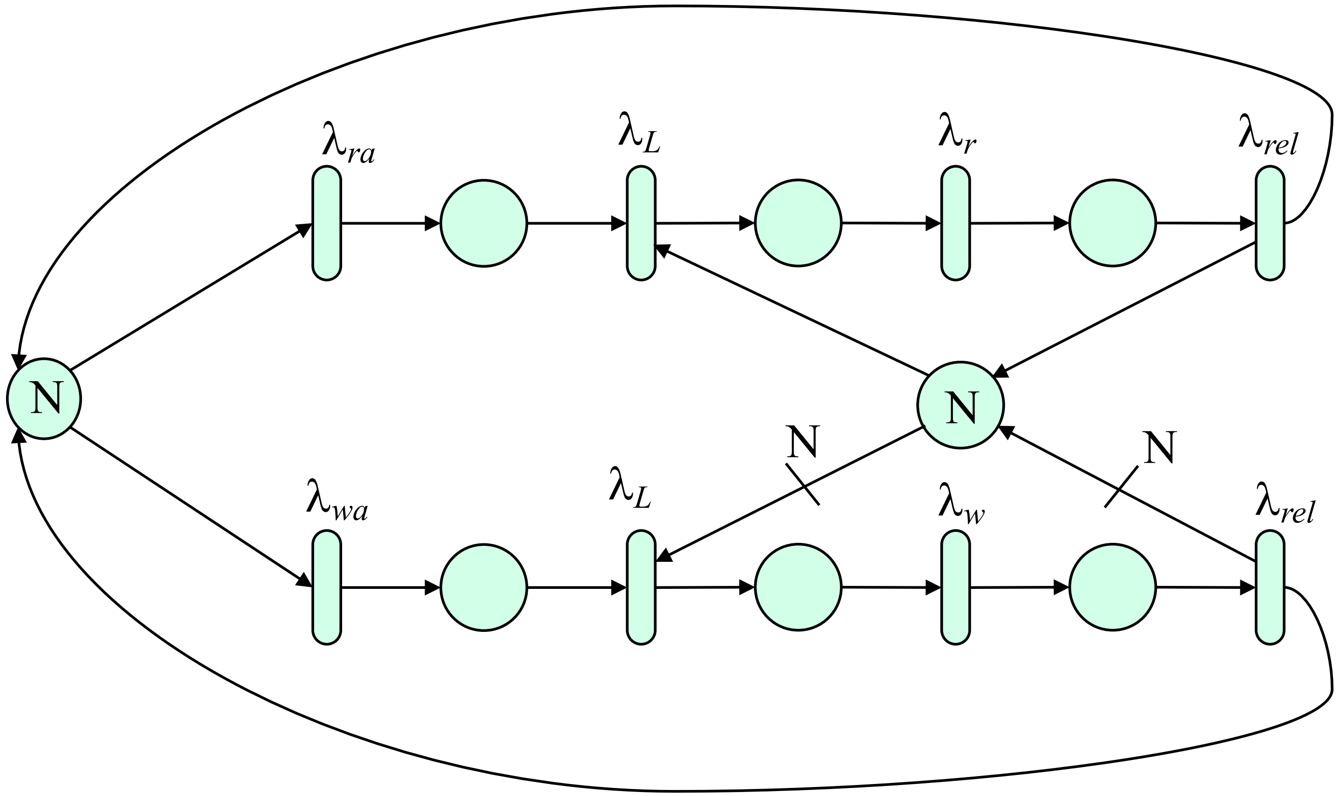
- **Why dos aborting transition work?**
- By memorylessness,
  - one can say that transitions that remain enabled continue or restart, as is convenient, without changing the behavior of the network
    - Recall midterm, problem 3!

SPN Example: Readers/Writers Problem

- There are at most N requests in the system at a time.
- Read requests arrive at rate λ_{ra}, and write requests at rate λ_{wa}.
- Any number of readers may read from a file at a time, but only one writer may write at a time.
- A reader and writer may not access the file at the same time.
- Locks are obtained with rate λ_L (for both read and write locks);
- Reads and writes are performed at rates λ_r and λ_w respectively.
- Locks are released at rate λ_{rel}.

Note:

SPN Representation of Reader/Writers Problem



Notes on SPNs

- SPNs are much easier to read, write, modify, and debug than Markov chains.
- SPN to Markov chain conversion can be automated to afford numerical solutions to Markov chains.
- Most SPN formalisms include a special type of arc called an *inhibitor arc*,
  - inhibit a transition if the connected place has "too many" tokens
- Some also include the identity (do nothing) function.
- Limited in their expressive power: may only perform +, -, >, and test-for-zero operations.
- These very limited operations make it very difficult to model complex interactions.
- Simplicity allows for certain analysis, e.g., a network protocol modeled by an SPN may detect deadlock (if inhibitor arcs are not used).
- More general and flexible formalisms are needed to represent real systems.