Stochastic Petri Nets

Performance and reliability modeling

István Majzik, Kristóf Marussy
BME Department of Measurement and Information Systems

Overview

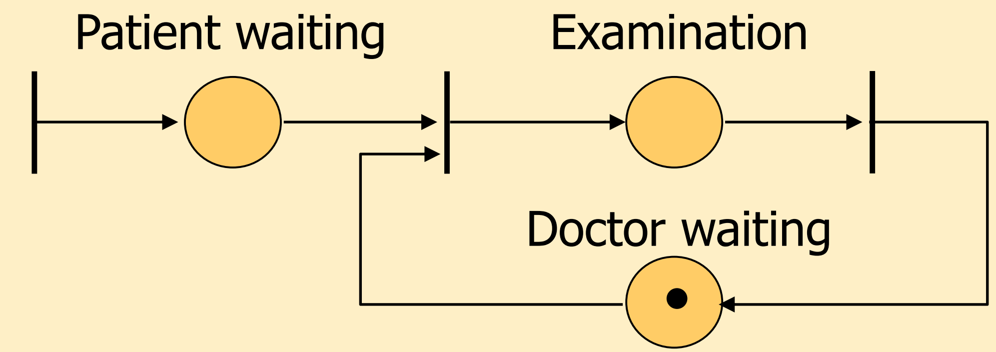
- Motivation
- Stochastic processes and models
  - Continuous-time Markov chains
- Stochastic Petri nets
  - SPN, GSPN, DSPN, TPN
  - Timing semantics
- Summary

Motivation

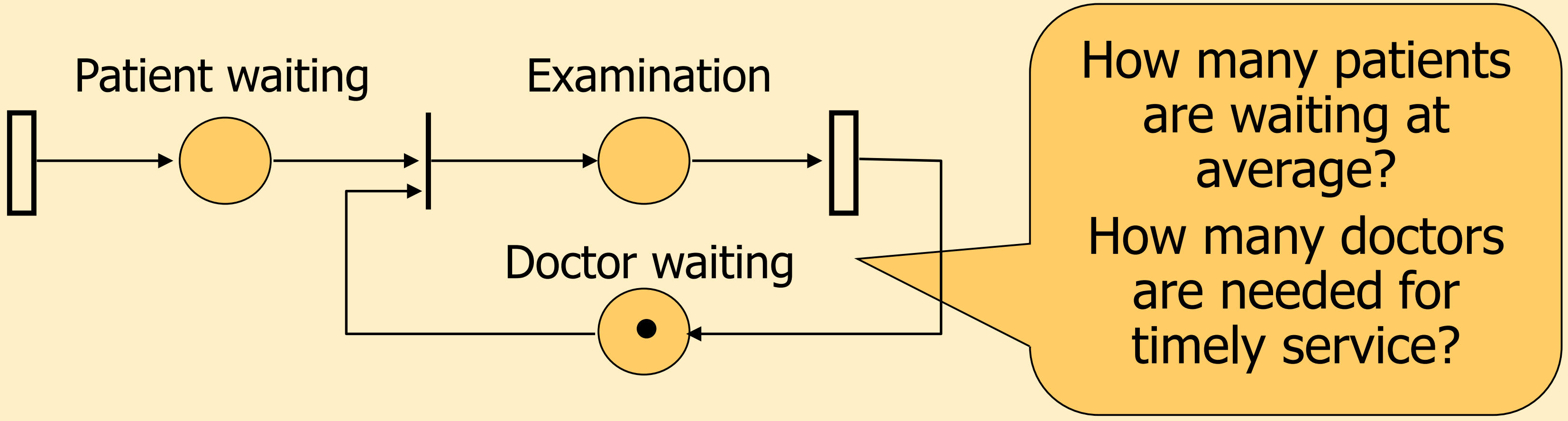
- The course so far:
  Modeling functional, logical behavior
  - Safety and liveness requirements
  - Reachability of a given state or transition
- Extension: extra-functional, quantitative modeling
  - Performance requirements
  - Reliability (dependability) requirements
- Characteristics of extra-functional requirements
  - Timing (e.g. deadlines, response and processing times)
  - Probabilities (e.g. component failure, packet loss)
- Modeling IT systems
  - Discrete state space
  - Continuous time

Simple example

- Ordinary Petri-net model – Doctor's office:



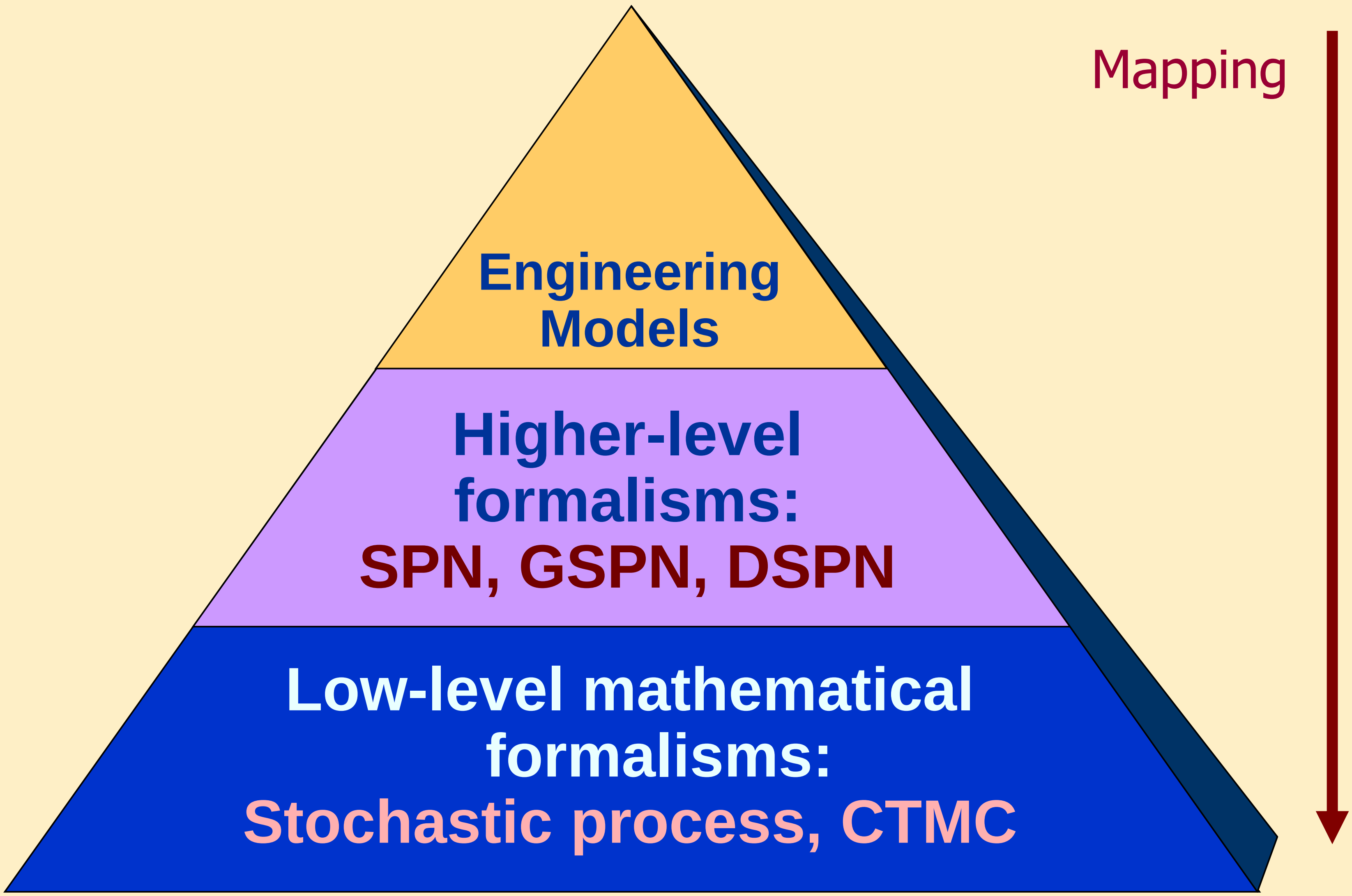
- Model extended with timing information:



Advantages and challenges

- Advantages of modeling: Analysis at design time (before costly implementation efforts!)
  - Confirm of design decision
  - Compare alternatives
  - “Tune” parameters
- Challenges in modeling: Realism
  - Parameters: Timing and probability informations
    - Are they available?
    - If they are approximations, how are they validated?
  - Complexity of models
    - How much abstraction can be used?

Models and formalisms

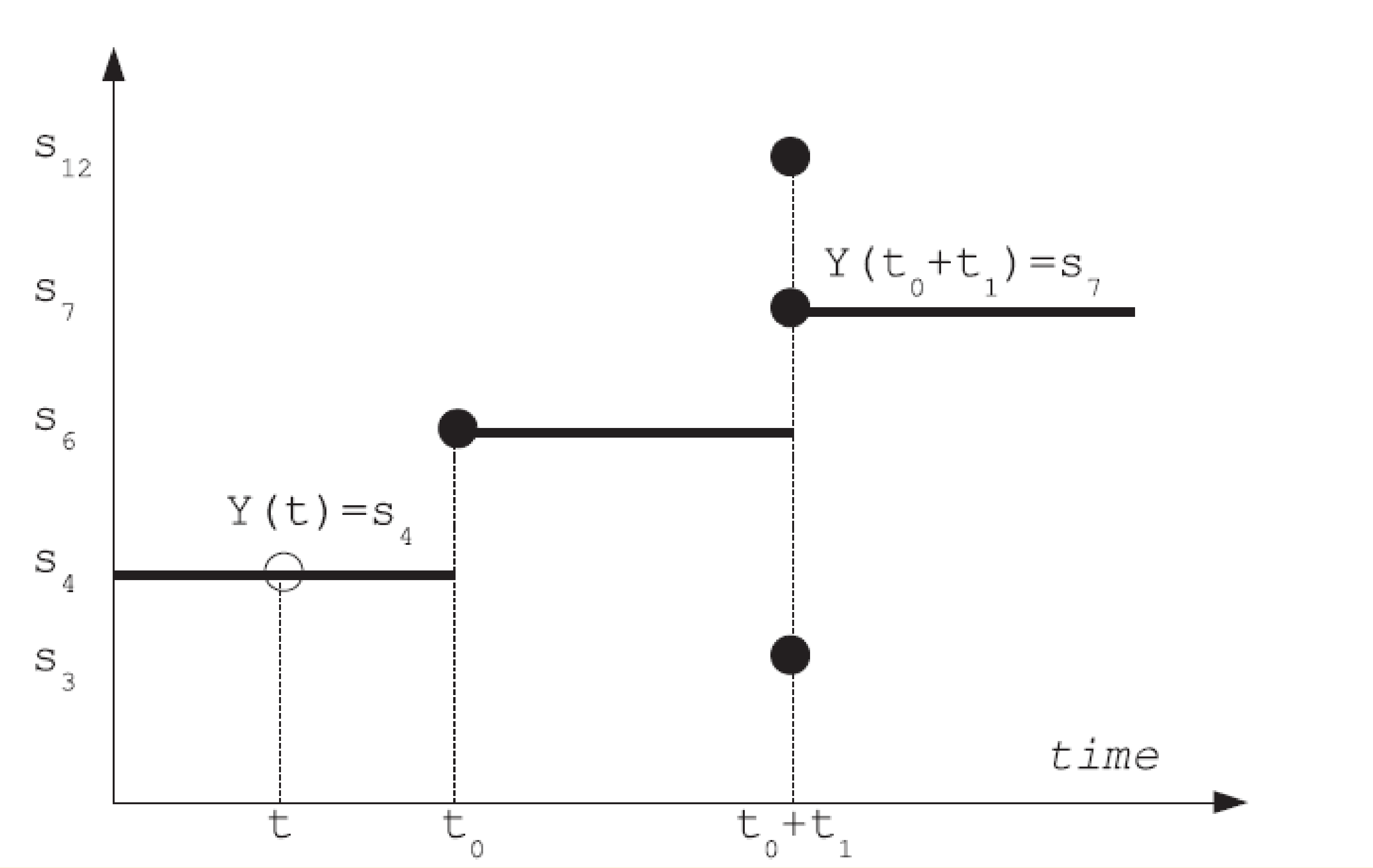


Low-level formalisms: Stochastic processes, continuous time Markov chains (CTMC)

Stochastic processes

- Random variable:
  Probabilistic selection of a value in a random field
  - A realization of random variable X is a value x
  - P.d.f and c.d.f for real-valued random variables
- Stochastic process:
  - Informally: modeling with random variables parameterized with time
  - Discrete case: States (state probabilities) changing w.r.t. time
    - Set of random variables indexed with parameter t (time): X(t)
    - The domain of each random variables is the same
- Behavior of the process:
  - Random field of trajectories over the discrete states
  - The set of all possible trajectories describes the process fully

Trajectory



- Holding times of states: t_0, t_1, …

Markov processes

- Stochastic process with the Markov property:

  For all $t > t_n > t_{n-1} > \ldots > t_0$, the process $X(t)$ satisfies:

  $P\{X(t)=x \mid X(t_n)=x_n, X(t_{n-1})=x_{n-1}, \ldots, X(t_0)=x_0\} = P\{X(t)=x \mid X(t_n)=x_n\}$
- "Memoryless" property
  - The future state (at time t) only depends on the current state (at time t_n), not on any earlier state
- Markov processes with discrete state space: Markov chains
  - The trajectories are described by the holding times of states
  - Holding time τ is exponentially distributed: $P\{\tau \leq t\}=1-e^{-\lambda t}$
    - This is the only continuous probability distribution with the Markov property
    - At any moment of time the remaining holding time in the active state is independent from the time already spent in the state

Continuous time Markov chains (CTMC)

- Continuous time, but discrete state space
- Some definitions:
  - States: $s_0, s_1, \ldots, s_n$
  - Transition probability matrix: $Q_{ij}(t_{n-1},t_n) = P\{S(t_n)=s_j \mid S(t_{n-1})=s_i\}$
  - (Time-)homogenous process: $Q_{ij}(t,t+\Delta t) = Q_{ij}(\Delta t)$
    - Transition probabilities do not change over time
  - Transition rate (intensity):

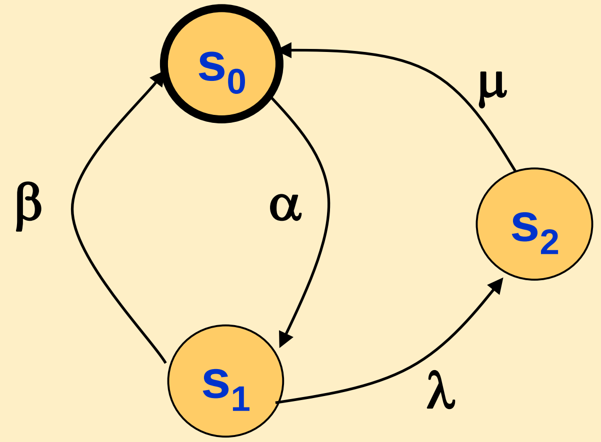
$$R_{ij}(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} Q_{ij}(\Delta t)$$

  - Rate of leaving state s:

$$E(s) = \sum_{s' \in S, s \neq s'} R_{s,s'}$$

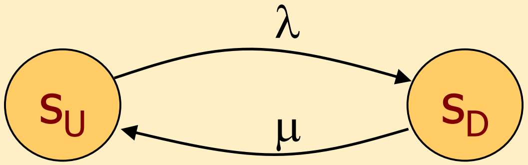
Example

- Graphical CTMC notation:
  - States (with initial state / initial state probabilities)
  - Transition rates between pairs of states
    (zero rates are omitted for brevity)



Applying CTMCs

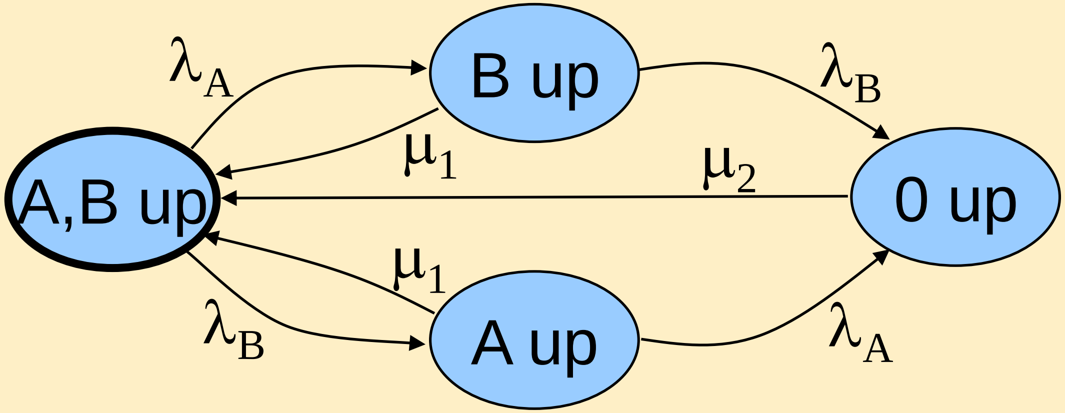
- Reliability modeling:
  - Atomic component: Error-free state s_U and error state s_D
  - Practical observations for electronic components:
    - Under typical load, the holding time of the error-free state is exponentially distributed
    - Parameter: failure rate, λ
    - For simplicity, we also consider an exponential distribution for the repair time: repair rate, μ
  - Hence, we obtain the CTMC:



- Performance modeling
  - Queueing theory for servers
    - M/M/1 queue:
      Markovian (exponentially distributed) arrival and service times
    - State space is a CTMC

Example: Reliability modeling

- System comprised of two servers (A and B):
  - Any server may fail
  - Repairs: single server of both servers at a time
- System states: Working servers
- State transitions (exponentially distributed delays):
  - Failure of server A: λ_A failure rate
  - Failure of server B: λ_B failure rate
  - Repair a single server: μ_1 repair rate
  - Repair both servers: μ_2 repair rate



CTMC notations

- CTMC=(S, **R**)

  S set of states

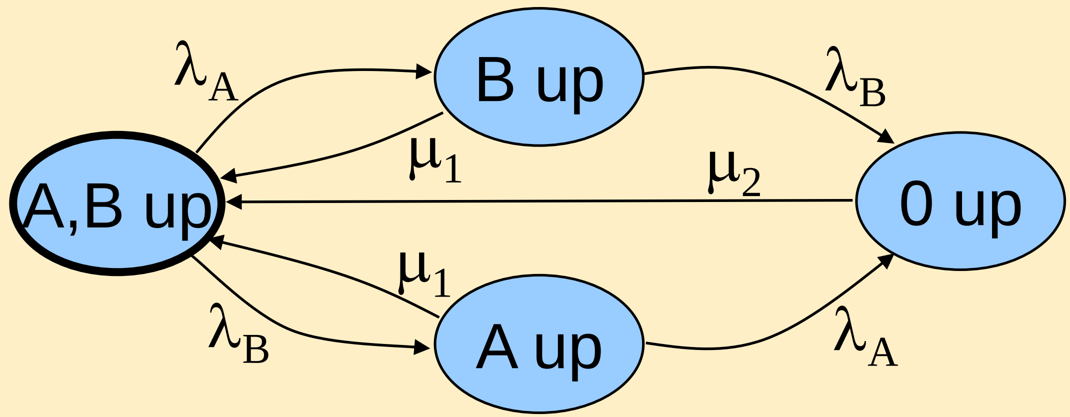
  R: $S \times S \rightarrow R_{\geq 0}$ transition rate (intensity matrix)

  - P{transition from s to s' in time ≤ t} = $1-e^{-R(s,s')t}$
  - P{exiting state s in time ≤ t} = $1-e^{-E(s)t}$

  Q = **R**–diag(E) „infinitesimal generator matrix"

- Trajectories:
  - $\sigma = s_0, t_0, s_1, t_1, \ldots$ (at time t_i the state s_i is exited)
  - σ@t state at time t
  - Path(s) set of paths from state s
  - P{s, σ} probability of taking $\sigma \in$ Path(s) from s

Example: Reliability modeling (cont.)



- $\mathbf{R} = \begin{array}{c c} & \begin{array}{cccc} A,B & B & A & 0 \end{array} \\ \begin{array}{c} A,B \\ B \\ A \\ 0 \end{array} & \begin{pmatrix} 0 & \lambda_A & \lambda_B & 0 \\ \mu_1 & 0 & 0 & \lambda_B \\ \mu_1 & 0 & 0 & \lambda_A \\ \mu_2 & 0 & 0 & 0 \end{pmatrix} \end{array}$

- $\mathrm{E} = \begin{array}{cccc} A,B & B & A & 0 \\ (\lambda_A + \lambda_B & \mu_1 + \lambda_B & \mu_1 + \lambda_A & \mu_2) \end{array}$

- $\mathbf{Q} = \begin{array}{c c} & \begin{array}{cccc} A,B & B & A & 0 \end{array} \\ \begin{array}{c} A,B \\ B \\ A \\ 0 \end{array} & \begin{pmatrix} -(\lambda_A + \lambda_B) & \lambda_A & \lambda_B & 0 \\ \mu_1 & -(\mu_1 + \lambda_B) & 0 & \lambda_B \\ \mu_1 & 0 & -(\mu_1 + \lambda_A) & \lambda_A \\ \mu_2 & 0 & 0 & -(\mu_2) \end{pmatrix} \end{array}$

CTMC solution

- Transient probabilities:
  - $\pi(s,s',t) = P\{\sigma \in Path(s) \mid \sigma@t=s'\}$
    probability of getting to s' at time t starting from s
  - $\underline{\pi}(s,t)$ row vector of state probabilities at time t starting from s
  - Transient CTMC solution:

$$\frac{d\,\underline{\pi}(s,t)}{dt} = \underline{\pi}(s,t)\underline{\underline{Q}}$$

- Steady-state probabilities:
  - $\pi(s,s') = \lim_{t\to\infty} \pi(s,s',t)$ steady state probability of s' starting from s
  - $\underline{\pi}(s)$ row vector of steady-state probabilities
  - Steady-state CTMC solution:

$$\underline{\pi}(s)\underline{\underline{Q}} = 0 \quad \text{where} \quad \sum_{s'} \pi(s,s') = 1$$

CTMC in formal methods

- Characteristics of Makov chains
  - Pro: Availability of mathematical methods for solution
  - Con: Large state spaces are difficult to create and handle
    - Too low level: individual states and transitions
    - No support for concurrency and synchronization
    - No hierarchical modeling
- Markov chains in practice
  - Low-level formalism for higher level models
    - Stochastic Petri nets
    - Stochastic process algebras
  - Reachability graph of a stochastic Petri net is a CTMC
    - Stochastic Petri net analysis can be performed by Markov chain solution
  - Analogy: the reachablity graph of a Petri net is a Kripke structure
    - Model checking can be performed on the Kripke structure

Stochastic Petri nets

Definition

- Idea:
  - Time elapses as transitions fire (we model durations of activities, events, or state changes on the transitions)
- A Petri net is called stochastic when
  - Each transition has an associated firing delay
  - The firing delay is random
    (described by a nonnegative real valued random variables)
  - Firing delays of each transition are statistically independent
- Stochastic Petri net variants
  - Stochastic Petri net (SPN)
  - Generalized stochastic Petri net (GSPN)
  - Deterministic and stochastic Petri net (DSPN)

Stochastic Petri nets (SPN)

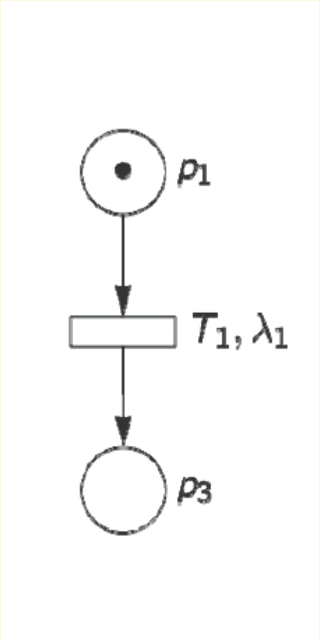
- Extension of ordinary Petri nets
  - Each firing delay is exponentially distributed
- Changes to the firing semantics
  - Enablement: same as in Petri nets
  - Firing rule: A transition fires at time t+d if
    - it became enabled at time t
    - d is the random delay generated from the firing distribution
    - it was always enabled during the time interval [t, t+d)

Notation

- Transitions rates
  - λ_i is the rate parameter of the firing delay d_i of transition T_i (always a positive number)
- Graphical notation:
  - Transitions are empty rectangles
- Transition with rate λ:
  - Firing delay d_i satisfies:

$$P\left\{d_i \le t\right\} = 1 - e^{-\lambda_i t}$$

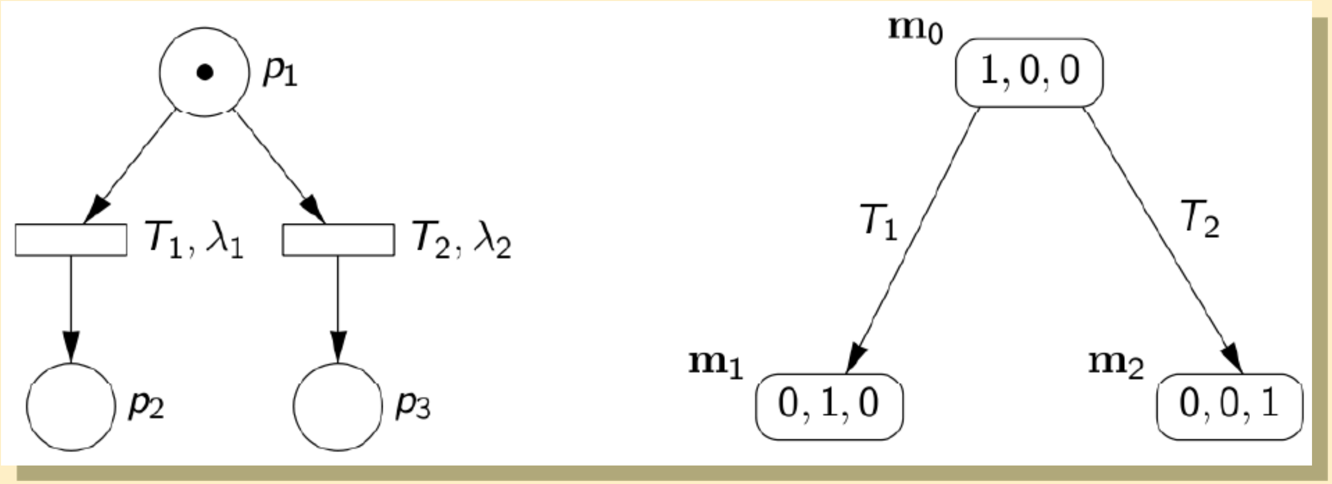
$$P\left\{d_i > t\right\} = e^{-\lambda_i t}$$



Multiple enabled transitions

- The transition with the smallest firing delay is fired
  - The enabled transitions are in a race
  - Probabilistic choice based on the generated firing delays
- What happens to the other enabled transitions after firing?
  - We are in a new marking
  - Do we have to draw a new firing delay from the distribution?
    - In an SPN it does not matter, because firing delays are exponentially distributed and have the Markov property
    - The remaining firing delay has the same exponential distribution regardless the time the transition spent being enabled
    - The remaining firing delay and the time spent being enabled are statistically independent

Transitions in conflict



- Holding time of marking m_0:
  - Minimum of two exponentially distributed firing delays
    - This is also exponentially distributed, and it has a rate $\lambda_1 + \lambda_2$
  - Holding time is exponentially distributed with rate $\lambda_1 + \lambda_2$
  - Expected holding time: $1/(\lambda_1 + \lambda_2)$

Generalization

- If n transitions are enabled with rates $\lambda_1, \lambda_2, \ldots, \lambda_n$ in the marking m, then
  - The holding time of m is exponentially distributed with rate:

$$\lambda_1 + \lambda_2 + \ldots + \lambda_n$$

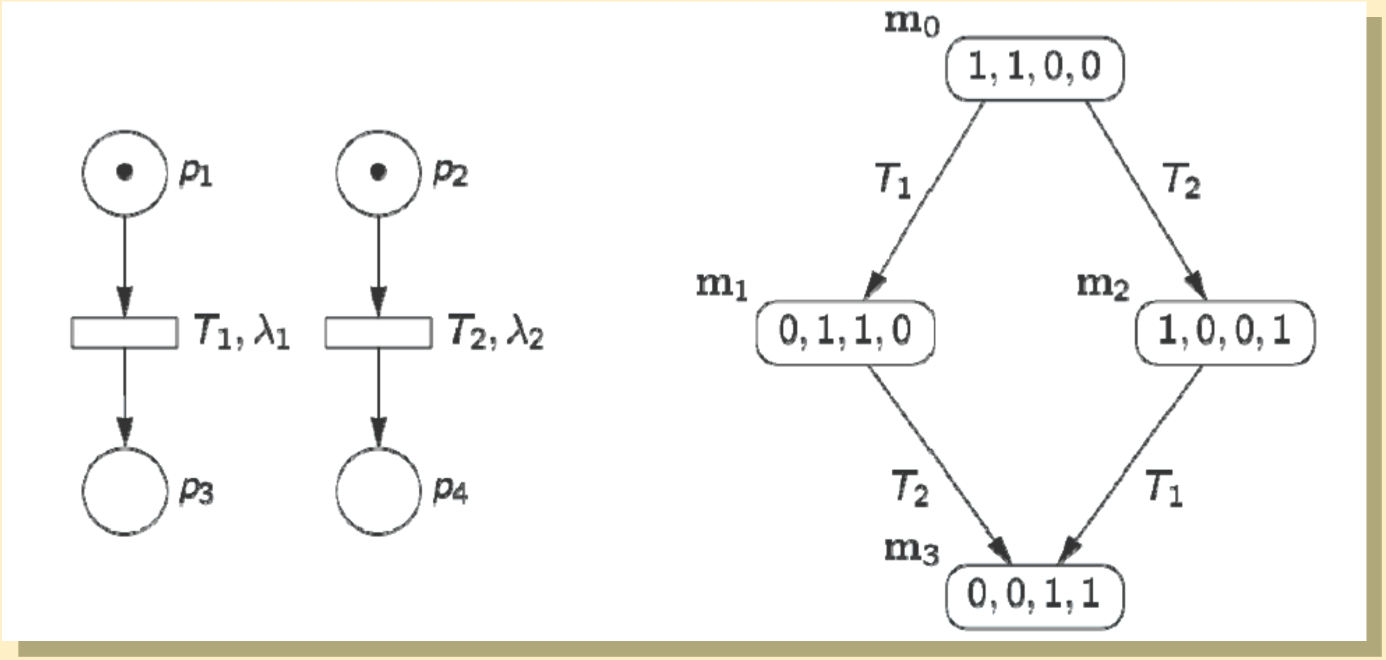
  - Expected holding time of m:

$$\frac{1}{\lambda_1 + \lambda_2 + \ldots + \lambda_n}$$

  - Probability of the transition with rate λ_1 firing first:

$$\frac{\lambda_1}{\lambda_1 + \lambda_2 + \ldots + \lambda_n}$$

Concurrent transitions



- If T_1 fires after some delay $d_1 \geq 0$, what will be the firing delay of T_2 in the new marking?
  - It is still exponentially distributed with rate λ_2 due to the Markov property
- What is the holding time of m_0?
  - Exponentially distributed random variable with rate $\lambda_1 + \lambda_2$

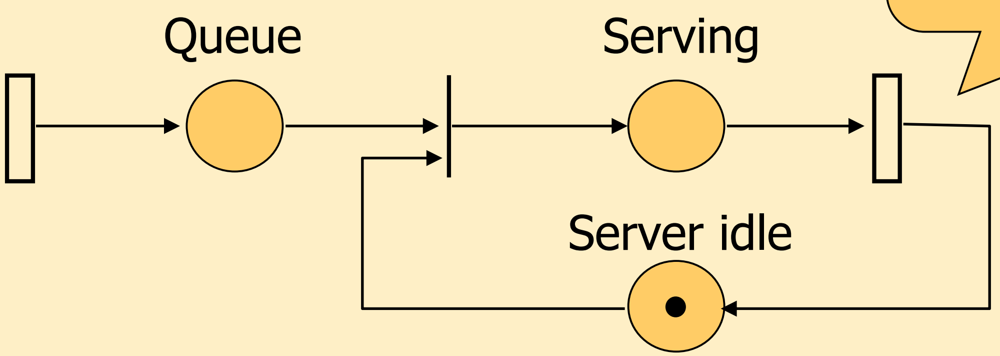
Summary: Properties of SPNs

- Time to a new marking is exponentially distributed
  - Even if transitions are in conflict or are concurrent
- The timed reachability graph is a CTMC
  - Its structure does not depend on the firing rates
  - SPNs are analyzed with CTMC solution methods
- Analysis results
  - Steady state distribution (exists for bounded reversible SPNs):
    - Probabilities of markings (asymptotically)
    - Expected numbers of tokens on places
    - Actual rate of firing for each transitions
  - Transient solution:
    - Probabilities of markings as a function of time elapsed since start

Example: M/M/1 queue

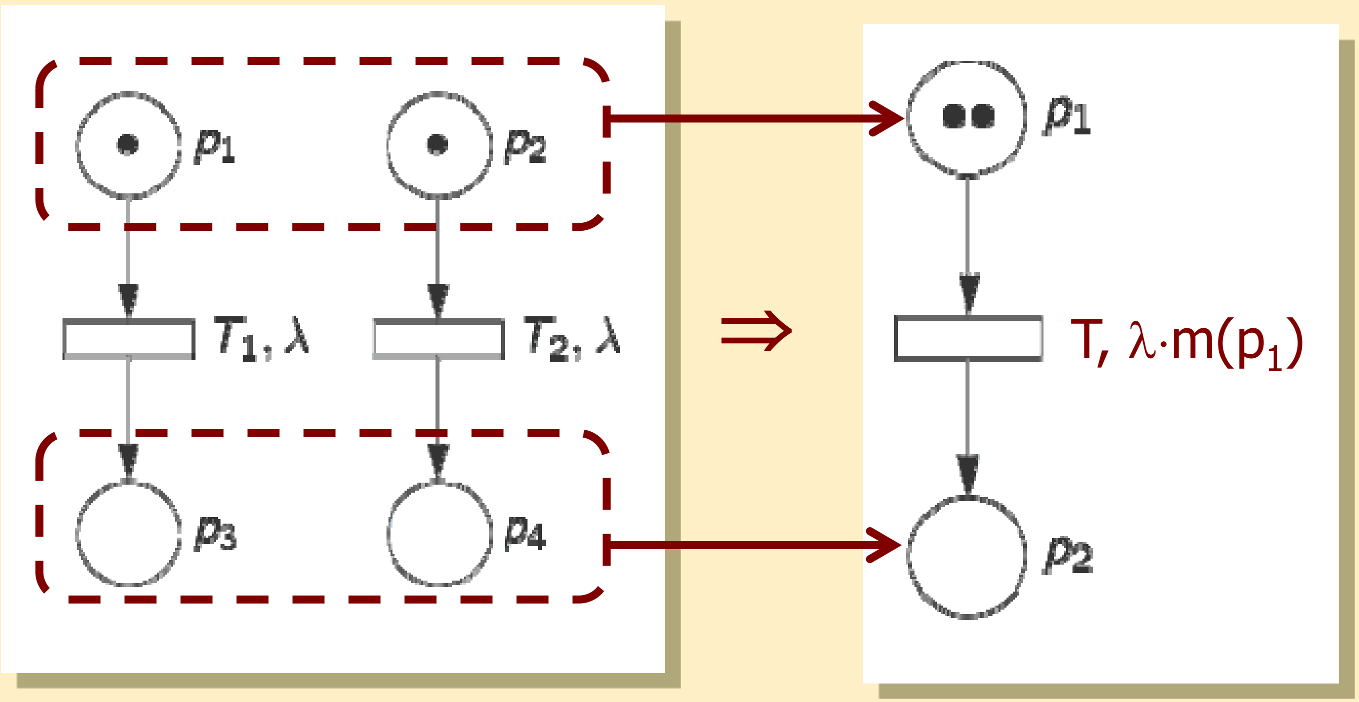
- Single server handling requests from a queue
- Exponentially distributed:
  - Time between subsequent requests
  - Time spent handling a request

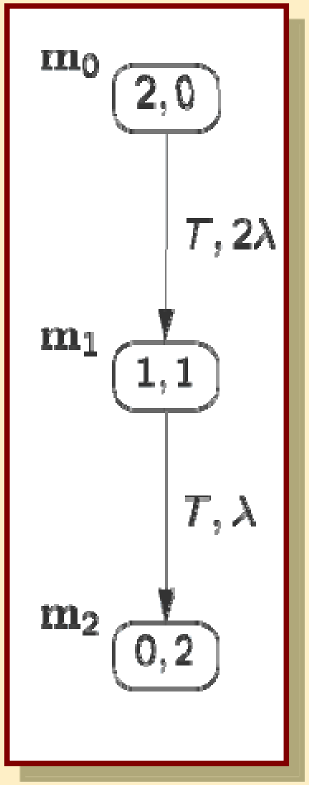
The doctor's office example from the beginning of lecture!



- We can find (as a function of model parameters):
  - Server utilization (probability of "idle" marking)
  - Averange queue length (expected number of tokens)

Simplification of parallel transitions





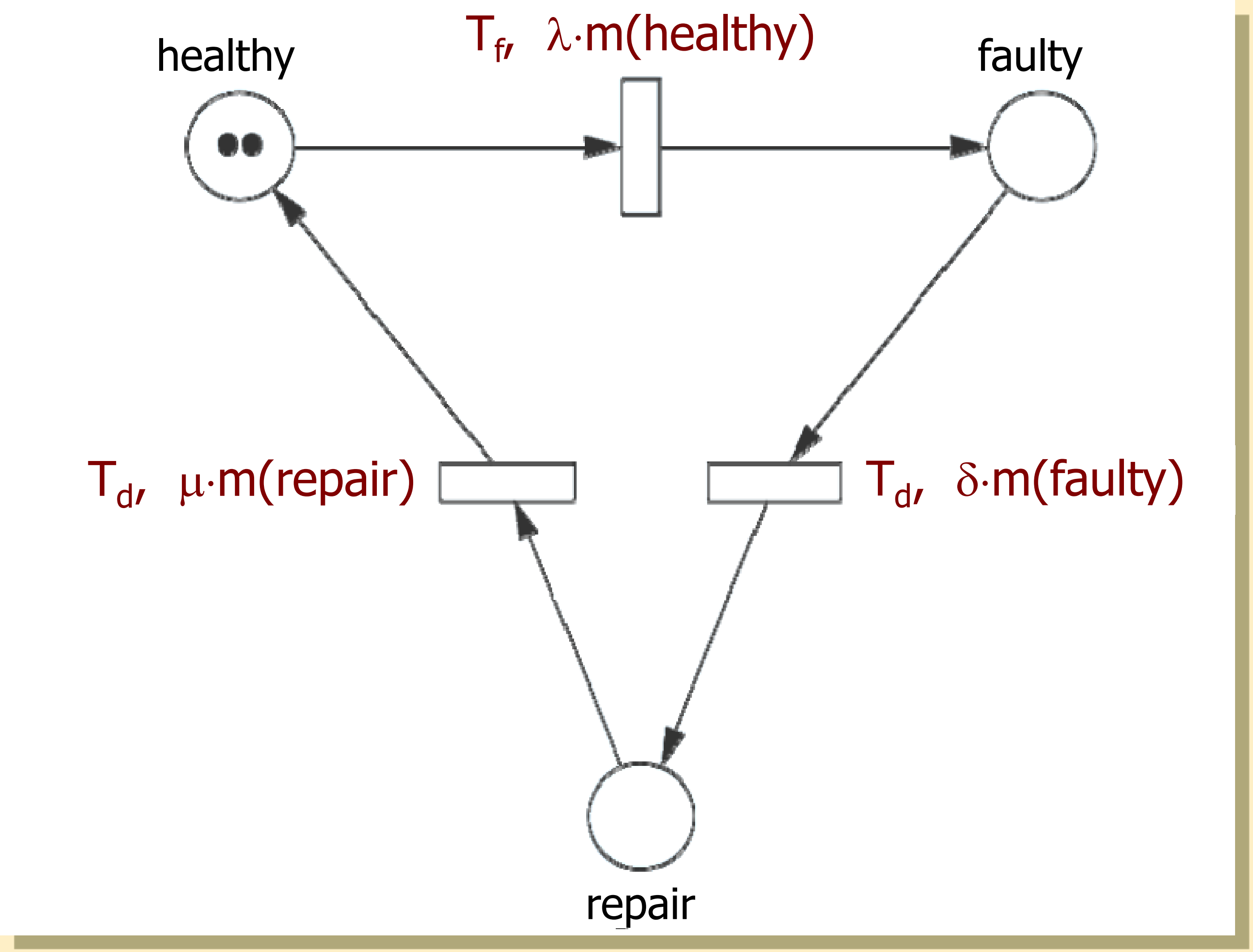
- Marking-dependent transition rates
  - Does not increase descriptive power (we can model the same things)
  - The transition rate may depend on the number of tokens on a place connected by an input or inhibitor arc

Example: Reliability model of a redundant system

- Two identical servers
- Error rate of a server: λ
  - λ is the parameter of the exponential distribution of the time to failure
  - Servers fail independently from each other
- Error detection time:
  exponentially distributed with rate δ
  - We may detect multiple failures concurrently
- Repair time:
  exponentially distributed with rate μ
  - Multiple concurrent repairs are allowed (more than one technician is available)

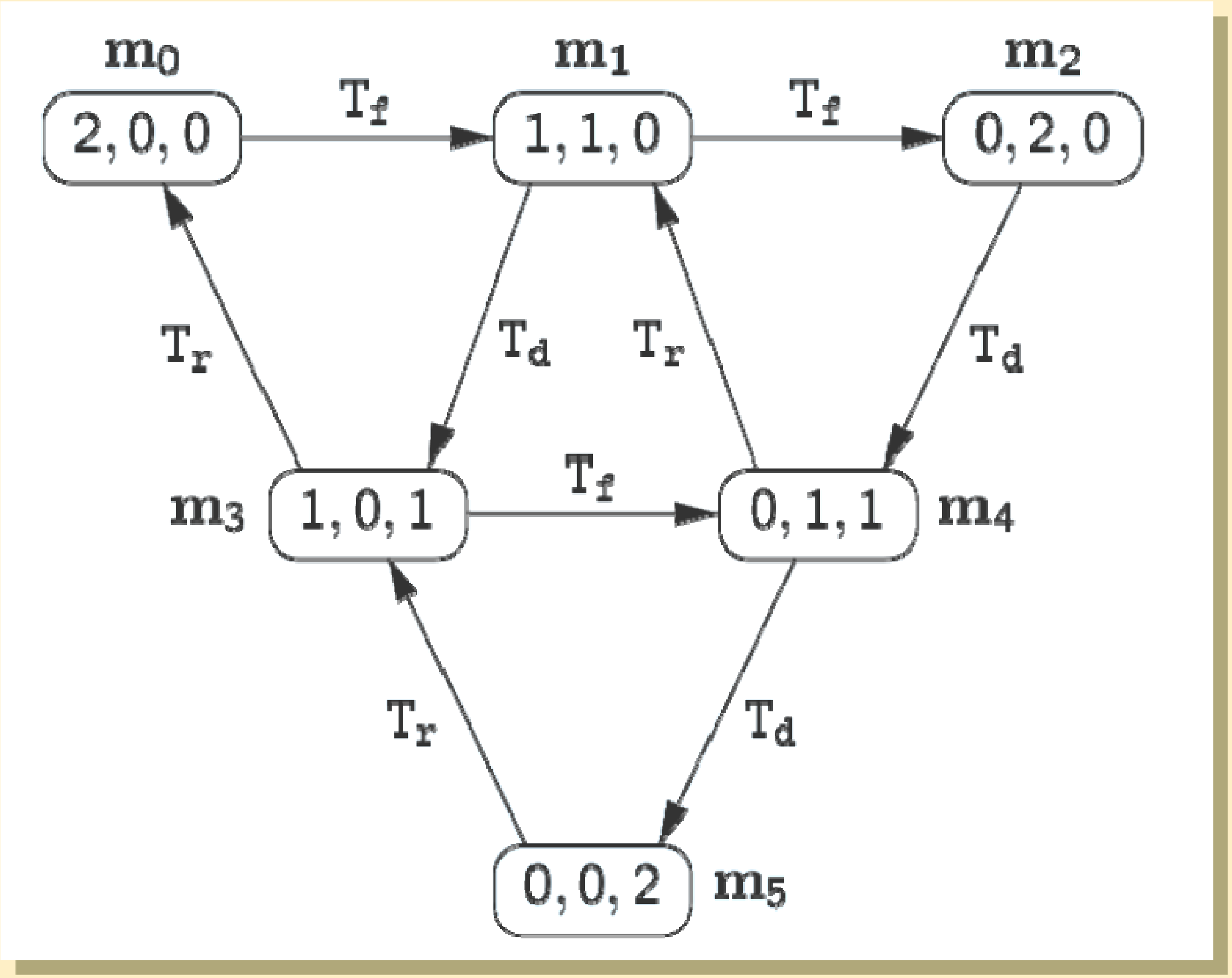
Example: Reliability model of a redundant system

- SPN model:



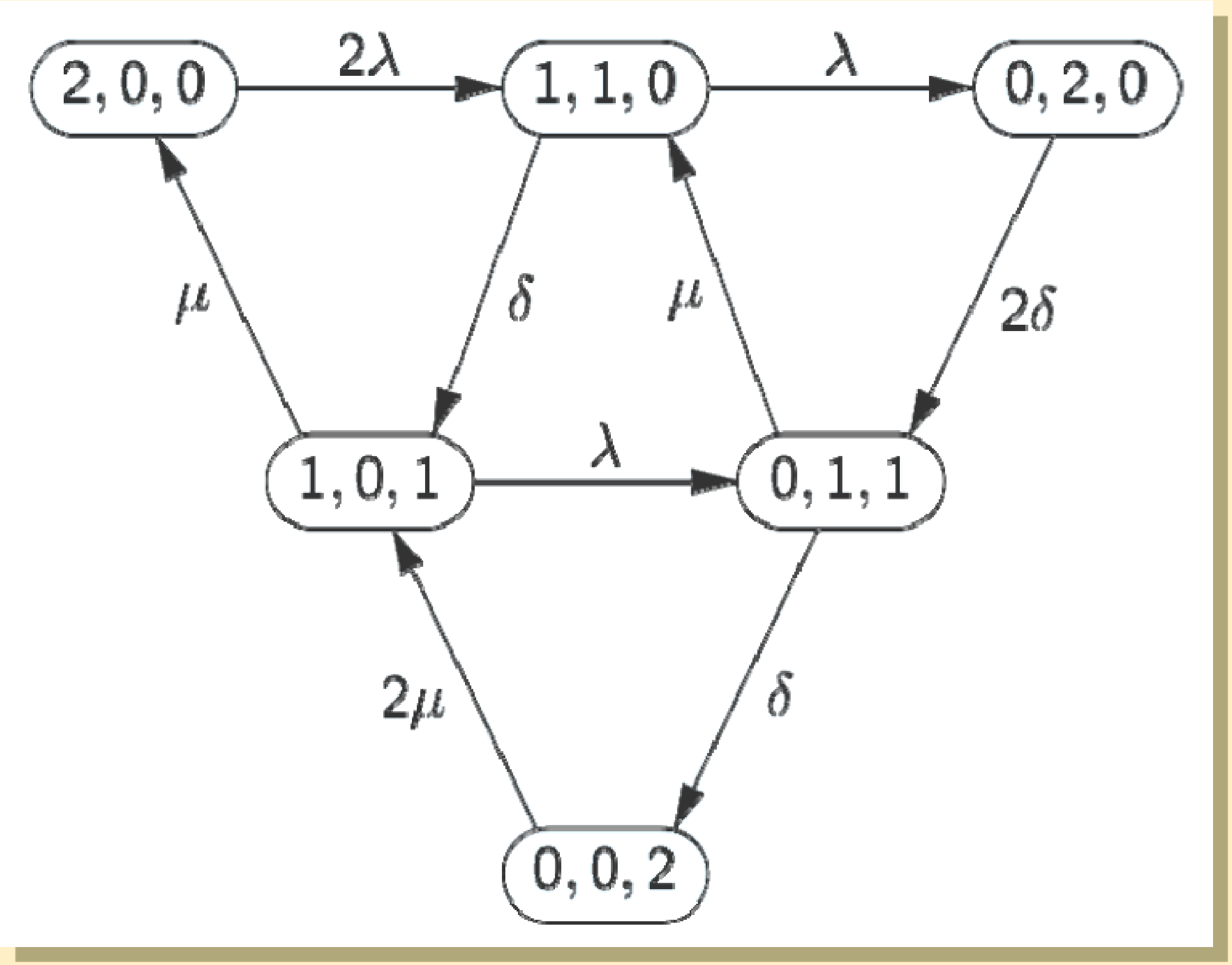
Example: Reliability model of a redundant system

- Reachability graph: (healthy, faulty, repair)



Example: Reliability model of a redundant system

- **Associated CTMC:** (healthy, faulty, repair)



Further classes of stochastic Petri nets

Generalized stochastic Petri Nets (GSPN)

- Extension of SPN:
  - Immediate transitions
    - Modeling logical behavior (as opposed to timed behavior)
    - Notation: filled black rectangles
  - Transition priorities
    - Conflict resolution: higher priority can fire
  - Inhibitor arcs
  - Guard expressions
    - Simplification by replacing some arcs with predicates
- Reachability graph is still represented by a CTMC
  - Vanishing markings: left by immediate transition firing, eliminated from CTMC
  - Tangible markings: CTMC states

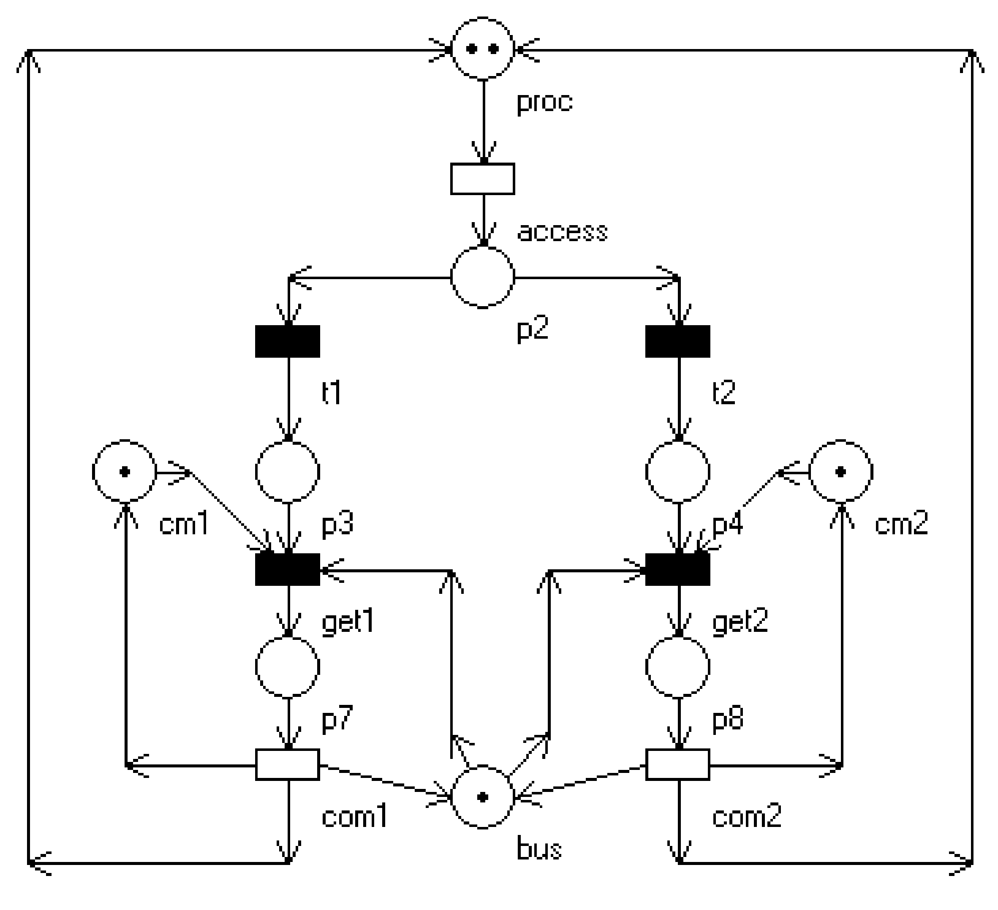
Formal definition

GSPN=(P, T, I, O, m_0, H, Π, L, G)

- $H \subseteq P \times T$ inhibitor arcs
- Π: $T \rightarrow Z$ priority
  - Timed transitions: priority = 0
  - Immediate transitions: priority >0, use for conflict resolution
- L: $T \rightarrow R^+$ transition parameters
  - Timed transitions: rate of exponential firing delay distribution
  - Immediate transitions: weight for random choice between immediate transition of the same priority
- G: $T \rightarrow$ *Boolean-formulas* transition guards
  - Must be satisfied for transition enablement
  - Predicate on markings,
    e.g. [m(P)>2], where m(P) is the marking of P

GSPN example

- Multiple processzor (proc)
  - Submit request for communication (access)
- Shared bust (bus) with communication ports (cm1, cm2)
  - Random choice between cm1 and cm2 according to weights
- Analysis questions:
  - Expected number of processors waiting for a communication port
  - Bus utilization
  - Port utilization
  - ...



Deterministic and stochastic Petri nets

- Further extensions:
  - Transitions with deterministic firing delay
    - Constant time to fire a transition after it is enabled
    - Modeling activities with deterministic (fixed) duration (e.g. repair time in a reliability model)
    - Notation: filled gray rectangle
- Efficient analysis is only possible if:
  - No more than one enabled deterministic transition in a marking
  - Required for Markovian analysis of the reachability graph

Timed Petri nets (TPN)

- General distributions for firing delays
- Reachability graph is generally not a CTMC
  - Structure may depend on distribution parameters
  - Markovian analysis is not possible
    - Analytic solution only for special cases
  - Usual solution: simulation
    - Difficult if there are many orders of magnitude differences between firing delays
      (e.g. time to failure is much larger than repair time)
- Resampling semantics in new markings
  - Firing distribution is not memoryless
  - It matters when is the resampling performed

Resampling semantics for timed transitions

- Conflict resolution methods:
  - Preselection: Does not depend on firing delay
  - Race: Transition with smallest firing delay wins (more frequently used)
- Variations of resampling races:

Resampling semantic:	If a transition remains enabled in the new marking	If a transition that previously lost enablement becomes enabled again in the new marking
"Race with resampling"	Sample according to the initial distirbution: "restart"	Sample according to the initial distirbution: "restart"
"Race with enabling memory"	Sample according to the remaining time: "continue"	Sample according to the initial distirbution: "restart"
"Race with age memory"	Sample according to the remaining time: "continue"	Sample according to the remaining time: "continue"

Stochatic reward nets

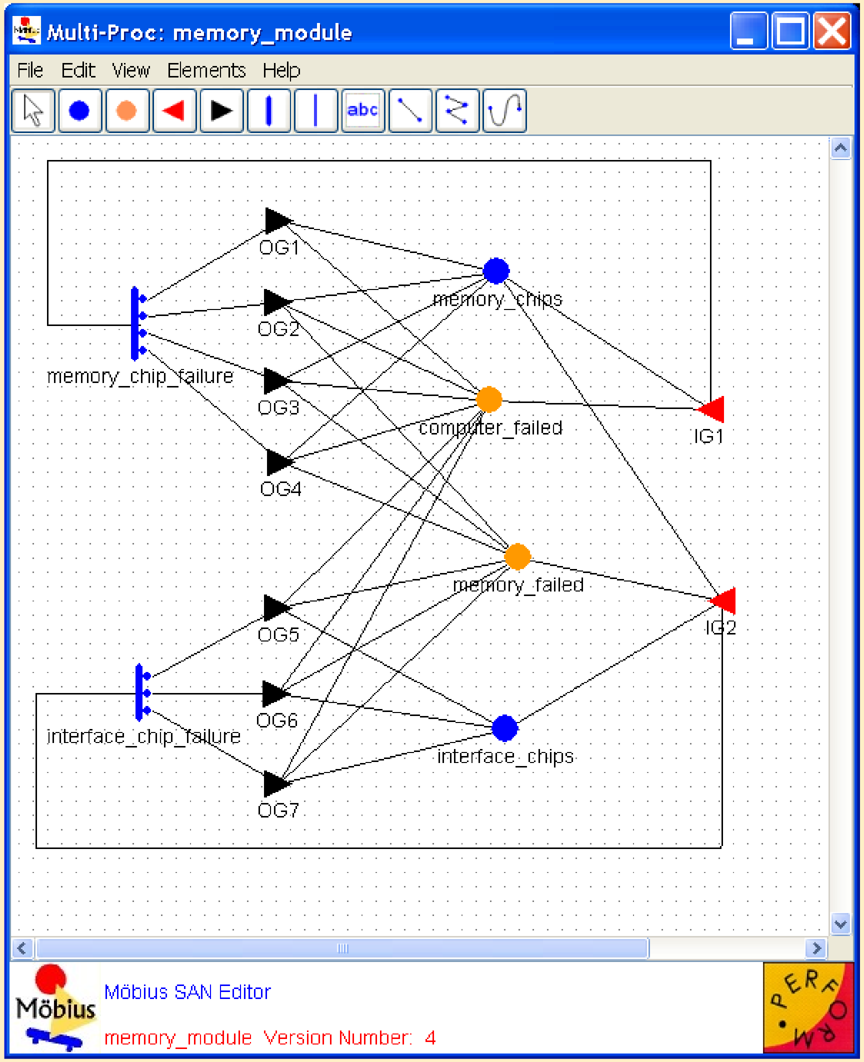
- Reward (or cost, i.e. negative reward) function
- Rate reward:
  - Reward/time unit as a function of the current marking
  - Reward accrued over an interval of time is determined by integration
  - Example: $30 profit, if the server is healthy, otherwise $20 penalty:

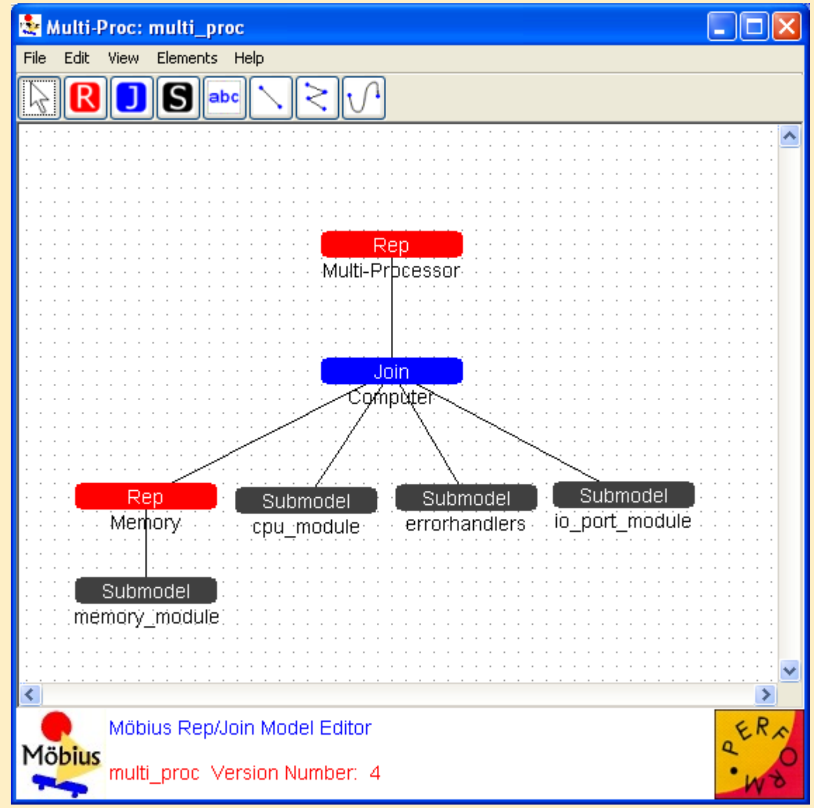
```
if (m(healthy)>0) then ra=30 else ra=-20
```

- Impulse reward:
  - Reward gained when firing a transition
  - Reward accrued over an interval of time is determined by summing the reward of individual firings
  - Example: Cost of repair is $500:

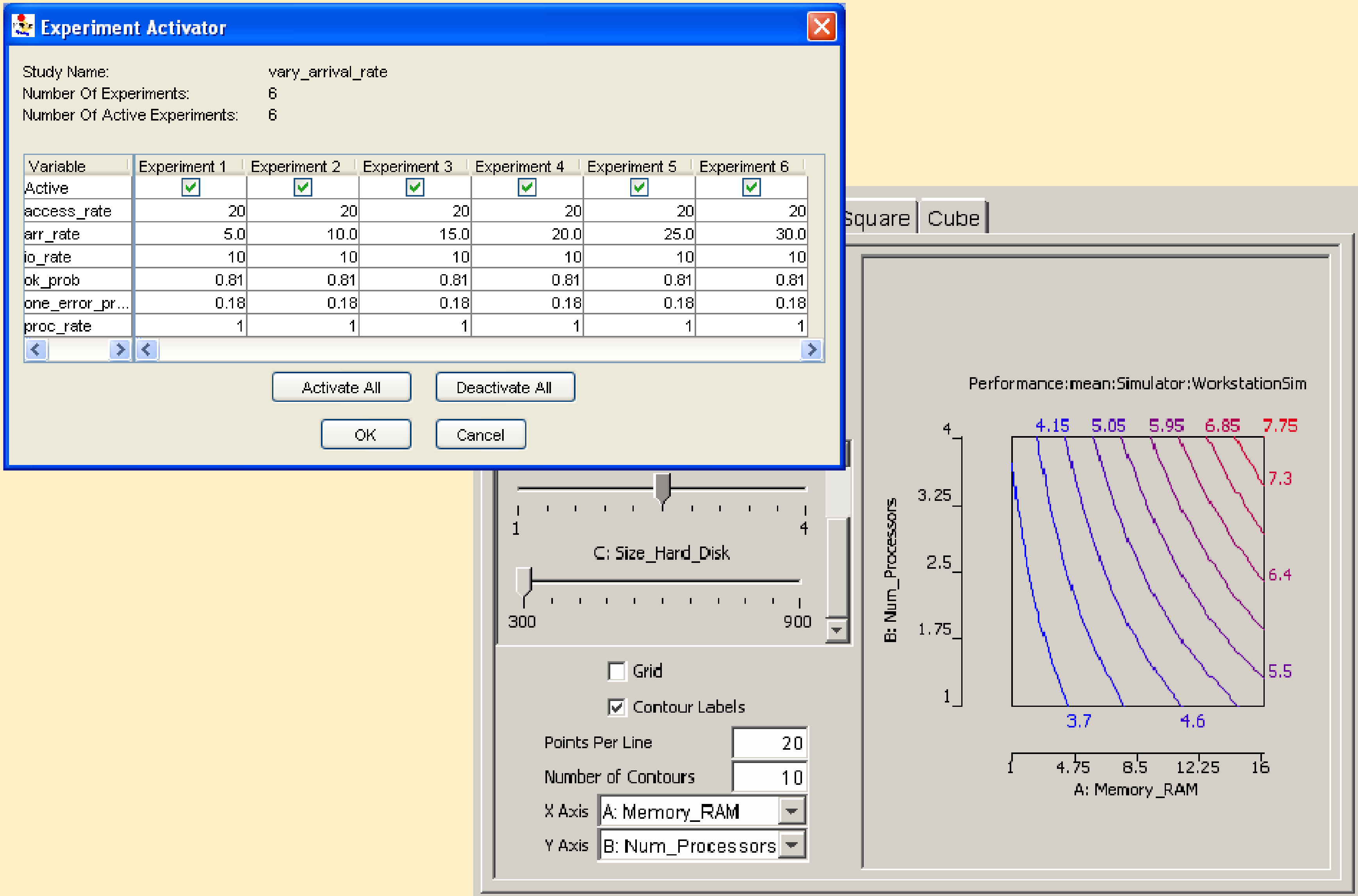
```
if (fire(Repair)) then ri=-500
```

Stochastic activity networks: Möbius tool





Stochastic activity networks: Möbius tool



Summary

- Motivation
- Stochastic processes and models
  - Continuous time Markov chains
- Stochastic Petri nets
  - SPN, GSPN, DSPN, TPN, SRN
  - Timing semantics