Engineering Genetic Circuits

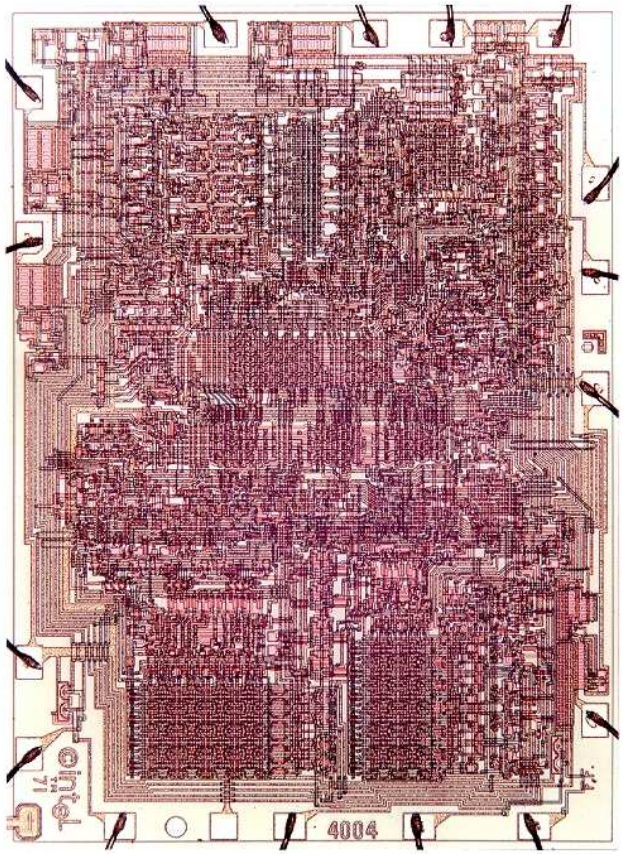
Chris J. Myers

Lecture 10: State-based Abstraction

The Law of Thumb

Somebody who thinks logically is a nice contrast to the real world.

Example Electrical Circuits

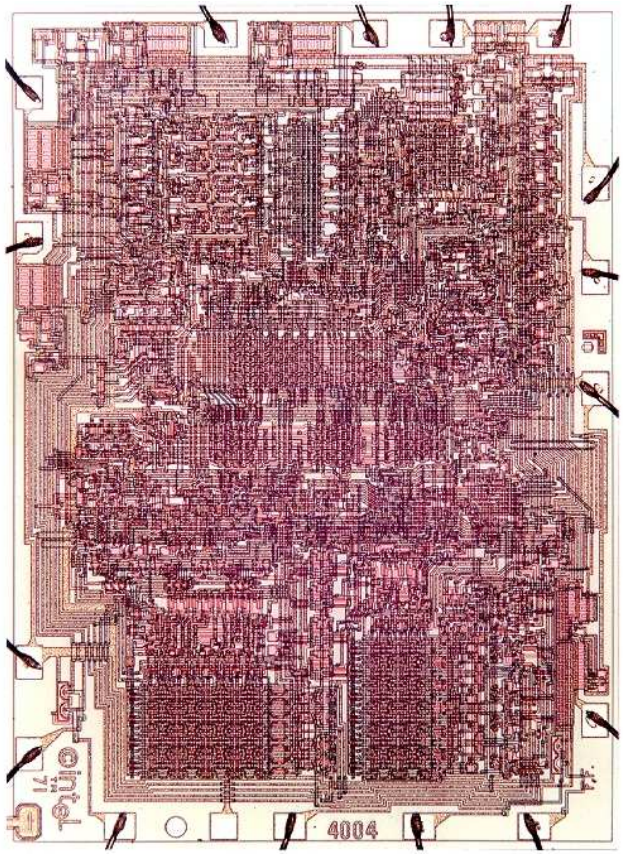


Intel 4004 (1971)
2,300 xtors / 108 KHz

Pentium 4 (2000)
42 million xtors / 1.5 GHz

(Courtesy of the Intel Museum)

Example Electrical Circuits



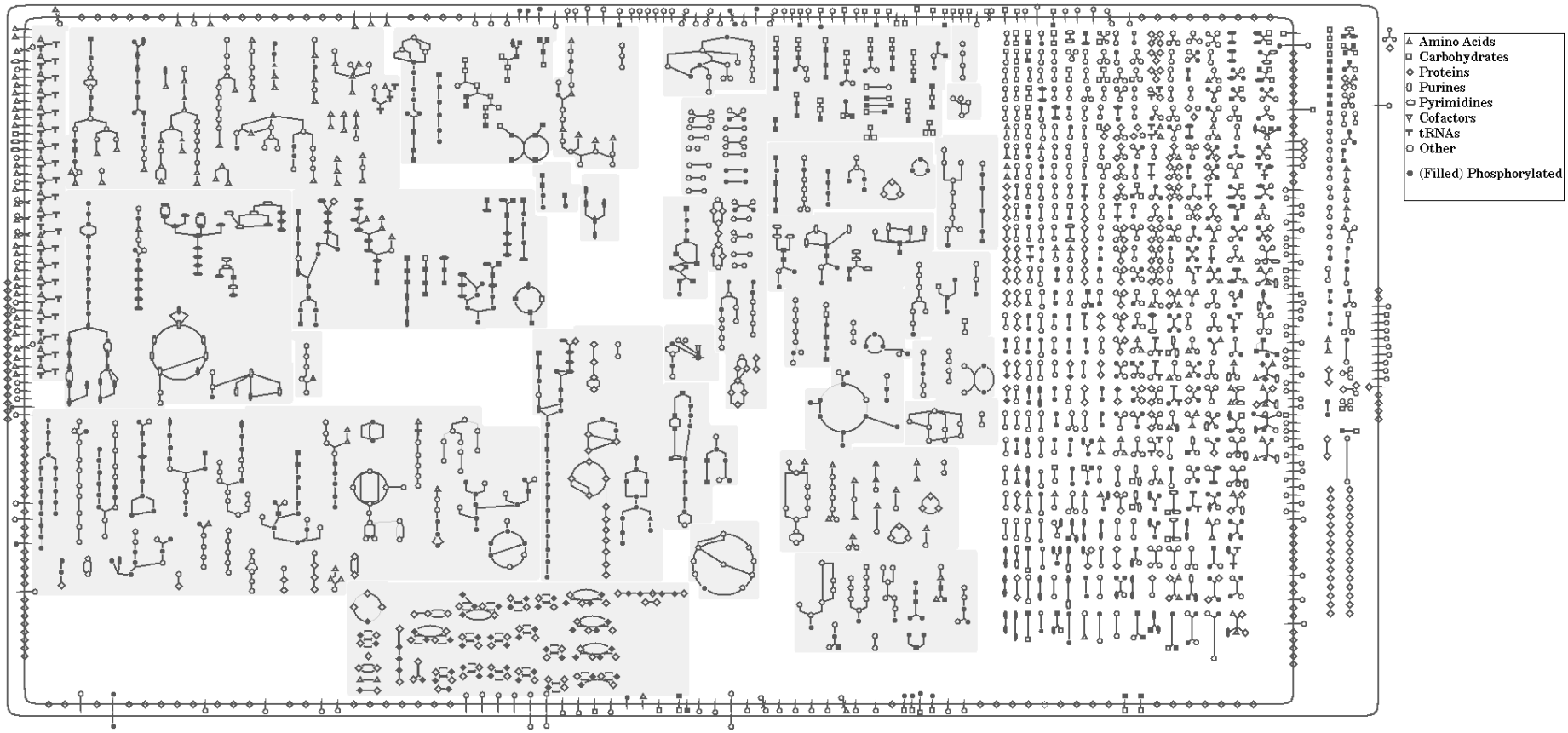
Intel 4004 (1971)
2,300 xtors / 108 KHz

Pentium 4 (2000)
42 million xtors / 1.5 GHz

If cars improved similarly, could now drive from SF to NYC in 13 seconds!

(Courtesy of the Intel Museum)

All Reactions in *E. coli* Metabolism



(Courtesy of `http://www.ecocyc.org`)

Logical Abstraction

- Electrical engineers routinely analyze circuits with thousands or even millions of interconnected complex components.
- Logical abstraction is essential to reason about such complex systems.
- Can logical abstraction be applied to biochemical circuits?
- Regulation of genetic circuits controlled by Hill functions.
- In the limit, these Hill functions become step functions which can be encoded logically.

Overview

- Logical encoding
- Piecewise models
- Stochastic finite-state machines
- Markov chain analysis
- Qualitative logical models

Logical Encoding

- Electrical circuits often classified as being either *analog* (i.e., having continuous valued states) or *digital* (i.e., having discrete valued states).
- Analog circuits must be analyzed using ODEs while digital circuits can be analyzed using *switch-level simulation*.
- Digital circuits are actually also analog circuits, but logical abstraction reduces their complexity of analysis.
- Logical abstraction essential since complex integrated circuits cannot be efficiently analyzed using ODEs.
- Can the efficiency of genetic circuit analysis also be improved using such a logical abstraction?

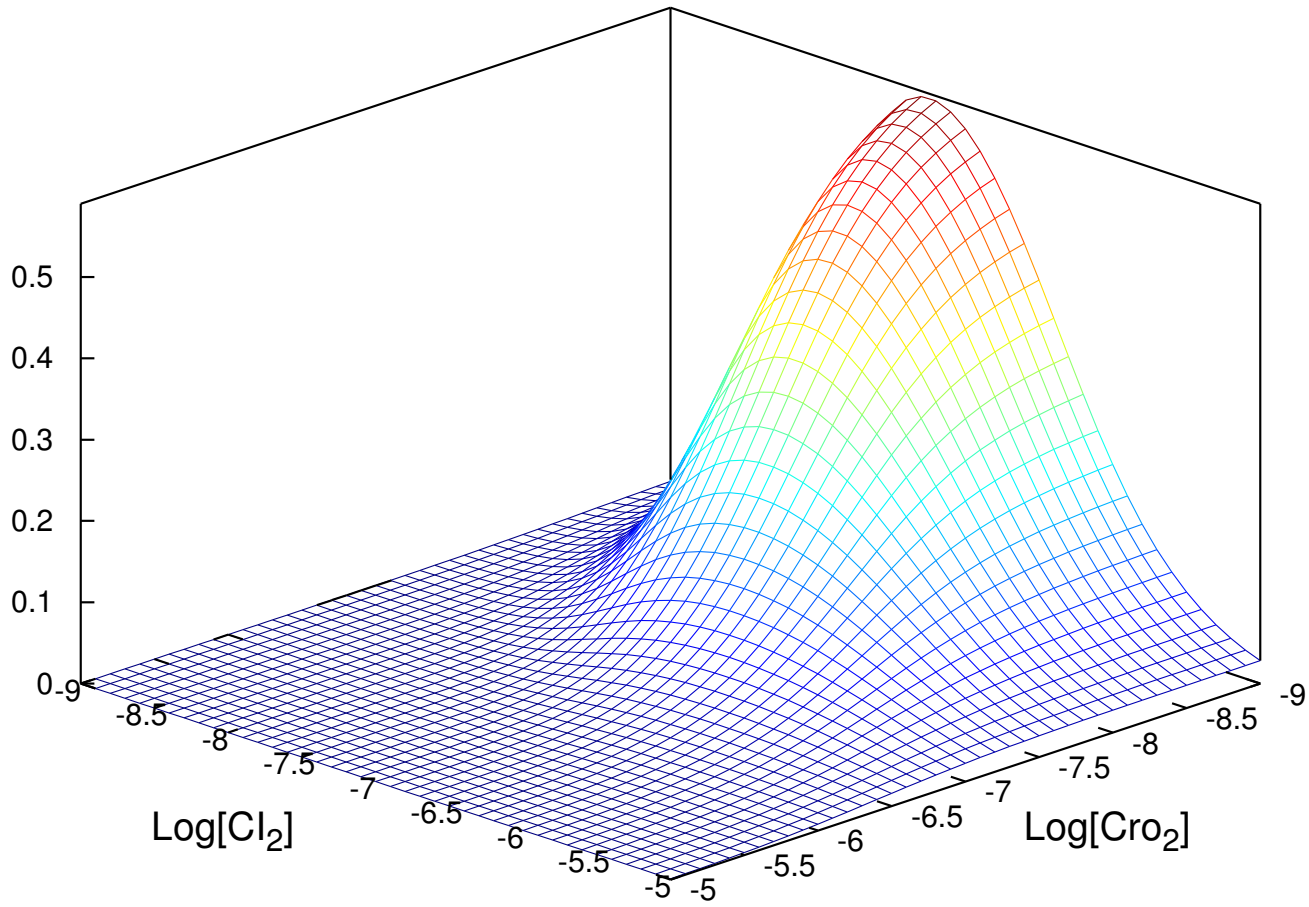
Hill Functions

- Inhibition and activation can be modeled with Hill functions:

$$\frac{1}{(1+K_j^n x_j^n)} \quad \text{OR} \quad \frac{K_j^n x_j^n}{(1+K_j^n x_j^n)}$$

  where $\theta_j = \sqrt[n]{a/(K_j - aK_j)}$ is the *critical threshold* where the change occurs, and a is an amplifier in the range of $[0.5, 1.0)$.
- As n increases, time spent in the transition region decreases and the function begins to behave like a step function.
- In this case, x_j could be encoded using a binary variable which is false when $x_j < \theta_j$ and true when $x_j \geq \theta_j$.

Activity of the P_{RM} Promoter



$[Cl_2]$ cannot be mapped into a single binary variable.

Critical Thresholds and Intervals

- Assume that species x_j has N_j thresholds $\theta_j^1, \ldots, \theta_j^i, \ldots, \theta_j^{N_j}$ that satisfy:

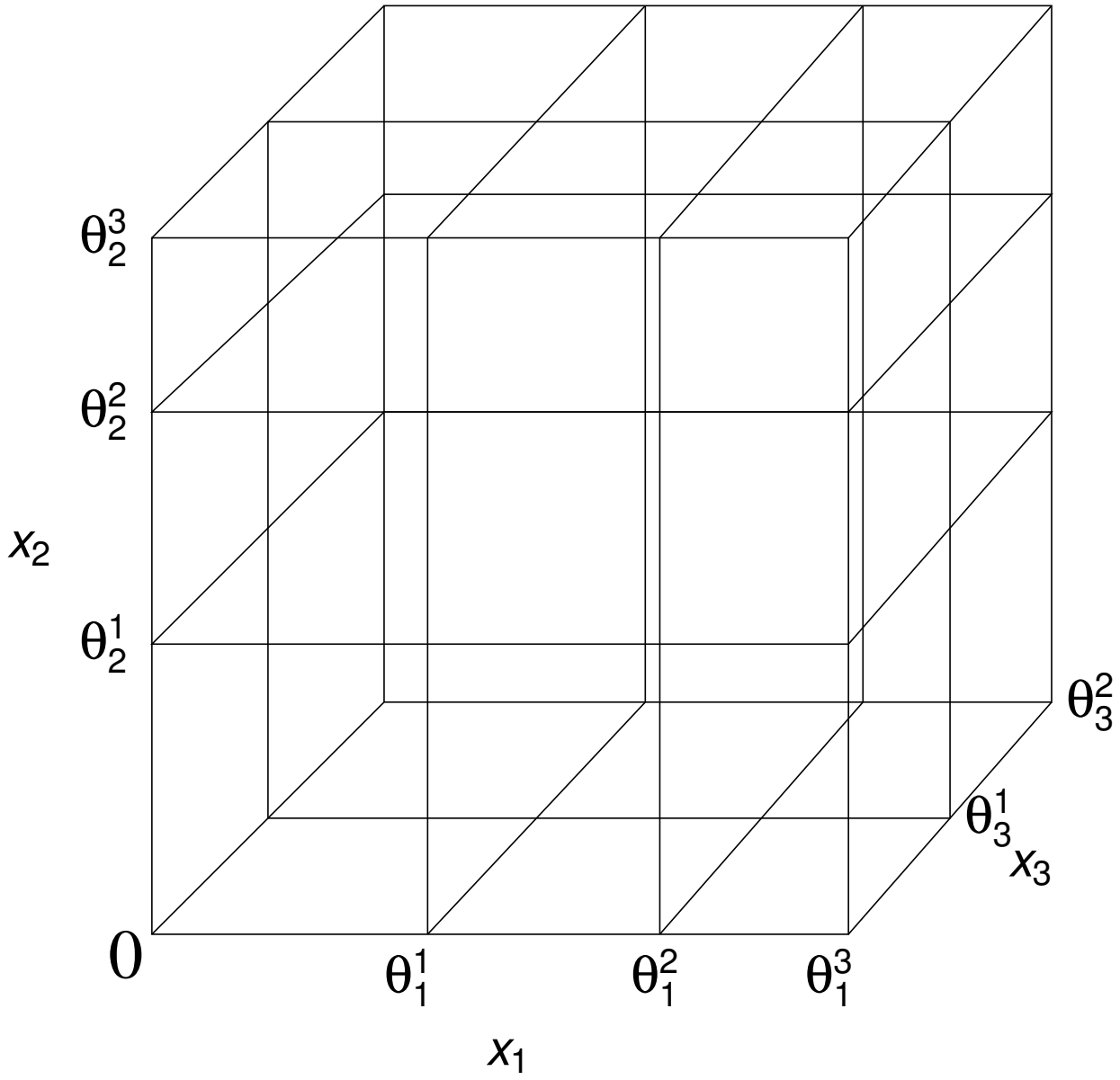
$$\theta_j^0 < \theta_j^1 < \ldots < \theta_j^i < \ldots < \theta_j^{N_j} < \theta_j^{N_j+1}$$

  where $\theta_j^0 = 0$ and $\theta_j^{N_j+1} = \infty$.
- States partitioned into *critical intervals* $(A_j^0, A_j^1, \ldots, A_j^{N_j})$ where $A_j^i = [\theta_j^i, \theta_j^{i+1})$.
- An n-ary variable b_j is created which can take any value in $\{0, 1, \ldots, N_j\}$.
- Initial value of b_j is the largest i such that $[x_j]_0 \geq \theta_j^i$ where $[x_j]_0$ is the initial concentration of x_j.
- Critical thresholds divide space into n-dimensional *regulatory domains* that are separated by hyperplanes $x_j = \theta_j^i$.
- The total number of these n-dimensional domains is:

$$\prod_{j=1}^{n}(N_j + 1)$$

- An assignment to each b_j uniquely selects an individual domain.

Regulatory Domains



Example Encoding for CI/CII Portion of Phage λ

- Abstracted model:

$$\frac{d[CI]}{dt} = \frac{np\ P_{RE} RNAP(k_b K_{o1} + k_a K_a [CII])}{1 + K_{o1} RNAP + K_a RNAP[CII]} - k_d[CI]$$

$$\frac{d[CII]}{dt} = \frac{np\ k_o P_R K_{o2} RNAP}{1 + K_{o2} RNAP + K_r K_d [CI]^2} - k_d[CII]$$

- Critical thresholds and intervals assuming the amplifier, a, is 0.5:

$$\theta^1_{CI} = \frac{1}{\sqrt{K_r K_d}} = 7, \qquad \theta^1_{CII} = \frac{1}{K_a RNAP} = 21$$

$$A^0_{CI} = [0,7), A^1_{CI} = [7,\infty), \qquad A^0_{CII} = [0,21), A^1_{CII} = [21,\infty)$$

Piecewise Linear Differential Equations

- *Piecewise linear differential equation* (PLDE):

$$\frac{dx_j}{dt} = f_j(\mathbf{x}) - g_j(\mathbf{x})x_j \quad j = 1, \ldots, n$$

where f_j and g_j are piecewise constant functions and $\mathbf{x} = [x_1, \ldots, x_n]$ is a vector of species concentrations.

- Each f_j and g_j changes value when a x_j crosses a threshold θ_j^i.

$$f_j(\mathbf{x}) \quad = \quad \sum_{l \in L} \alpha_{jl} B_{jl}(\mathbf{x}) \geq 0$$

where α_{jl} is a constant, and B_{jl} is composed of a conjunction of terms of the form $(b_k = i)$.

- Example:

$$\frac{d[CI]}{dt} \quad = \quad np\ P_{RE}(k_b + k_a(b_{CII} = 1)) - k_d[CI]$$

$$\frac{d[CII]}{dt} \quad = \quad np\ k_o P_R(b_{CI} = 0) - k_d[CII]$$

Solutions

- Inside each domain B, behavior is linear and quite simple.
- B is defined by a Boolean formula of the form: $(b_1 = i_1) \wedge \ldots \wedge (b_n = i_n)$ and denoted by a state vector of the form $(i_1 \ldots i_n)$.
- Denote $\mathbf{f} = [f_1, \ldots, f_n]$ within B by $\mathbf{f}^B$, and $\mathbf{g} = [g_1, \ldots, g_n]$ by $\mathbf{g}^B$.
- Within domain B, the behavior of $\mathbf{x}$ reduces to the simple linear differential equation $\frac{d\mathbf{x}}{dt} = \mathbf{f}^B - \mathbf{g}^B\mathbf{x}$ which has the solution:

$$\mathbf{X}(t) \quad = \quad \Phi^B + (\mathbf{X}(t_0) - \Phi^B)e^{\gamma(t_0 - t)} \quad \text{where } \Phi^B = \mathbf{f}^B / \mathbf{g}^B$$

Solutions: Example

- The domains are (00), (01), (10), and (11).
- The solutions are:

$$\Phi^{00} = \left(\frac{np\ k_o P_R}{k_d}, \frac{np\ k_b P_{RE}}{k_d}\right) = (19, 0.05)$$

$$\Phi^{01} = \left(0, \frac{np\ k_b P_{RE}}{k_d}\right) = (0, 0.05)$$

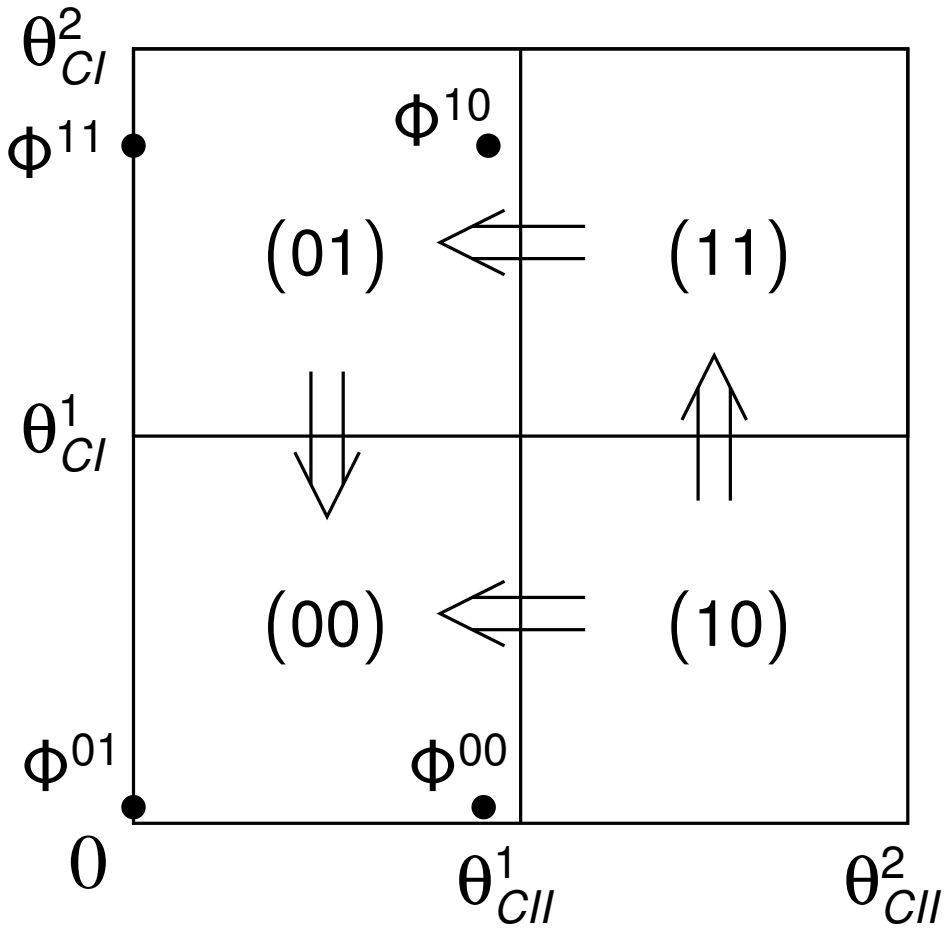
$$\Phi^{10} = \left(\frac{np\ k_o P_R}{k_d}, \frac{np\ P_{RE}(k_b + k_a)}{k_d}\right) = (19, 20)$$

$$\Phi^{11} = \left(0, \frac{np\ P_{RE}(k_b + k_a)}{k_d}\right) = (0, 20)$$

Black, White, and Transparent Boundaries

- As $t \to +\infty$, $\mathbf{X}(t)$ approaches Φ^B until it reaches a boundary.
- If Φ^B is within B, then $\mathbf{X}(t)$ reaches a stable stationary point at Φ^B.
- Assuming that B is bounded between θ_j^i and θ_j^{i+1}:
  - If $\Phi^B < \theta_j^i$, all trajectories in B that reach $x_j = \theta_j^i$ are leaving B.
  - If $\Phi^B > \theta_j^{i+1}$, all trajectories in B that reach $x_j = \theta_j^{i+1}$ are leaving B.
  - If $\theta_j^i < \Phi^B < \theta_j^{i+1}$, all trajectories that reach $x_j = \theta_j^i$ or $x_j = \theta_j^{i+1}$ enter B.
- A boundary between two domains is *transparent* if trajectories enter one domain and leave the other domain through this boundary.
- A boundary is *black* if trajectories leave both domains from this boundary.
- A boundary is *white* if trajectories enter both domains from this boundary.
- If a boundary is black or white, the result is a *sliding motion*.
- If the boundary is black, then the solution proceeds along the boundary until it either reaches another boundary or a stable point on the boundary.
- If the boundary is white, the solution can either proceed sliding along the boundary or leave it at any point, since a white wall is unstable.

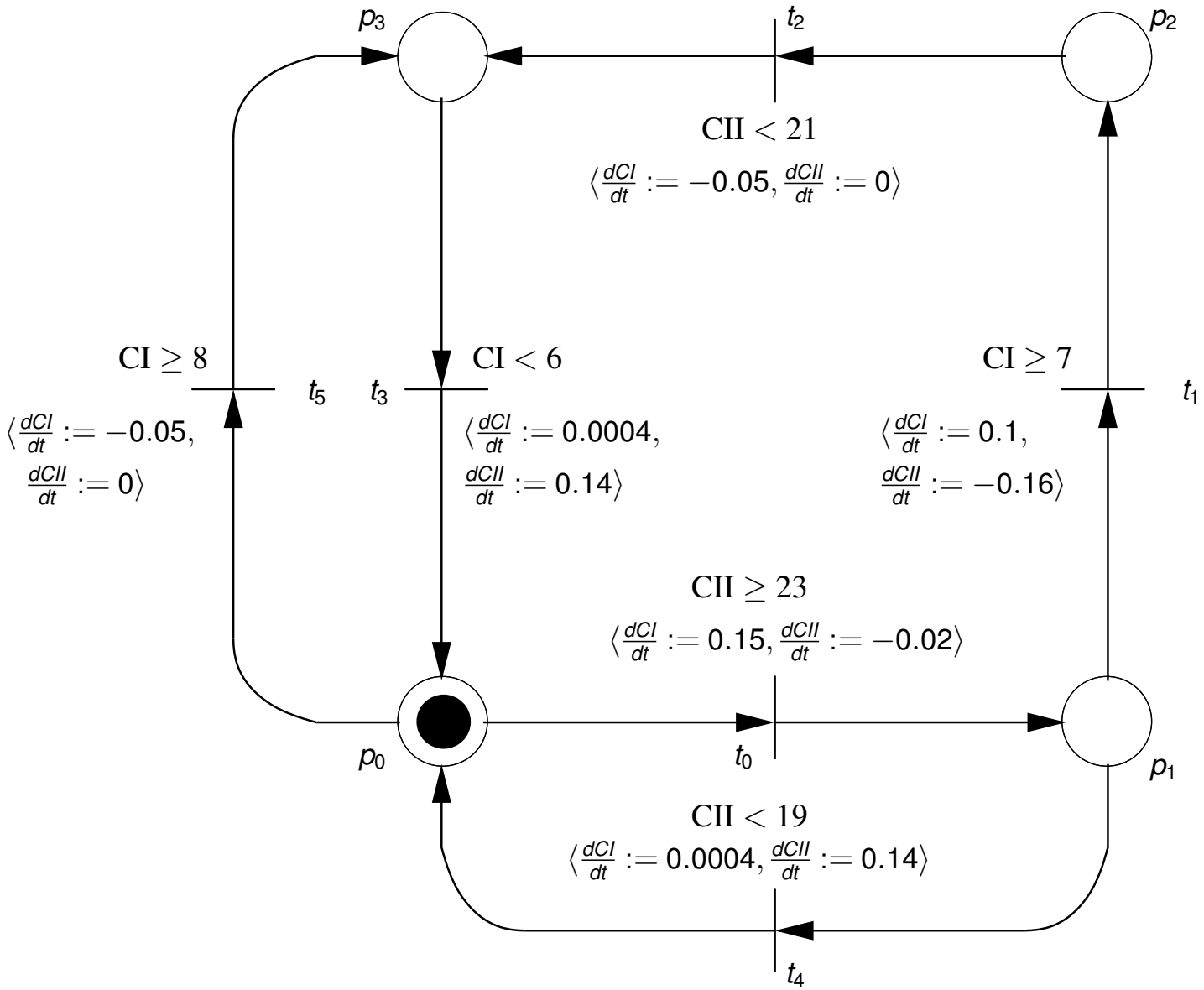
Example Flow Graph for CI/CII Portion of Phage λ



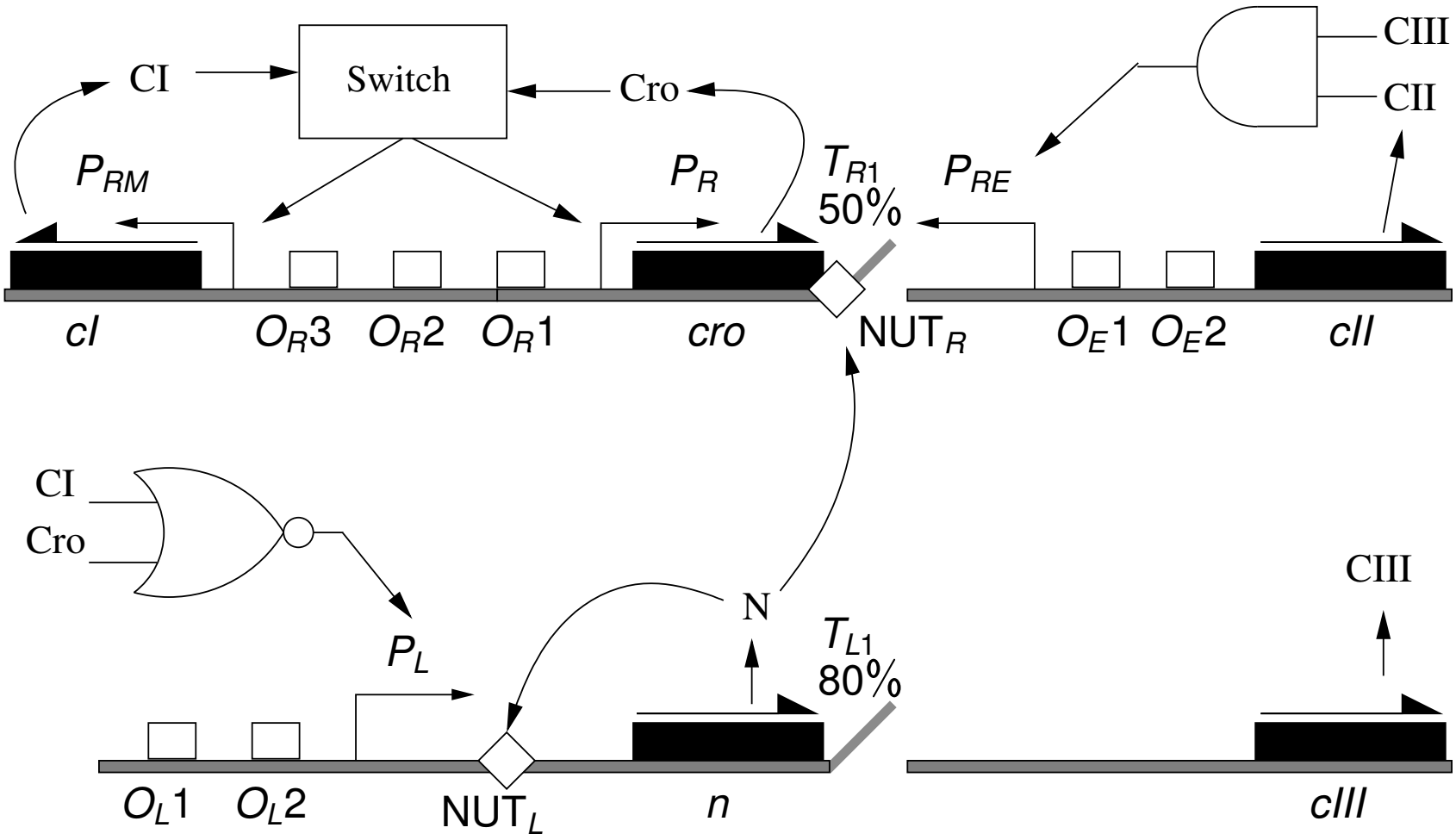
Labeled Hybrid Petri Nets

- *Hybrid Petri nets* can represent piecewise differential equation models.
- HPNs include both a discrete part that can model discrete states such as the current regulatory domain that the system is in as well as a continuous part that can model continuous quantities like species concentrations.
- Numerous ways to add continuous quantities to Petri nets.
- *Labeled hybrid Petri nets* (LHPNs) add the continuous values as auxiliary variables that evolve over time.
- These variables can be sampled in enabling conditions on transitions.
- Their rates of change can be modified by assignments on transitions.

LHPN Model of the CI/CII Portion of Phage λ



Piecewise Model for the Phage λ Decision Cirucit



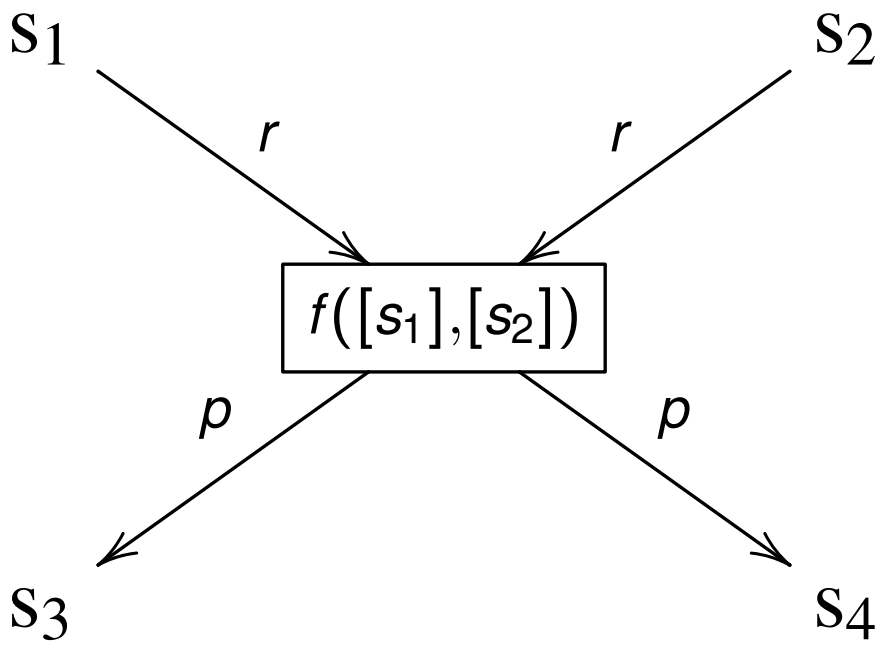
Stochastic Finite-State Machines

- Piecewise models still track exact concentrations of each species making their state space infinite.
- A *stochastic finite-state machine* (FSM) only tracks the n-ary encoding value for each species.
- Creates a purely logical representation of the genetic circuit.
- Analysis of a stochastic FSM can be accomplished using either stochastic simulation or Markov chain analysis.
- A stochastic FSM can often be efficiently analyzed while maintaining the high-level quantitative behavior.

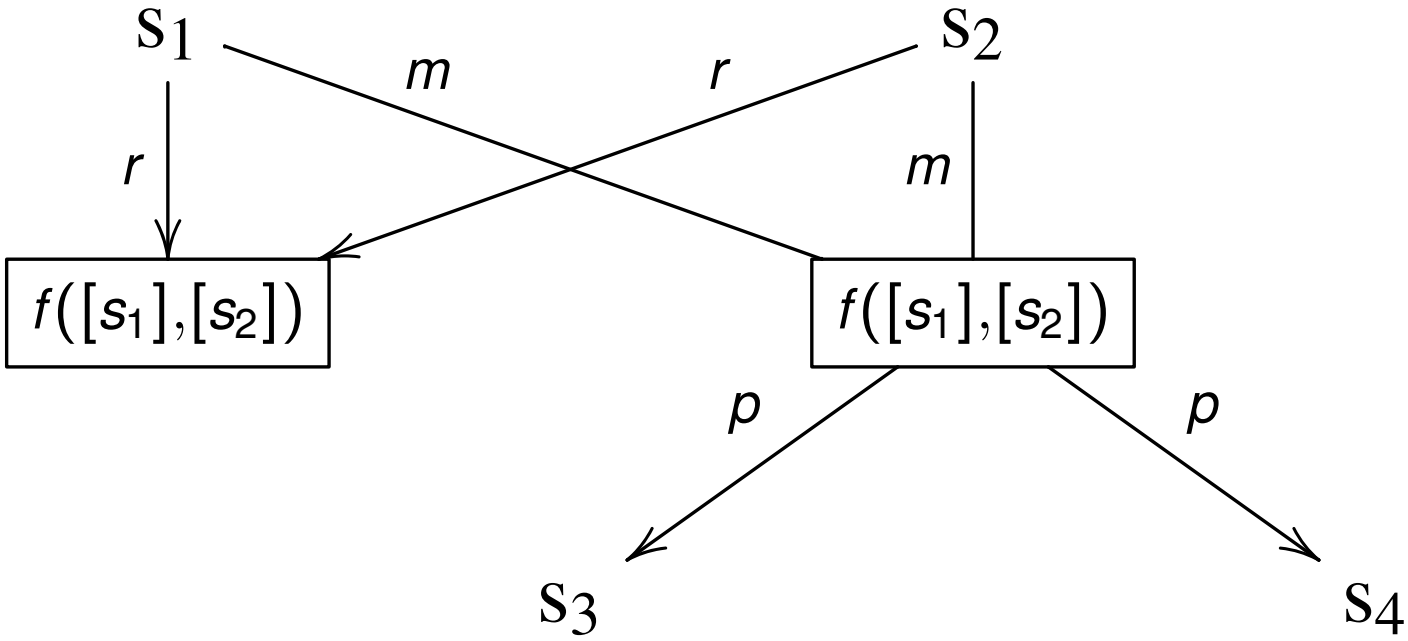
Reaction Splitization

- SAC model transformation requires reaction model to satisfy the property that all reactions have either one reactant *or* one product, but not both.
- Often true after applying the reaction-based abstractions, but if not, it can be made to hold using *reaction splitization*.

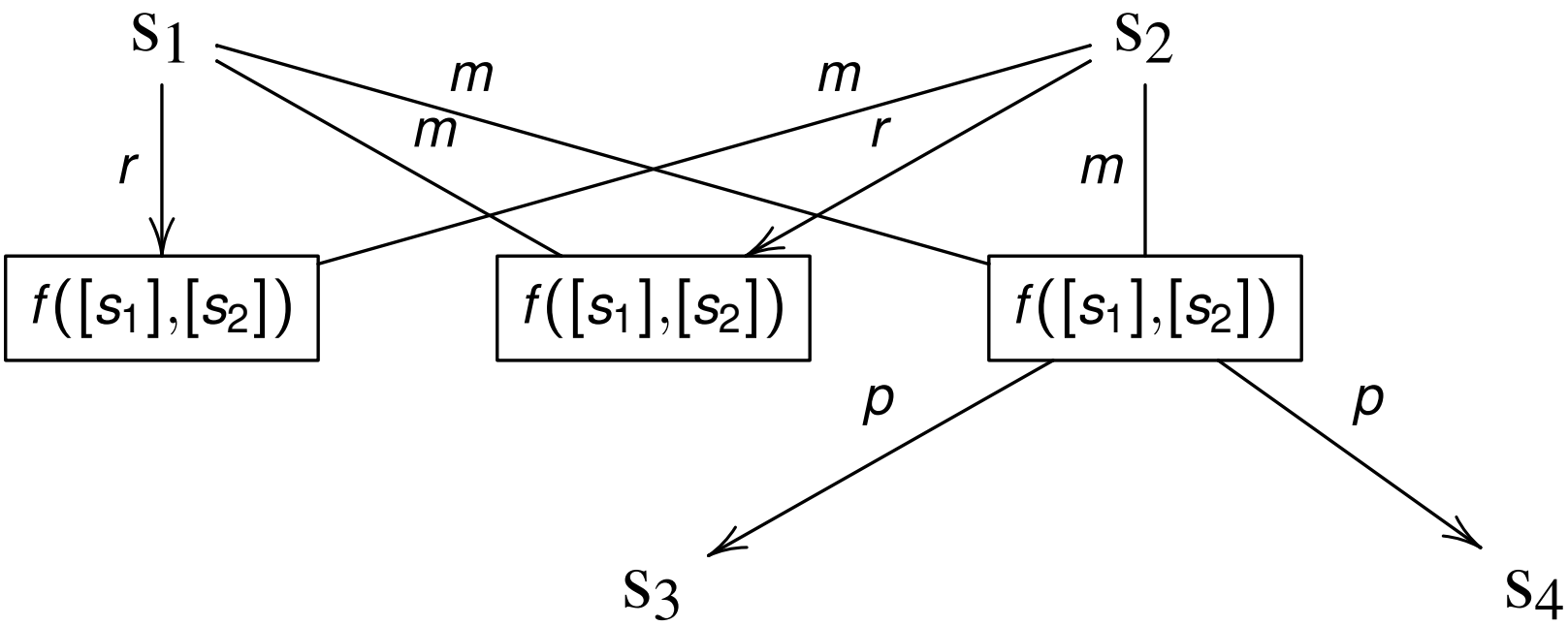
Reaction Splitization Example



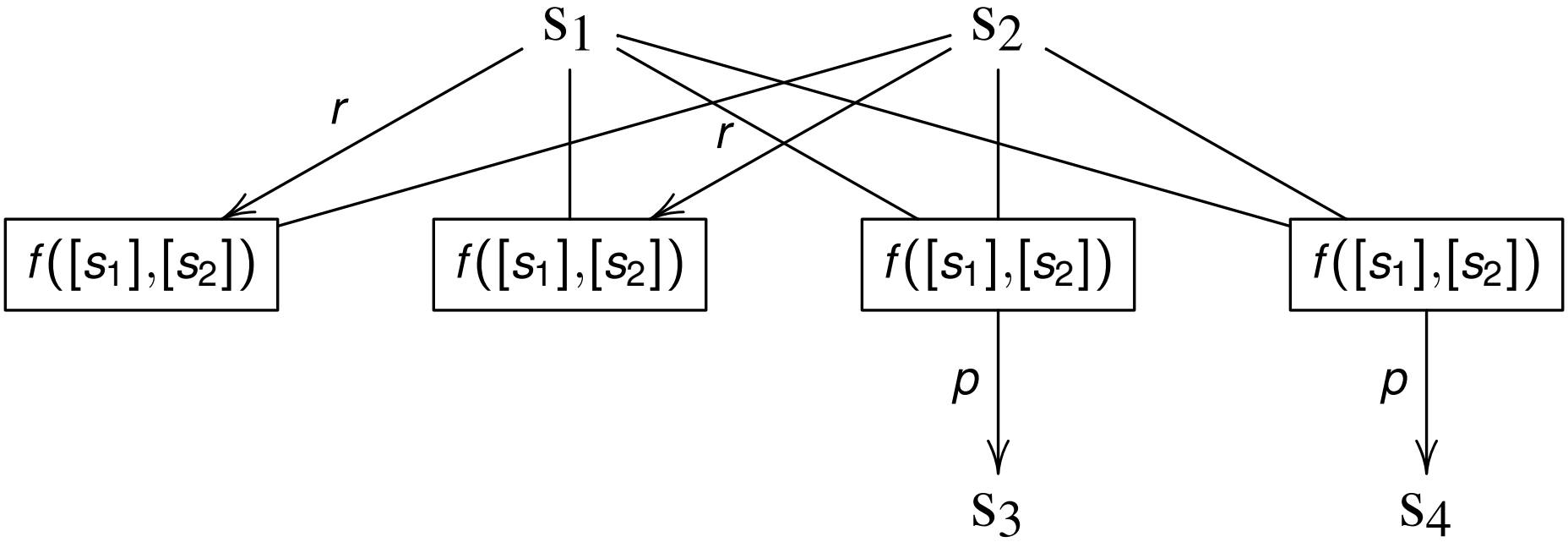
Reaction Splitization Example



Reaction Splitization Example



Reaction Splitization Example



Guarded Commands

- A stochastic FSM is specified using a set of *guarded commands*.
- Each guarded command, $c_k \in C$, has a form:

$$G_k(\mathbf{b}) \xrightarrow{q_k} b_j := i$$

  where the function $G_k(\mathbf{b})$ is the *guard*, q_k is the transition rate, and i is the n-ary value assigned to b_j as a result of c_k.
- A guard is a conjunction of literals of the form $(b_j = i)$.
- Each guarded command, c_k, is required to monotonically change the state of some variable in $\mathbf{b}$.
- If b_j is assigned to i by c_k, then the guard must include a term of the form $(b_j = i-1)$ or $(b_j = i+1)$.

Guarded Command Generation: CI

- Production of CI:

$$
\begin{aligned}
b_{CI} = 0 \wedge b_{CII} = 0 &\xrightarrow{q_1} b_{CI} := 1 \\
b_{CI} = 1 \wedge b_{CII} = 0 &\xrightarrow{q_2} b_{CI} := 2 \\
b_{CI} = 0 \wedge b_{CII} = 1 &\xrightarrow{q_3} b_{CI} := 1 \\
b_{CI} = 1 \wedge b_{CII} = 1 &\xrightarrow{q_4} b_{CI} := 2 \\
b_{CI} = 0 \wedge b_{CII} = 2 &\xrightarrow{q_5} b_{CI} := 1 \\
b_{CI} = 1 \wedge b_{CII} = 2 &\xrightarrow{q_6} b_{CI} := 2 \\
b_{CI} = 0 \wedge b_{CII} = 3 &\xrightarrow{q_7} b_{CI} := 1 \\
b_{CI} = 1 \wedge b_{CII} = 3 &\xrightarrow{q_8} b_{CI} := 2
\end{aligned}
$$

- Degradation of CI:

$$
\begin{aligned}
b_{CI} = 2 &\xrightarrow{q_9} b_{CI} := 1 \\
b_{CI} = 1 &\xrightarrow{q_{10}} b_{CI} := 0
\end{aligned}
$$

Guarded Command Generation: CII

- Production of CII:

$$
\begin{aligned}
b_{CI} = 0 \wedge b_{CII} = 0 &\xrightarrow{q_{11}} b_{CII} := 1 \\
b_{CI} = 0 \wedge b_{CII} = 1 &\xrightarrow{q_{12}} b_{CII} := 2 \\
b_{CI} = 0 \wedge b_{CII} = 2 &\xrightarrow{q_{13}} b_{CII} := 3 \\
b_{CI} = 1 \wedge b_{CII} = 0 &\xrightarrow{q_{14}} b_{CII} := 1 \\
b_{CI} = 1 \wedge b_{CII} = 1 &\xrightarrow{q_{15}} b_{CII} := 2 \\
b_{CI} = 1 \wedge b_{CII} = 2 &\xrightarrow{q_{16}} b_{CII} := 3 \\
b_{CI} = 2 \wedge b_{CII} = 0 &\xrightarrow{q_{17}} b_{CII} := 1 \\
b_{CI} = 2 \wedge b_{CII} = 1 &\xrightarrow{q_{18}} b_{CII} := 2 \\
b_{CI} = 2 \wedge b_{CII} = 2 &\xrightarrow{q_{19}} b_{CII} := 3
\end{aligned}
$$

- Degradation of CII:

$$
\begin{aligned}
b_{CII} = 3 &\xrightarrow{q_{20}} b_{CII} := 2 \\
b_{CII} = 2 &\xrightarrow{q_{21}} b_{CII} := 1 \\
b_{CII} = 1 &\xrightarrow{q_{22}} b_{CII} := 0
\end{aligned}
$$

Transition Rate Generation

- For each guarded command c_k that increases b_j to i:

$$q_k = \frac{m \cdot f_j(\Theta)}{\theta_j^i - \theta_j^{i-1}}$$

  where m is the stochiometry of s in the corresponding reaction, $f(\Theta)$ is the rate law for this reaction, and Θ is the critical levels that satisfy $G_k(\mathbf{b})$.

- For each guarded command c_k that decreases b_j to i:

$$q_k = \frac{m \cdot f_j(\Theta)}{\theta_j^{i+1} - \theta_j^i}$$

Transition Rate Generation Example: CI

- Production of CI:

$$b_{CI} = 0 \wedge b_{CII} = 0 \xrightarrow{q_1} b_{CI} := 1 \qquad q_1 = 10 \cdot f_3(0)/\theta_1^{CI}$$
$$b_{CI} = 1 \wedge b_{CII} = 0 \xrightarrow{q_2} b_{CI} := 2 \qquad q_2 = 10 \cdot f_3(0)/(\theta_2^{CI} - \theta_1^{CI})$$
$$b_{CI} = 0 \wedge b_{CII} = 1 \xrightarrow{q_3} b_{CI} := 1 \qquad q_3 = 10 \cdot f_3(\theta_1^{CII})/\theta_1^{CI}$$
$$b_{CI} = 1 \wedge b_{CII} = 1 \xrightarrow{q_4} b_{CI} := 2 \qquad q_4 = 10 \cdot f_3(\theta_1^{CII})/(\theta_2^{CI} - \theta_1^{CI})$$
$$b_{CI} = 0 \wedge b_{CII} = 2 \xrightarrow{q_5} b_{CI} := 1 \qquad q_5 = 10 \cdot f_3(\theta_2^{CII})/\theta_1^{CI}$$
$$b_{CI} = 1 \wedge b_{CII} = 2 \xrightarrow{q_6} b_{CI} := 2 \qquad q_6 = 10 \cdot f_3(\theta_2^{CII})/(\theta_2^{CI} - \theta_1^{CI})$$
$$b_{CI} = 0 \wedge b_{CII} = 3 \xrightarrow{q_7} b_{CI} := 1 \qquad q_7 = 10 \cdot f_3(\theta_3^{CII})/\theta_1^{CI}$$
$$b_{CI} = 1 \wedge b_{CII} = 3 \xrightarrow{q_8} b_{CI} := 2 \qquad q_8 = 10 \cdot f_3(\theta_3^{CII})/(\theta_2^{CI} - \theta_1^{CI})$$

- Degradation of CI:

$$b_{CI} = 2 \xrightarrow{q_9} b_{CI} := 1 \qquad q_9 = f_2(\theta_2^{CI})/(\theta_2^{CI} - \theta_1^{CI})$$
$$b_{CI} = 1 \xrightarrow{q_{10}} b_{CI} := 0 \qquad q_{10} = f_2(\theta_1^{CI})/\theta_1^{CI}$$

Stochastic FSM Simulation

- If the process being modeled is in the state $\mathbf{b}$, c_k can be executed if its guard is satisfied (i.e., $G_k(\mathbf{b})$ evaluates to true).
- The result of executing the guarded command is that a new state $\mathbf{b}'$ is reached in which $b_j' = i$ and $b_l' = b_l$ and for all $l \neq j$.
- The probability that c_k is executed is:

$$P(c_k) \quad = \quad G_k(\mathbf{b}) \cdot q_k \cdot \Delta t$$

  where Δt must be small enough such that the probability that two or more commands are executed in that time interval is negligible.
- The probability that no transition is taken in a Δt time step is:

$$(1 - (\sum_{k=0}^{|C|} G_k(\mathbf{s}) \cdot q_k \cdot \Delta t))$$

Stochastic FSM Simulation

- A stochastic FSM can be analyzed using multiple stochastic simulation runs beginning in state $\mathbf{b_0}$.
- In each state, simulation process determines whether or not to execute a guarded command in the next Δt.
- If guarded command executed, assignment is performed resulting in a new state, and transition probabilities are recalculated.
- Process continues until desired simulation time has been reached.
- This process is inefficient, since for small Δt, number of simulation steps that do not result in a state change increases significantly.
- More efficient to use SSA to jump to time of the next state change.
- Can also be efficiently analyzed using Markov chain analysis.

Markov Chains

- *Markov processes* satisfy the *Markov property*:

$$Pr[X(t+\tau) \leq y \mid X(s) = x(s), \forall s \leq t] = Pr[X(t+\tau) \leq y \mid X(t) = x(t)], \quad \forall \tau > 0$$

- Process is *memoryless* (i.e., the time that will be spent in a state is independent of the time already spent there).
- A *homogeneous* Markov process does not depend on the time t:

$$Pr[X(t+\tau) = y \mid X(t) = x] = Pr[X(\tau) = y \mid X(0) = x], \forall t, \tau > 0$$

- A *Markov chain* is a Markov process with a discrete state space.

Discrete-Time Markov Chains (DTMC)

- States observed only at discrete time points.
- Homogenous DTMC's specified by a *transition probability matrix*, P, composed of *single-step transition probabilities* of this form:

$$p_{ij} \quad = \quad Pr[X_{n+1} = j \mid X_n = i\,], \text{ for all } n = 0, 1, \ldots$$

  where $0 \leq p_{ij} \leq 1$ and $\sum_{\text{all j}} p_{ij} = 1$.
- Weather in Salt Lake City, Utah in January (snowy, overcast, clear):

$$P \quad = \quad \begin{pmatrix} 0.4 & 0.4 & 0.2 \\ 0.7 & 0.3 & 0.0 \\ 0.3 & 0.2 & 0.5 \end{pmatrix}$$

n-Step Transition Probabilities

- n-step transition probabilities can be obtained as follows:

$$P^n = PP^{(n-1)}$$

- Example:

$$P^2 = \begin{pmatrix} 0.5 & 0.32 & 0.18 \\ 0.49 & 0.37 & 0.14 \\ 0.41 & 0.28 & 0.31 \end{pmatrix}$$

n-Step Transition Probabilities

- n-step transition probabilities can be obtained as follows:

$$P^n = PP^{(n-1)}$$

- Example:

$$P^6 = P^\infty = \begin{pmatrix} 0.48 & 0.33 & 0.19 \\ 0.48 & 0.33 & 0.19 \\ 0.48 & 0.33 & 0.19 \end{pmatrix}$$

Classifications of States

- A state is *transient* when there is a non-zero probability that the DTMC will at some point never return to that state.
- A state is *recurrent* when the DTMC is guaranteed to return to this state at some point in the future.
- A state is *positive-recurrent* when its mean time to revisit is finite.
- A state is *null-recurrent* when its mean time to revisit is infinite.
- In a finite Markov chain, all states are transient or positive-recurrent.
- A state j is *periodic with period* p when upon leaving j it can only be returned to after a number of transitions that is a multiple of $p > 1$.
- A state with $p = 1$ is *aperiodic*.
- An *ergodic* Markov chain is positive-recurrent and aperiodic.
- A DTMC is *irreducible* if every state can be reached by every other state.
- A finite, aperiodic, irreducible Markov chain is ergodic.

Limiting and Steady-State Distributions

- Often interested in determining the probability of being in a state:

$$\pi_i(n) \quad = \quad Pr[X_n = i\,]$$

- Probability vector for all states is written is follows:

$$\pi(n) \quad = \quad [\pi_1(n), \pi_2(n), \ldots, \pi_i(n), \ldots]$$

- The limit as n goes to ∞ is the *limiting distribution*:

$$\pi \quad = \quad \lim_{n \to \infty} \pi(n)$$

- In an ergodic Markov chain, limiting distribution is also known as a *steady-state distribution* and satisfies:

$$\pi \quad = \quad \pi P$$

Computing the Steady-State Distribution

- Can multiply P by itself repetitively or squaring the matrix.
- Drawback is the time and memory requirements for matrix multiplication.
- P is typically a very large and sparse matrix (i.e., many entries are zero).
- After squaring, zero entries take non-zero values.
- If kept sparse, P can be represented with sparse matrix data structure.
- Numerous methods developed to find the steady-state distribution.
- Two types of methods presented here:
  - *Direct methods*
  - *Iterative methods*

Direct Methods

- Solve system of equations given by $\pi = \pi P$.
- Use Gaussian elimination or other methods.
- Example:

$$[s\ o\ c] \quad = \quad [s\ o\ c] \begin{pmatrix} 0.4 & 0.4 & 0.2 \\ 0.7 & 0.3 & 0.0 \\ 0.3 & 0.2 & 0.5 \end{pmatrix}$$

Direct Methods

- Solve system of equations given by $\pi = \pi P$.
- Use Gaussian elimination or other methods.
- Example:

$$[s\ o\ c] = [(0.4s + 0.7o + 0.3c)(0.4s + 0.3o + 0.2c)(0.2s + 0.5c)]$$

Direct Methods

- Solve system of equations given by $\pi = \pi P$.
- Use Gaussian elimination or other methods.
- Example:

$$\begin{aligned} s &= 0.4s + 0.7o + 0.3c \\ o &= 0.4s + 0.3o + 0.2c \\ c &= 0.2s + 0.5c \end{aligned}$$

Direct Methods

- Solve system of equations given by $\pi = \pi P$.
- Use Gaussian elimination or other methods.
- Example:

$$\begin{aligned} s &= 0.48 \\ o &= 0.33 \\ c &= 0.19 \end{aligned}$$

Iterative Methods

- Iterate using $\pi_n = \pi_{n-1} P$ until convergence.
- Convergence can be slow and periodicity must be determined.
- Convergence rate dependent on initial state vector.
- Example:

$$\pi(1) = [1.0 \;\; 0.0 \;\; 0.0] \begin{pmatrix} 0.4 & 0.4 & 0.2 \\ 0.7 & 0.3 & 0.0 \\ 0.3 & 0.2 & 0.5 \end{pmatrix}$$

$$\pi(1) = [0.4 \;\; 0.4 \;\; 0.2]$$

Iterative Methods

- Iterate using $\pi_n = \pi_{n-1} P$ until convergence.
- Convergence can be slow and periodicity must be determined.
- Convergence rate dependent on initial state vector.
- Example:

$$\pi(2) = [0.4 \;\; 0.4 \;\; 0.2] \begin{pmatrix} 0.4 & 0.4 & 0.2 \\ 0.7 & 0.3 & 0.0 \\ 0.3 & 0.2 & 0.5 \end{pmatrix}$$

$$\pi(2) = [0.5 \;\; 0.32 \;\; 0.18]$$

Iterative Methods

- Iterate using $\pi_n = \pi_{n-1} P$ until convergence.
- Convergence can be slow and periodicity must be determined.
- Convergence rate dependent on initial state vector.
- Example:

$$\begin{aligned} \pi(3) &= [0.5 \;\; 0.32 \;\; 0.18] \begin{pmatrix} 0.4 & 0.4 & 0.2 \\ 0.7 & 0.3 & 0.0 \\ 0.3 & 0.2 & 0.5 \end{pmatrix} \\ \pi(3) &= [0.48 \;\; 0.33 \;\; 0.19] \end{aligned}$$

Iterative Methods

- Iterate using $\pi_n = \pi_{n-1} P$ until convergence.
- Convergence can be slow and periodicity must be determined.
- Convergence rate dependent on initial state vector.
- Example:

$$\pi(4) = [0.48 \;\; 0.33 \;\; 0.19] \begin{pmatrix} 0.4 & 0.4 & 0.2 \\ 0.7 & 0.3 & 0.0 \\ 0.3 & 0.2 & 0.5 \end{pmatrix}$$

$$\pi(4) = [0.48 \;\; 0.33 \;\; 0.19]$$

Convergence

- When iterations are converging rapidly, can check:

$$\|\pi(k) - \pi(k-1)\| \quad < \quad \varepsilon$$

  where $\|\mathbf{x}\| = \sqrt{x_1^2 + \ldots + x_n^2}$ and ε is the desired accuracy.
- When iterations are converging slowly, it is better to check:

$$\|\pi(k) - \pi(k-m)\| \quad < \quad \varepsilon$$

  where m should be set based on the convergence rate.
- Problems occur when probabilities are small, so should normalize:

$$\max_i \left(\frac{|\pi_i(k) - \pi_i(k-m)|}{|\pi_i(k)|} \right) \quad < \quad \varepsilon$$

Periodicity

- The period of a irreducible Markov chain is:

$$p = \gcd(l_1, \ldots, l_i, \ldots, l_c)$$

  where gcd is the greatest common divisor, l_i is the length of the i^{th} cycle in the Markov chain, and c is the total number of cycles.
- A periodic Markov chain (i.e., $p > 1$) does not converge for all m.
- Example:

$$\pi(1) = [1.0 \;\; 0.0 \;\; 0.0] \begin{pmatrix} 0.0 & 0.7 & 0.3 \\ 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \end{pmatrix}$$

$$\pi(1) = [0.0 \;\; 0.7 \;\; 0.3]$$

Periodicity

- The period of a irreducible Markov chain is:

$$p = \gcd(l_1, \ldots, l_i, \ldots, l_c)$$

  where gcd is the greatest common divisor, l_i is the length of the i^{th} cycle in the Markov chain, and c is the total number of cycles.
- A periodic Markov chain (i.e., $p > 1$) does not converge for all m.
- Example:

$$\pi(2) = [0.0 \;\; 0.7 \;\; 0.3] \begin{pmatrix} 0.0 & 0.7 & 0.3 \\ 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \end{pmatrix}$$

$$\pi(2) = [1.0 \;\; 0.0 \;\; 0.0]$$

Periodicity

- The period of a irreducible Markov chain is:

$$p = \gcd(l_1, \ldots, l_i, \ldots, l_c)$$

  where gcd is the greatest common divisor, l_i is the length of the i^{th} cycle in the Markov chain, and c is the total number of cycles.
- A periodic Markov chain (i.e., $p > 1$) does not converge for all m.
- Example:

$$\pi(3) = [1.0 \;\; 0.0 \;\; 0.0] \begin{pmatrix} 0.0 & 0.7 & 0.3 \\ 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \end{pmatrix}$$

$$\pi(3) = [0.0 \;\; 0.7 \;\; 0.3]$$

Periodicity: Simple Solution

- Check convergence only after each p steps.
- When converged, combine steady state distributions from last p steps.
- Normalize by p.
- Example:

$$\begin{aligned} \pi &= [1.0 \;\; 0.7 \;\; 0.3]/2 \\ &= [0.5 \;\; 0.35 \;\; 0.15] \end{aligned}$$

Continuous-Time Markov Chains (CTMC)

- States can change at any arbitrary point in time.
- Specified using a *transition rate matrix* rather than a transition probability matrix which is defined as follows for a homogeneous CTMC:

$$\begin{aligned} p_{ij}(\tau) &= Pr[X(s+\tau) = j \mid X(s) = i\,] \\ q_{ij} &= \lim_{\Delta t \to 0} \left\{ \frac{p_{ij}(\Delta t)}{\Delta t} \right\}, \text{ for } i \neq j \\ q_{ii} &= -\sum_{j \neq i} q_{ij} \end{aligned}$$

- Example:

$$Q = \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix}$$

Embedded Markov Chain

- Can find the steady-state distribution of a CTMC using its discrete-time *embedded Markov chain* (EMC).

$$\pi = \frac{-\phi D_Q^{-1}}{\|\phi D_Q^{-1}\|_1}$$

where ϕ is steady-state distribution for the EMC, D_Q^{-1} is the inverse of the diagonal matrix of Q, and $\|\phi D_Q^{-1}\|_1$ is the *norm* of the ϕD_Q^{-1} vector.

- Probabilities for an EMC for Q are defined as follows:

$$s_{ij} = \begin{cases} \frac{q_{ij}}{\sum_{i \neq j} q_{ij}}, & i \neq j \\ 0 & i = j \end{cases}$$

Embedded Markov Chain Example

$$Q = \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix}$$

Embedded Markov Chain Example

$$Q = \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix}$$

$$S = \begin{pmatrix} 0 & 0.67 & 0.33 \\ 1 & 0 & 0 \\ 0.5 & 0.5 & 0 \end{pmatrix}$$

Embedded Markov Chain Example

$$S = \begin{pmatrix} 0 & 0.67 & 0.33 \\ 1 & 0 & 0 \\ 0.5 & 0.5 & 0 \end{pmatrix}$$

$$\phi = [0.46 \;\; 0.39 \;\; 0.15]$$

Embedded Markov Chain Example

$$Q = \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix}$$

$$-D_Q^{-1} = \begin{pmatrix} 0.167 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.125 \end{pmatrix}$$

Embedded Markov Chain Example

$$
\begin{aligned}
\phi &= [0.46 \;\; 0.39 \;\; 0.15] \\
-D_Q^{-1} &= \begin{pmatrix} 0.167 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.125 \end{pmatrix} \\
-\phi D_Q^{-1} &= [0.08 \;\; 0.1 \;\; 0.02]
\end{aligned}
$$

Embedded Markov Chain Example

$$-\phi D_Q^{-1} = [0.08 \;\; 0.1 \;\; 0.02]$$

$$\|\phi D_Q^{-1}\|_1 = 0.2$$

Embedded Markov Chain Example

$$
\begin{aligned}
-\phi D_Q^{-1} &= [0.08 \;\; 0.1 \;\; 0.02] \\
\|\phi D_Q^{-1}\|_1 &= 0.2 \\
\pi = \frac{-\phi D_Q^{-1}}{\|\phi D_Q^{-1}\|_1} &= [0.4 \;\; 0.5 \;\; 0.1]
\end{aligned}
$$

Guarded Commands for CI

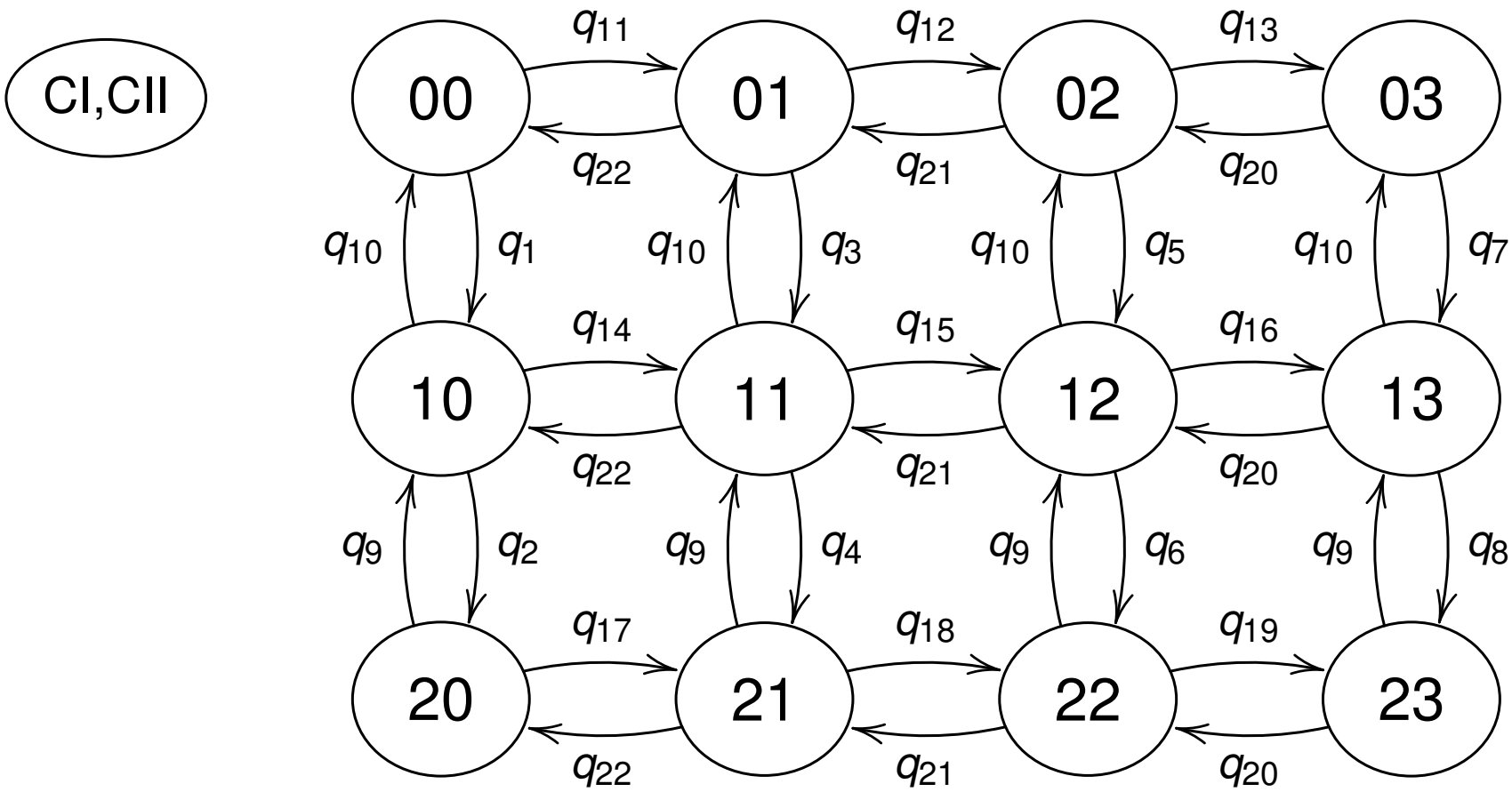
- Production of CI:

$$b_{CI} = 0 \wedge b_{CII} = 0 \xrightarrow{q_1} b_{CI} := 1 \qquad q_1 = 10 \cdot f_3(0)/\theta_1^{CI}$$
$$b_{CI} = 1 \wedge b_{CII} = 0 \xrightarrow{q_2} b_{CI} := 2 \qquad q_2 = 10 \cdot f_3(0)/(\theta_2^{CI} - \theta_1^{CI})$$
$$b_{CI} = 0 \wedge b_{CII} = 1 \xrightarrow{q_3} b_{CI} := 1 \qquad q_3 = 10 \cdot f_3(\theta_1^{CII})/\theta_1^{CI}$$
$$b_{CI} = 1 \wedge b_{CII} = 1 \xrightarrow{q_4} b_{CI} := 2 \qquad q_4 = 10 \cdot f_3(\theta_1^{CII})/(\theta_2^{CI} - \theta_1^{CI})$$
$$b_{CI} = 0 \wedge b_{CII} = 2 \xrightarrow{q_5} b_{CI} := 1 \qquad q_5 = 10 \cdot f_3(\theta_2^{CII})/\theta_1^{CI}$$
$$b_{CI} = 1 \wedge b_{CII} = 2 \xrightarrow{q_6} b_{CI} := 2 \qquad q_6 = 10 \cdot f_3(\theta_2^{CII})/(\theta_2^{CI} - \theta_1^{CI})$$
$$b_{CI} = 0 \wedge b_{CII} = 3 \xrightarrow{q_7} b_{CI} := 1 \qquad q_7 = 10 \cdot f_3(\theta_3^{CII})/\theta_1^{CI}$$
$$b_{CI} = 1 \wedge b_{CII} = 3 \xrightarrow{q_8} b_{CI} := 2 \qquad q_8 = 10 \cdot f_3(\theta_3^{CII})/(\theta_2^{CI} - \theta_1^{CI})$$

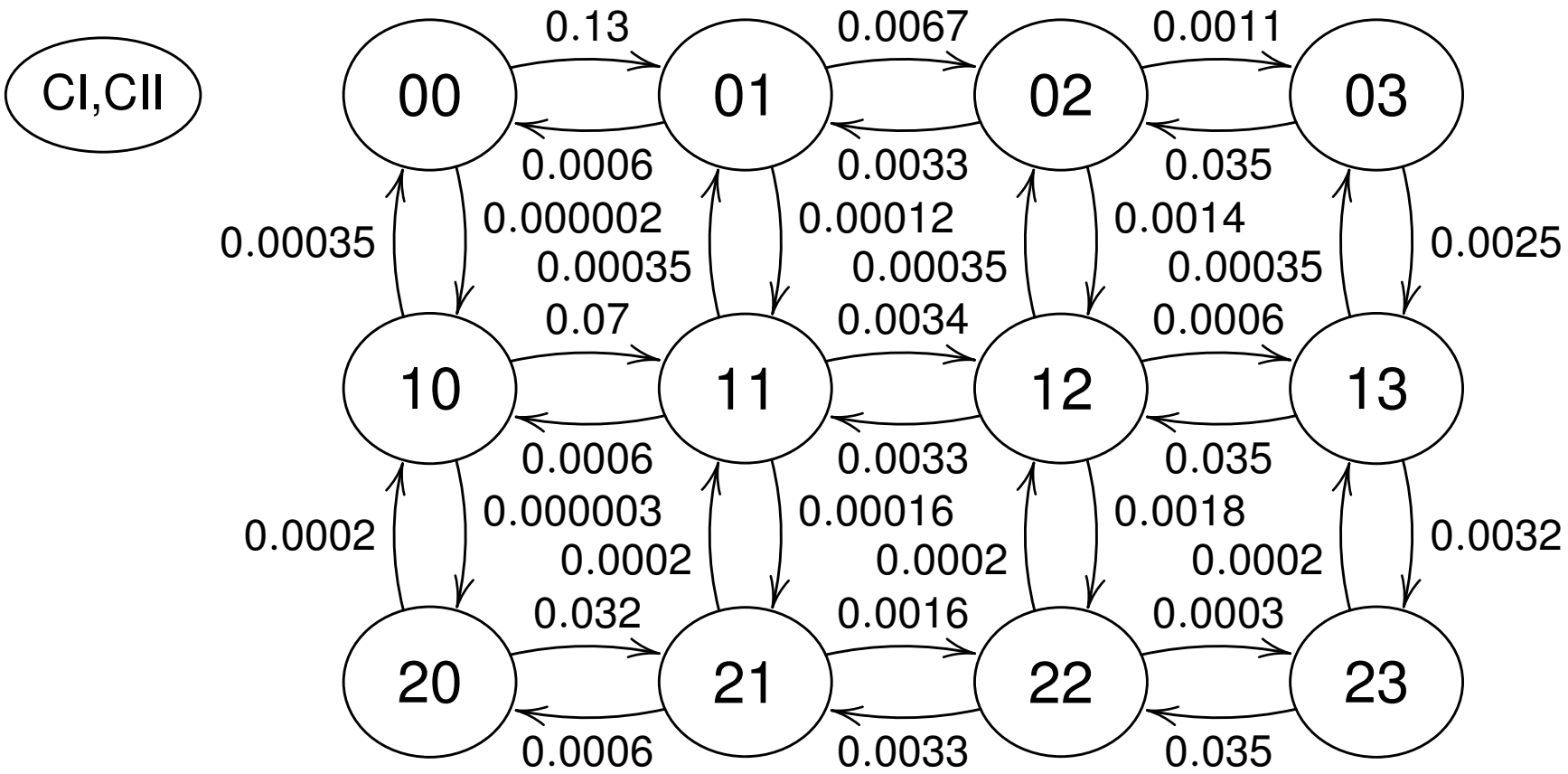
- Degradation of CI:

$$b_{CI} = 2 \xrightarrow{q_9} b_{CI} := 1 \qquad q_9 = f_2(\theta_2^{CI})/(\theta_2^{CI} - \theta_1^{CI})$$
$$b_{CI} = 1 \xrightarrow{q_{10}} b_{CI} := 0 \qquad q_{10} = f_2(\theta_1^{CI})/\theta_1^{CI}$$

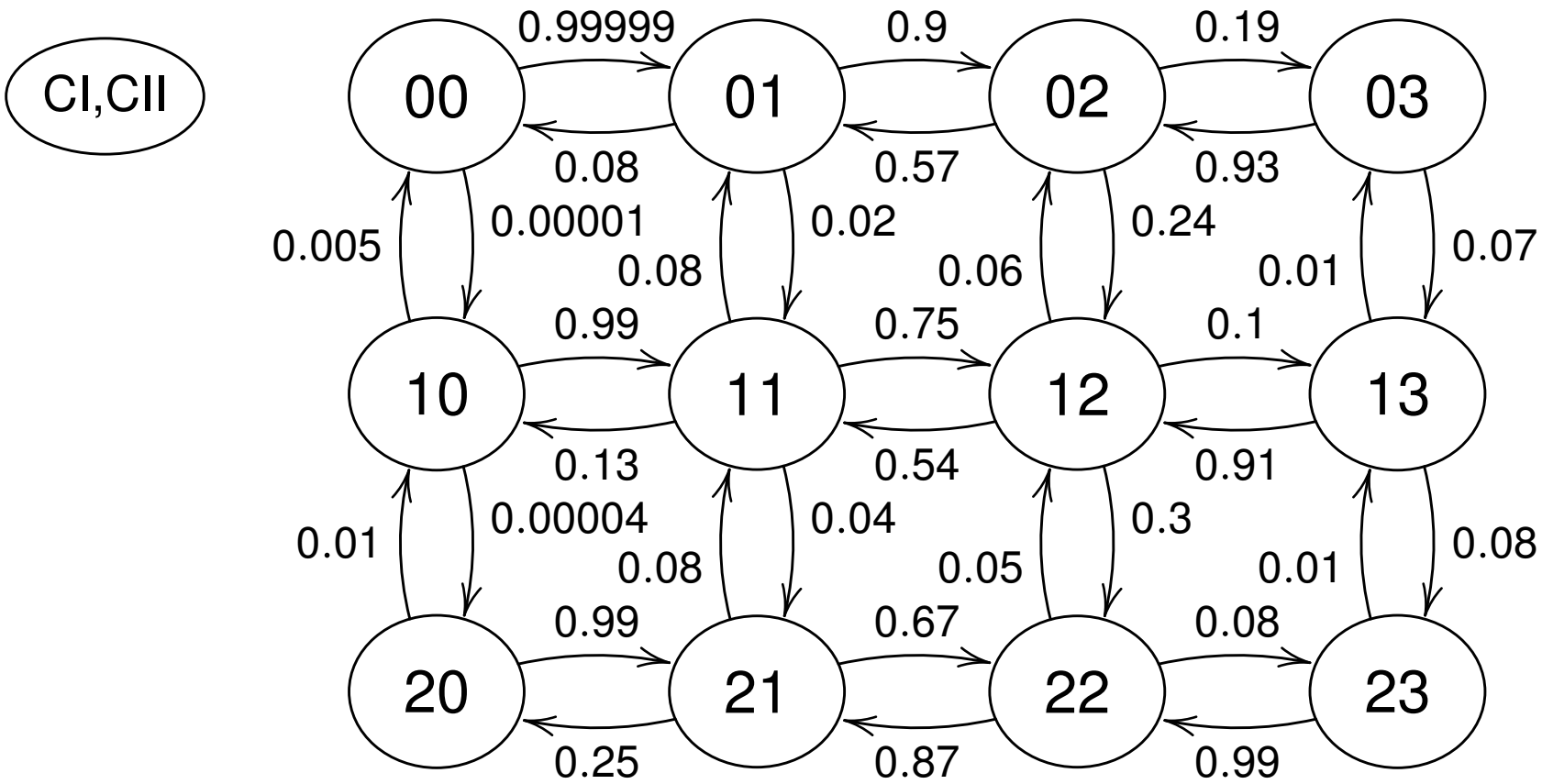
Reachable State Space



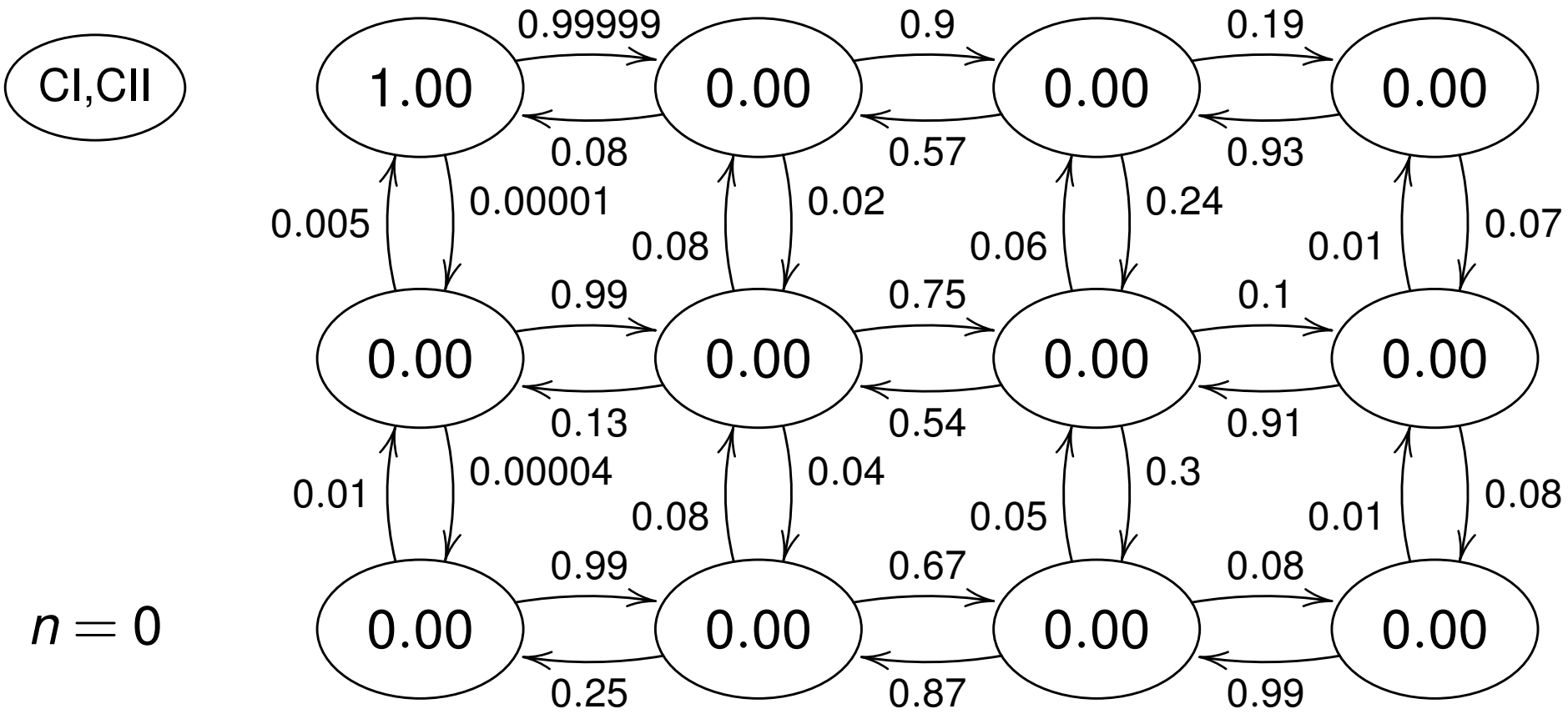
CTMC Transition Rates



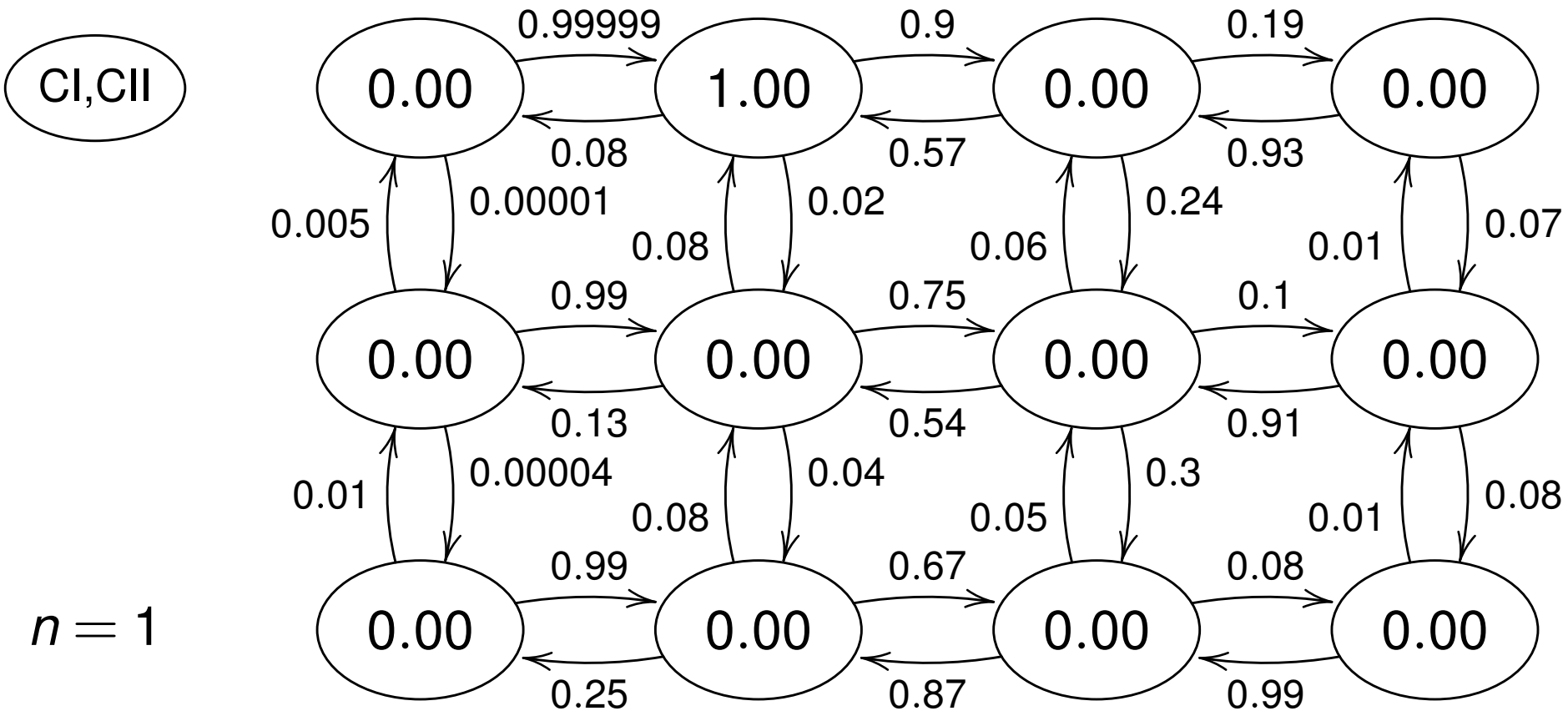
EMC Transition Probabilities



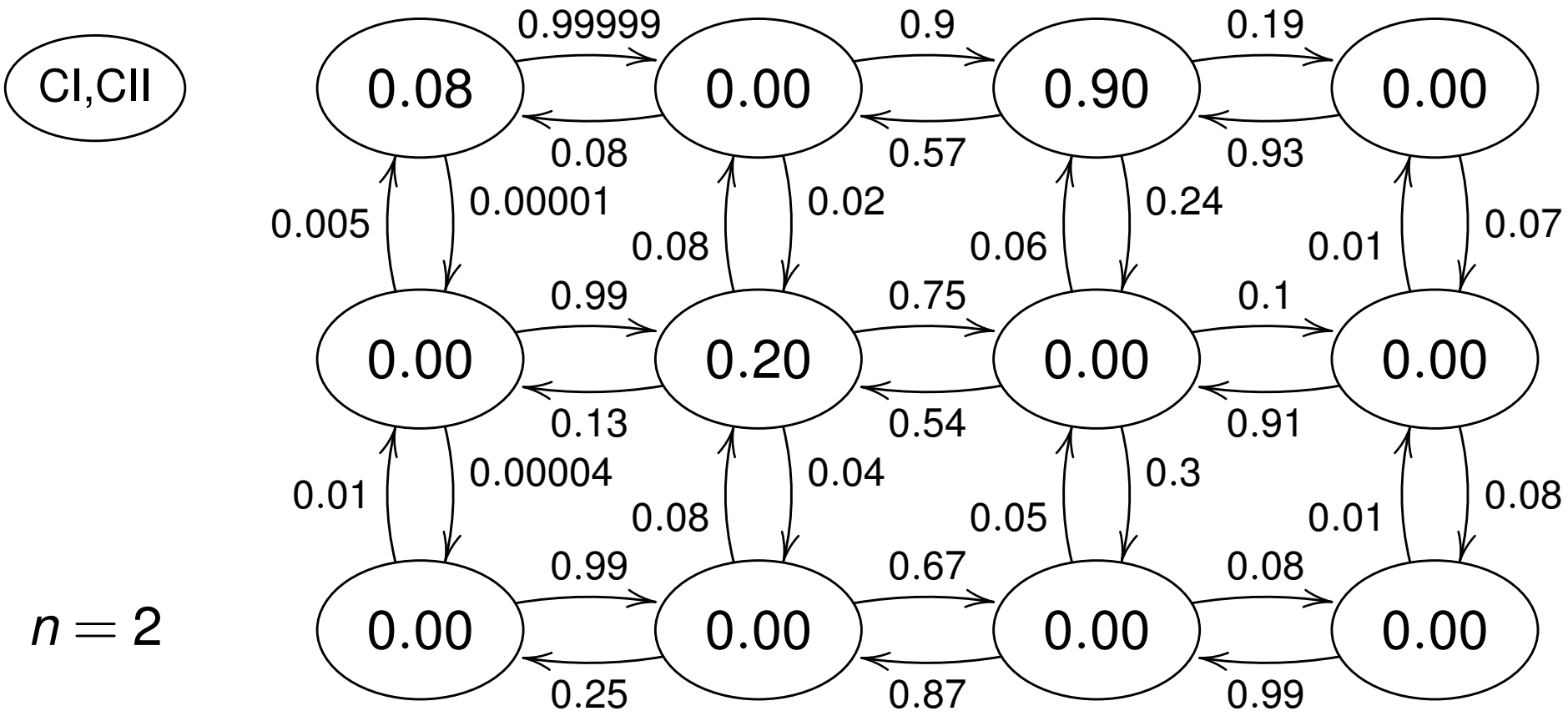
Initial State Probabilities



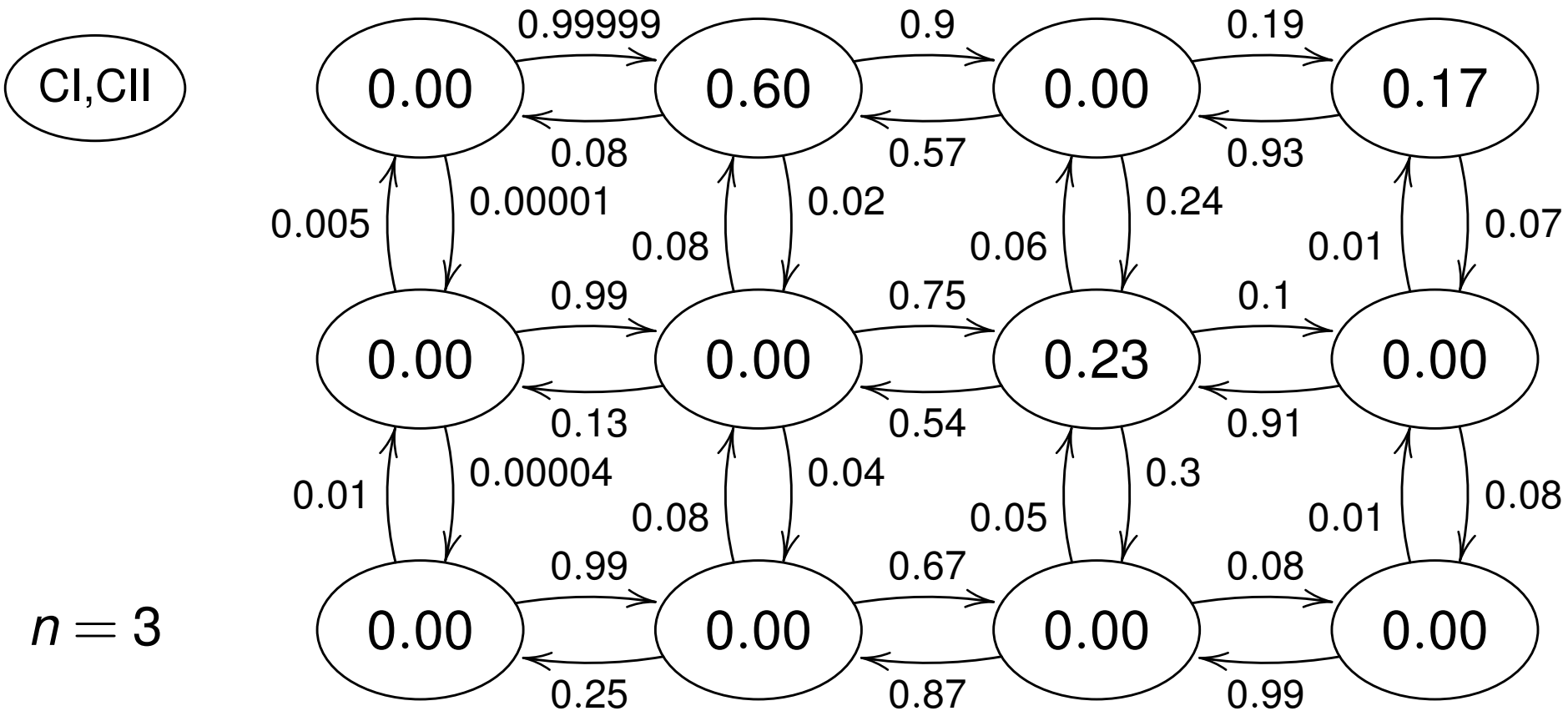
State Probabilities



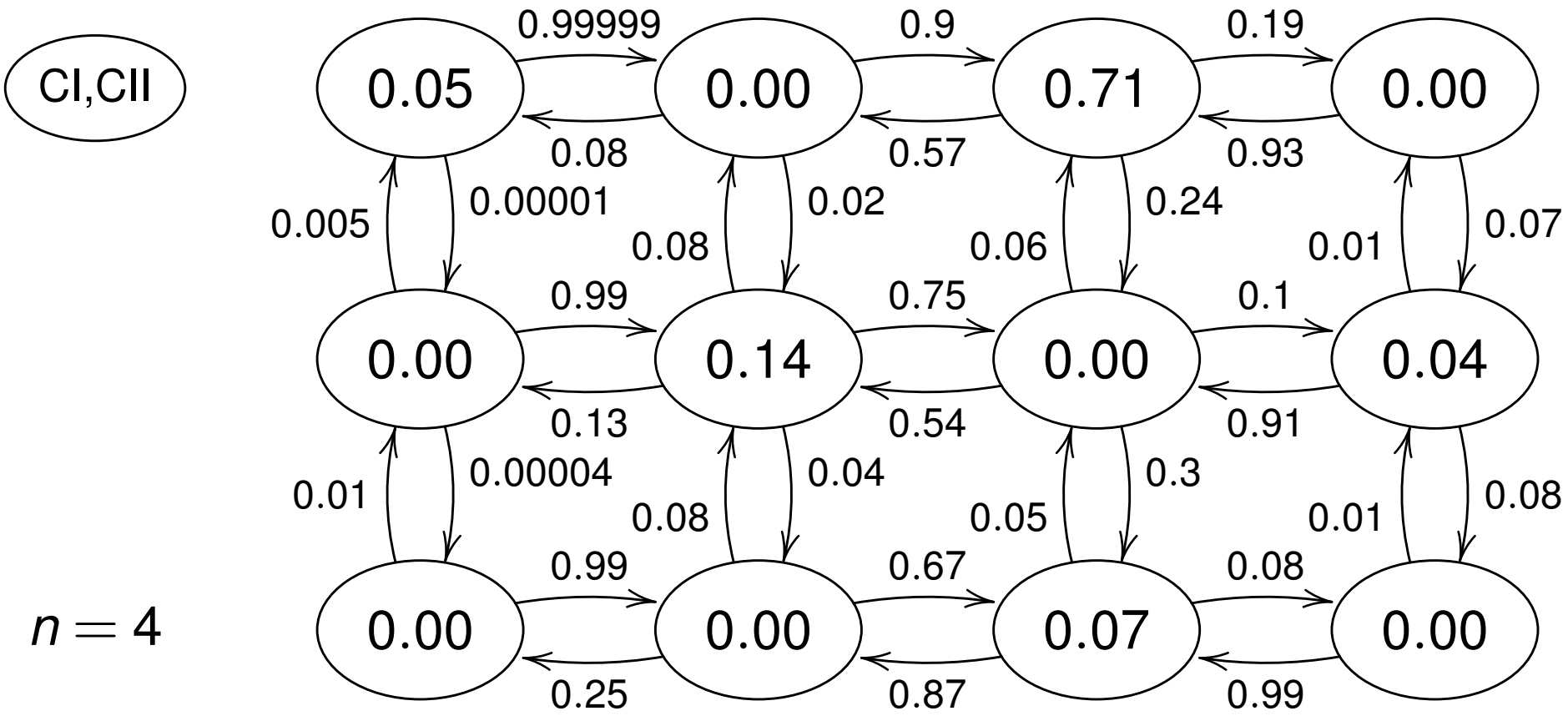
State Probabilities



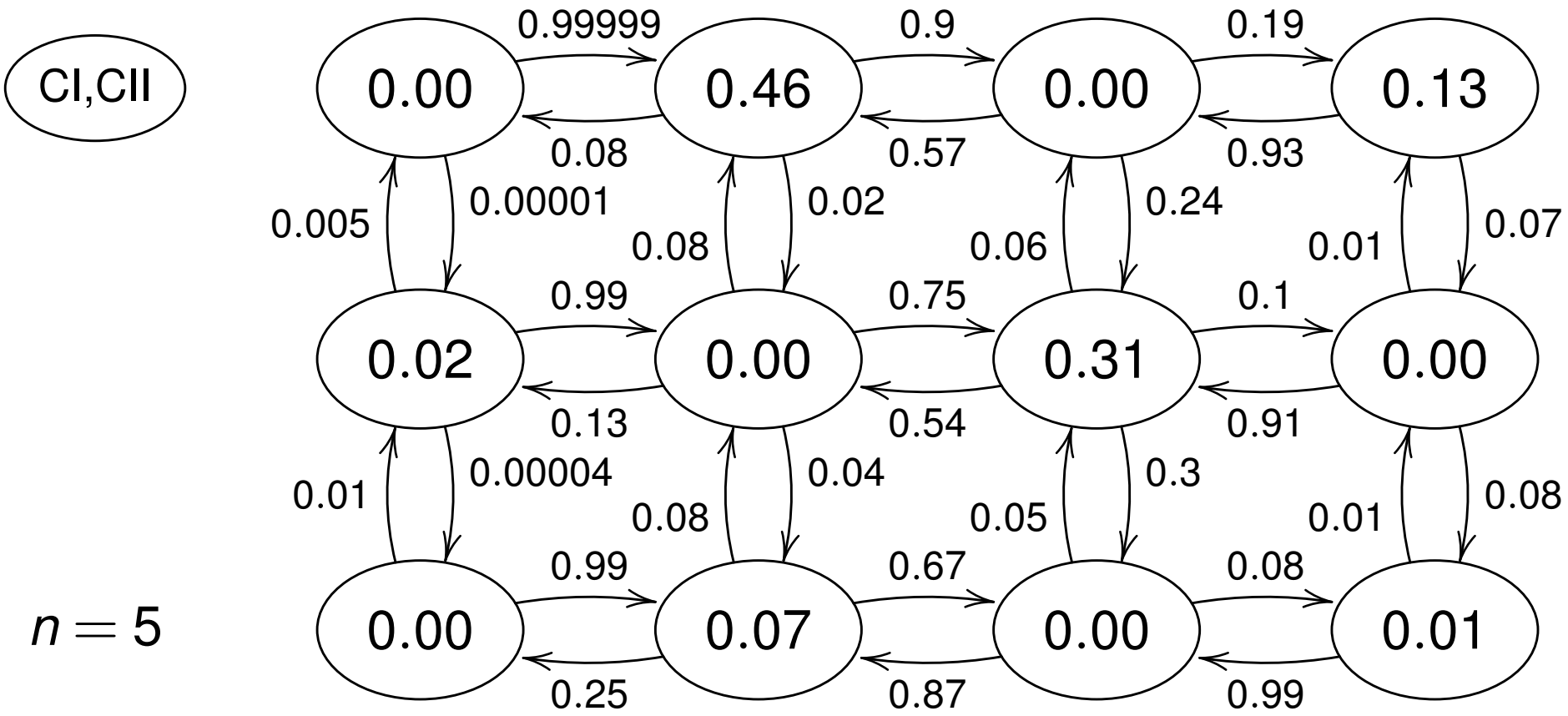
State Probabilities



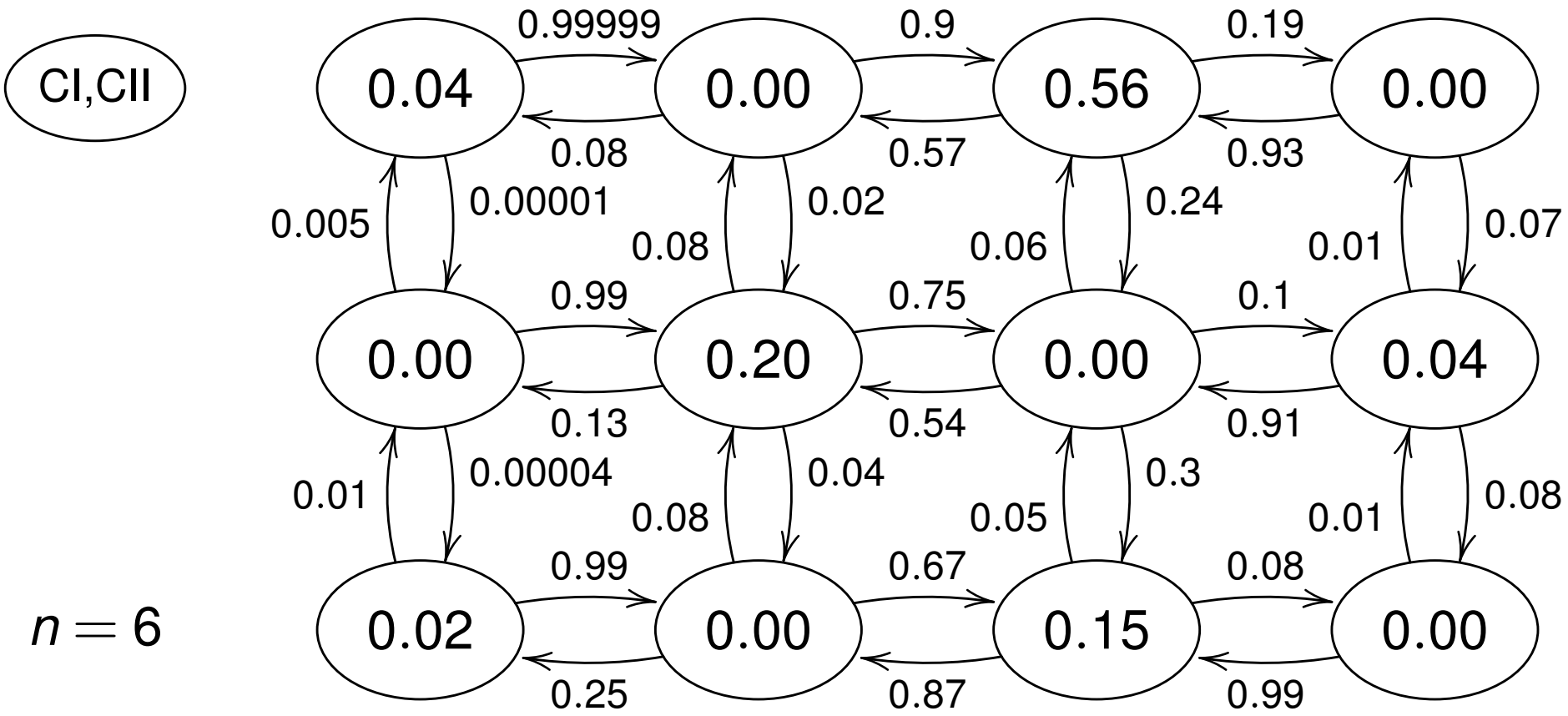
State Probabilities



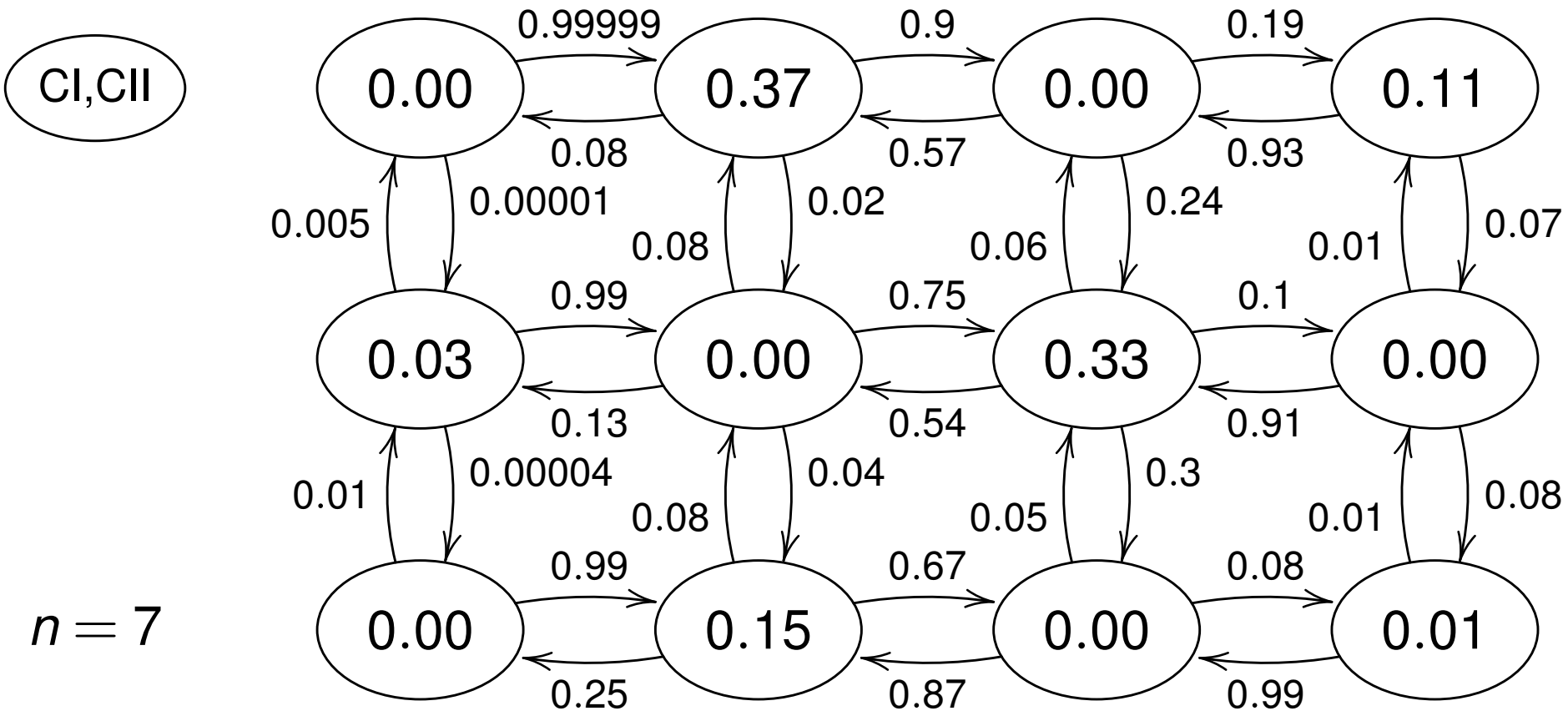
State Probabilities



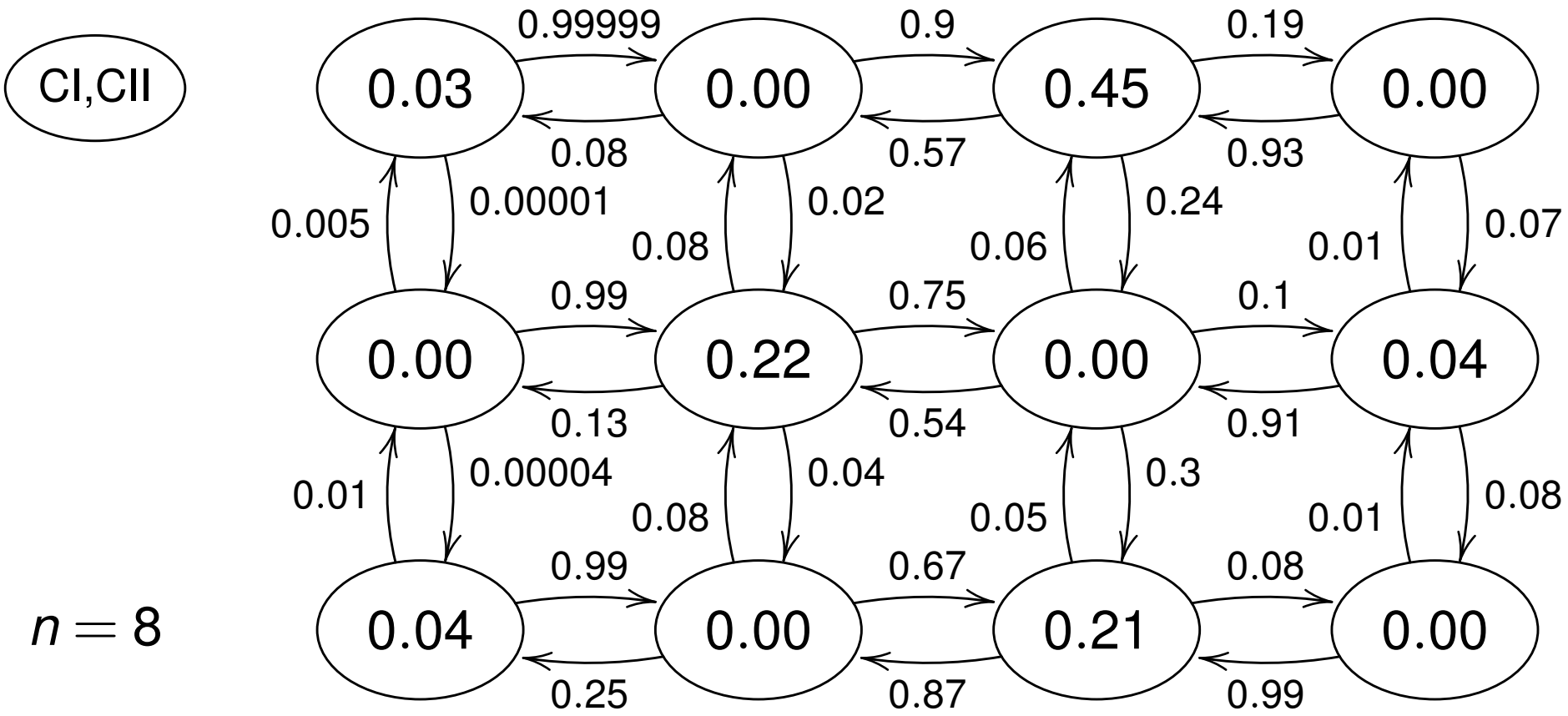
State Probabilities



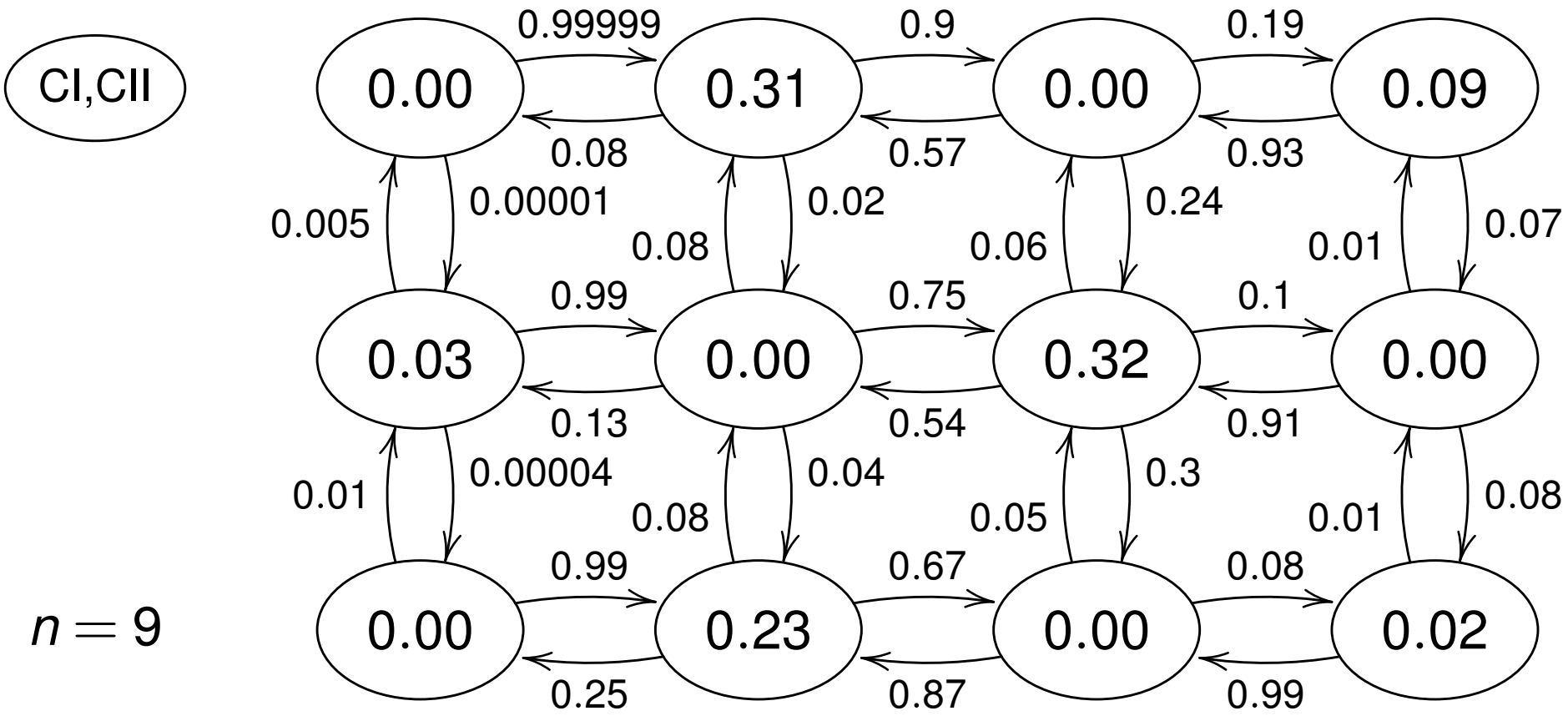
State Probabilities



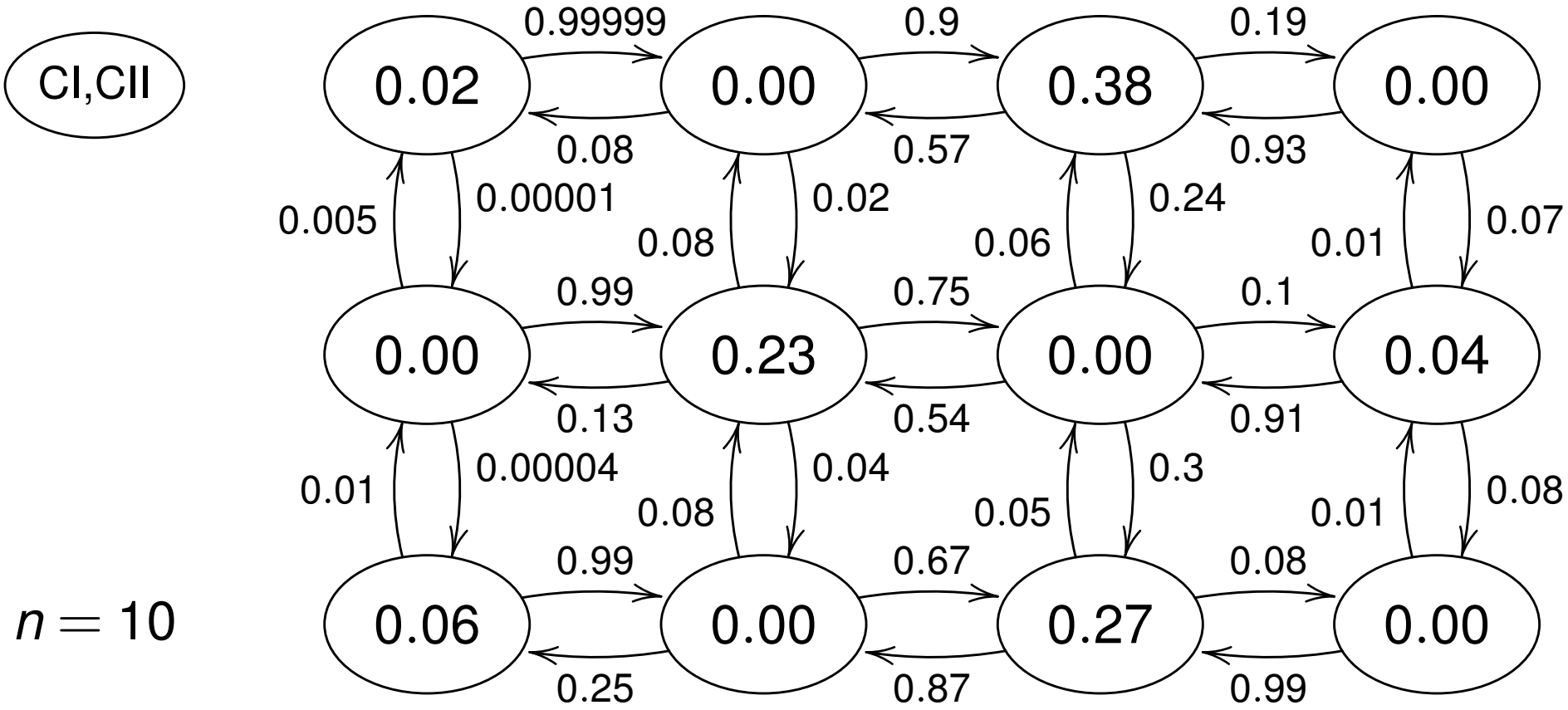
State Probabilities



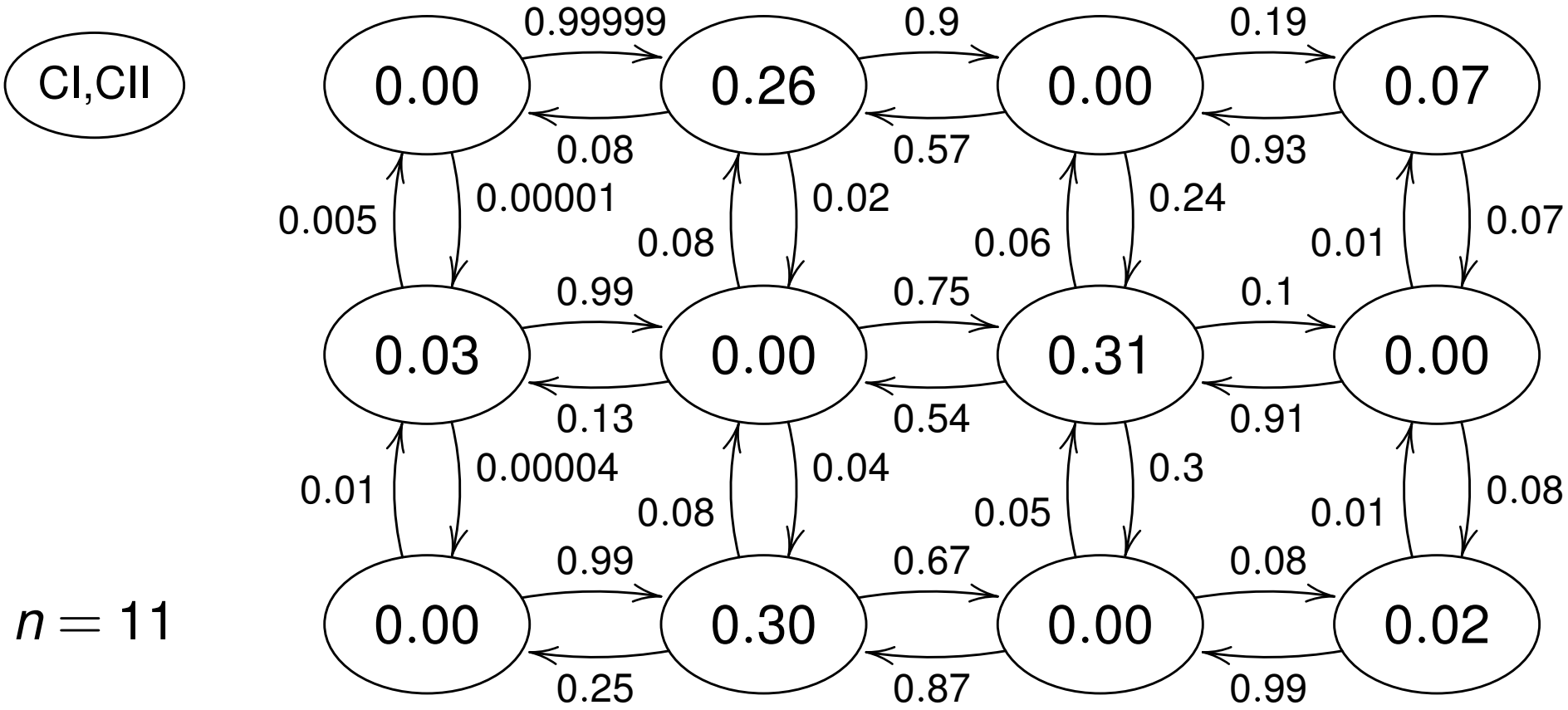
State Probabilities



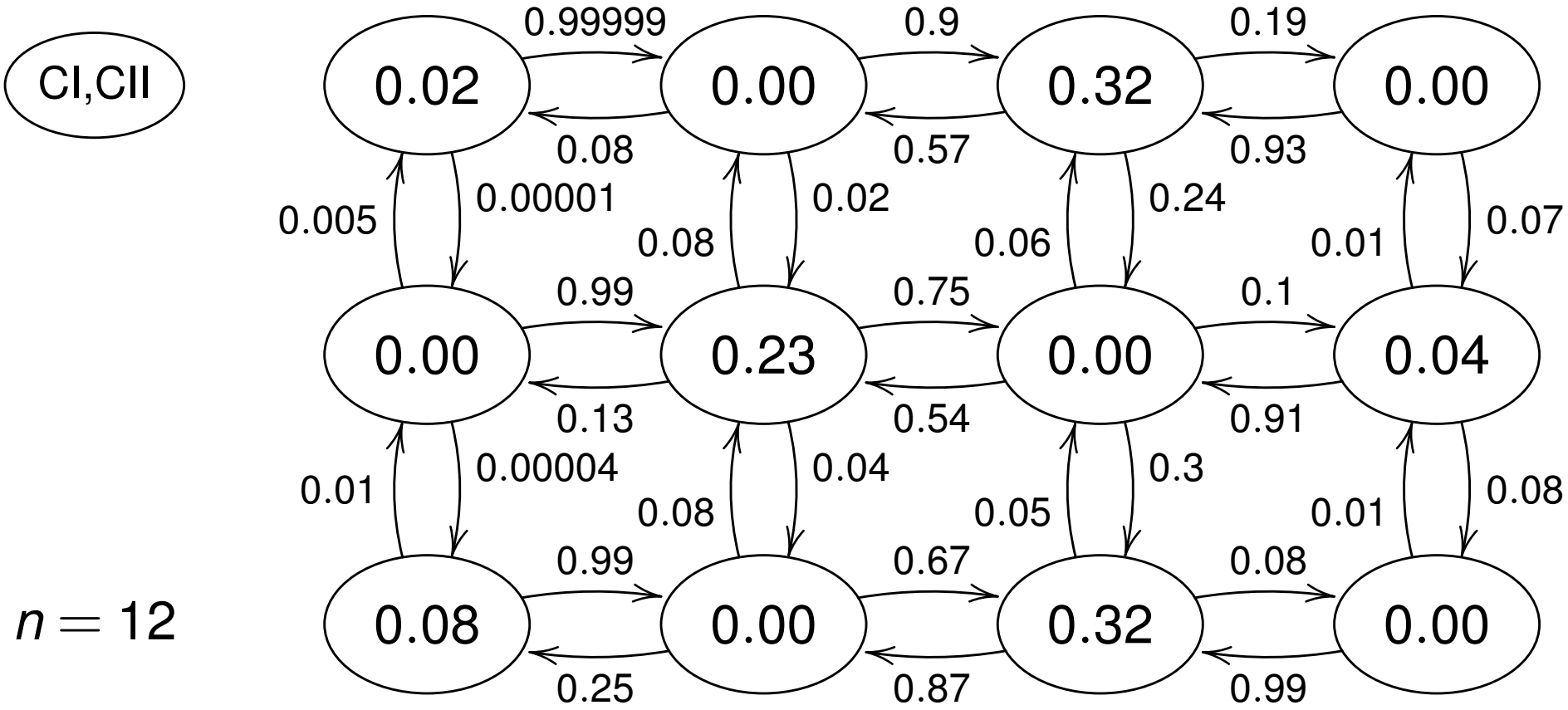
State Probabilities



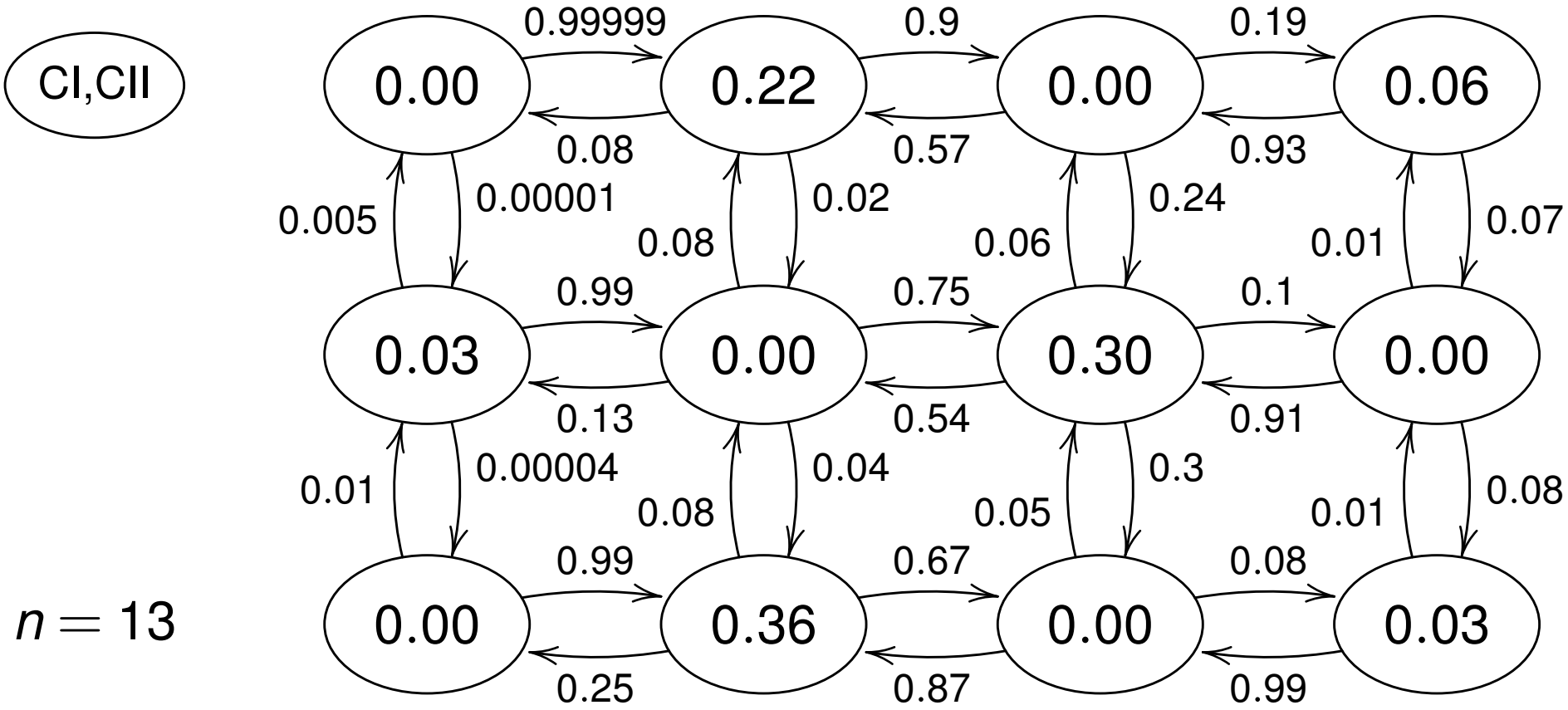
State Probabilities



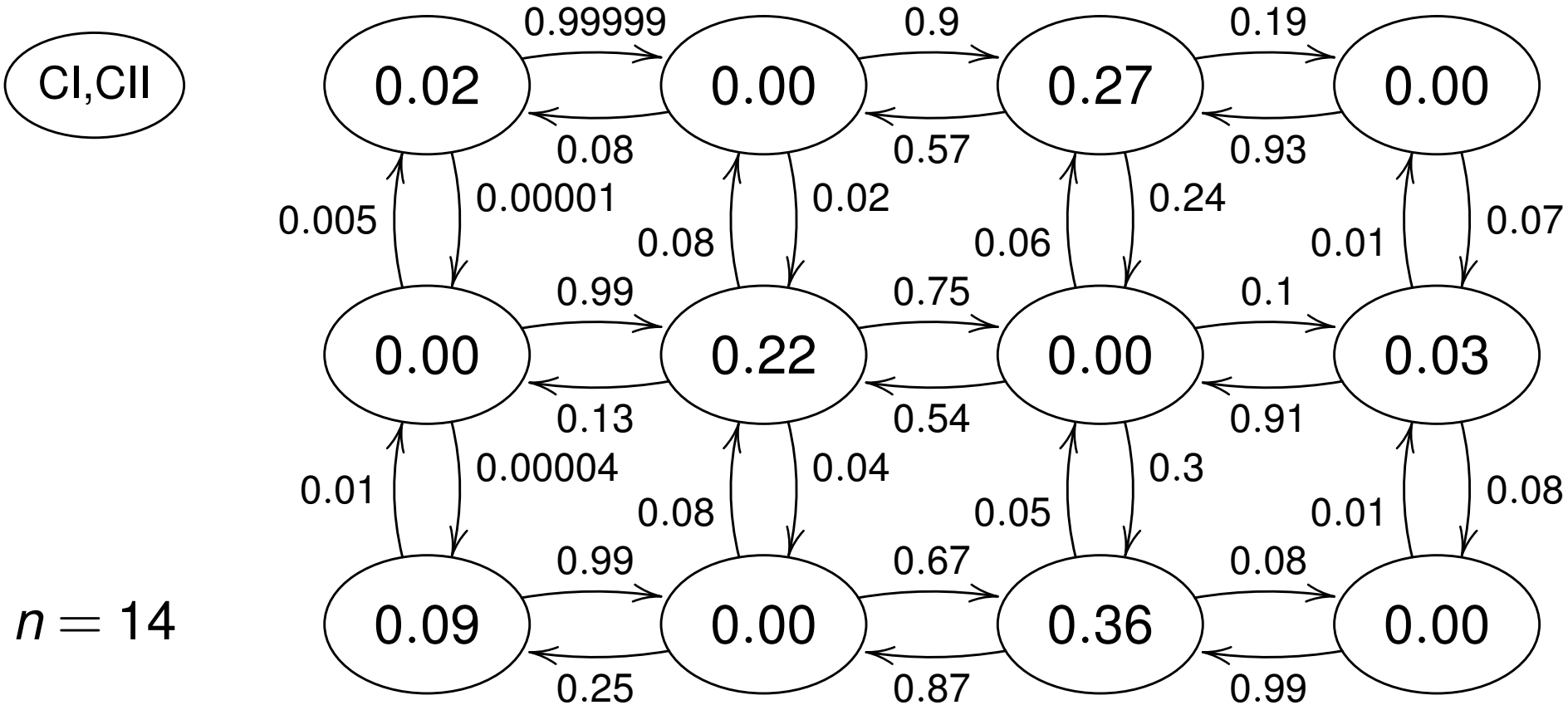
State Probabilities



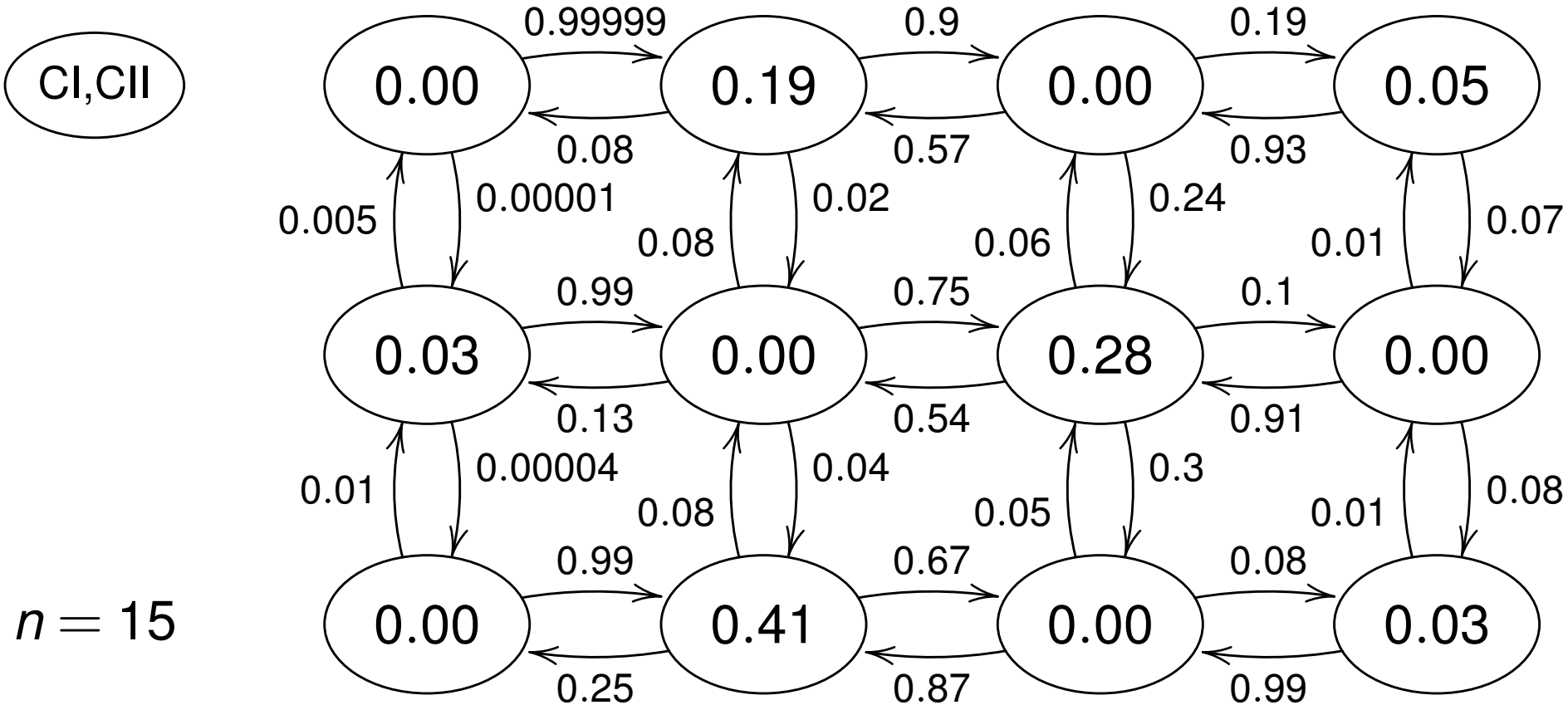
State Probabilities



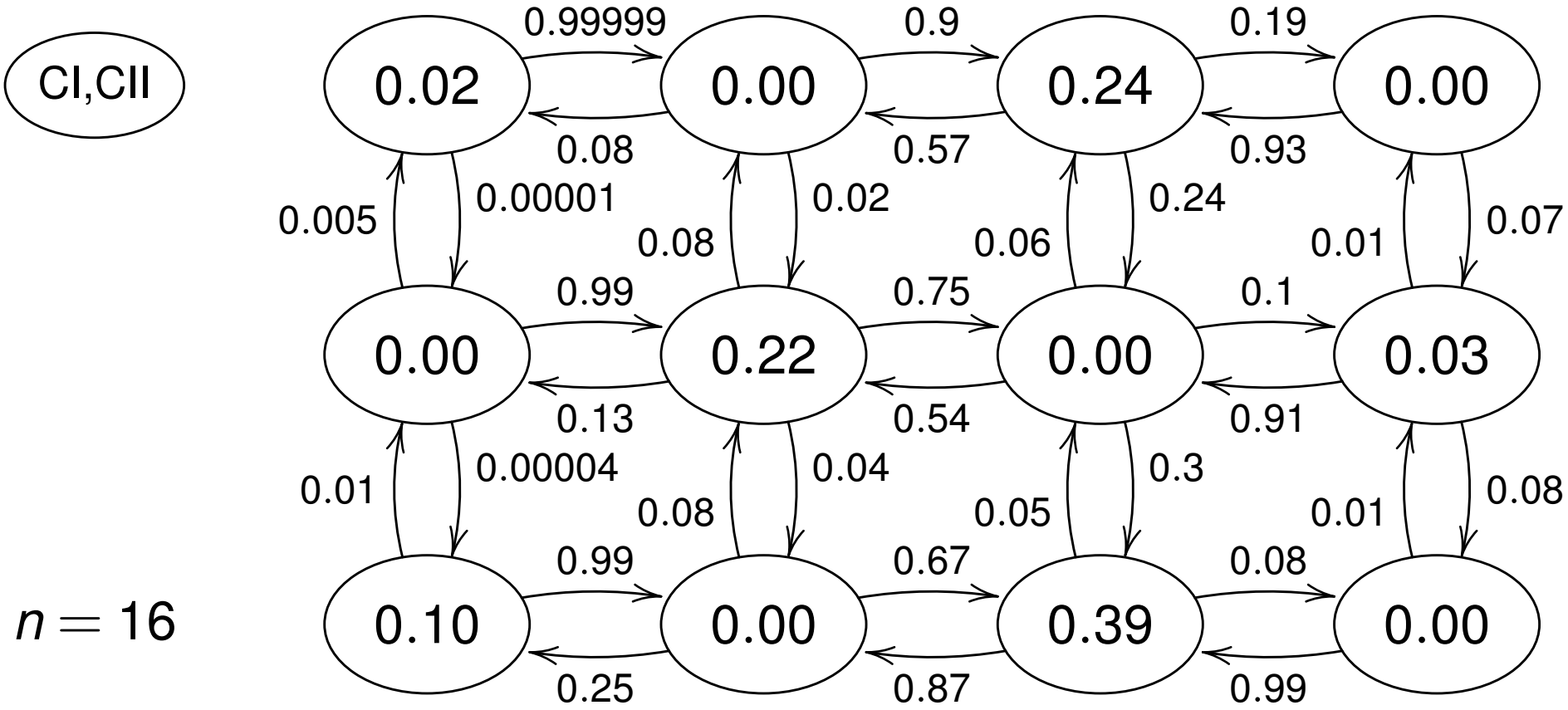
State Probabilities



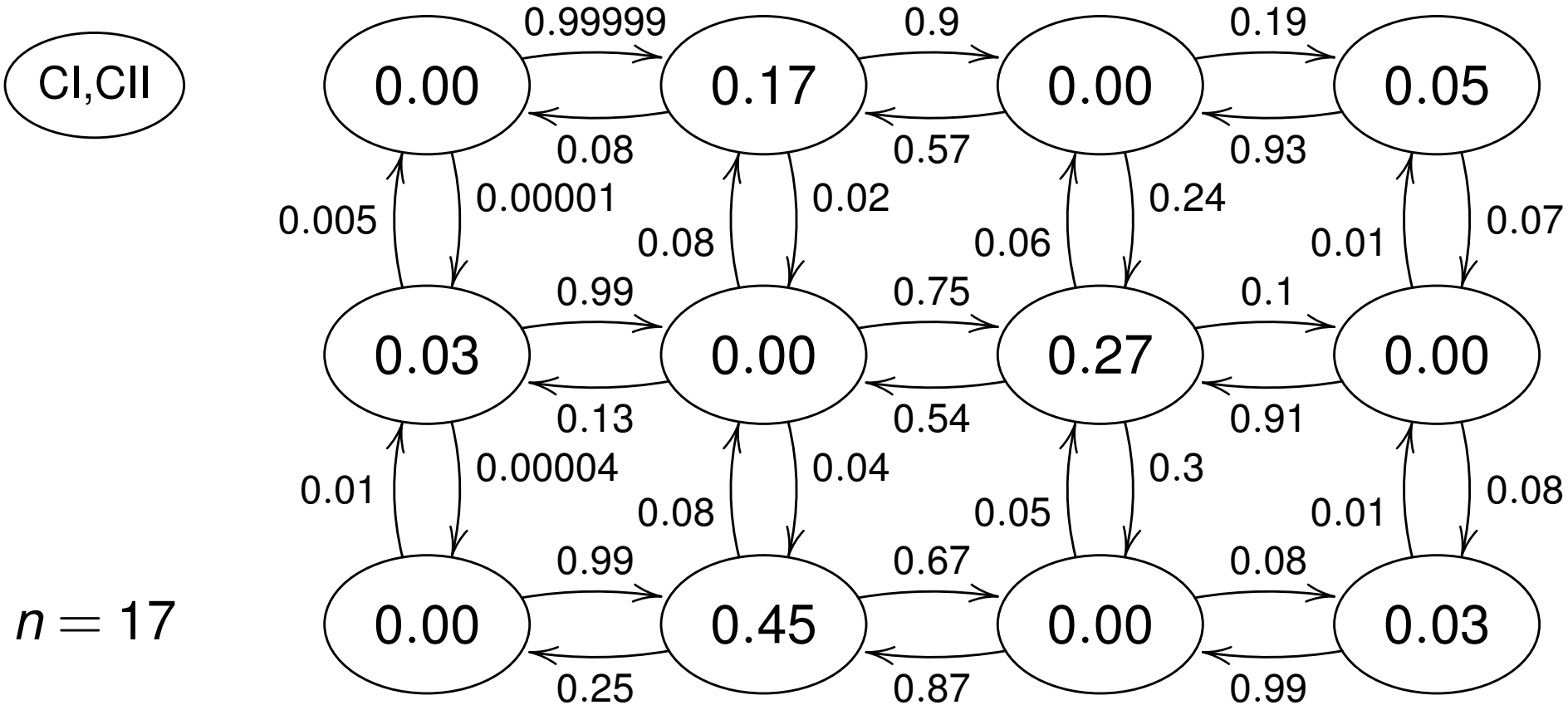
State Probabilities



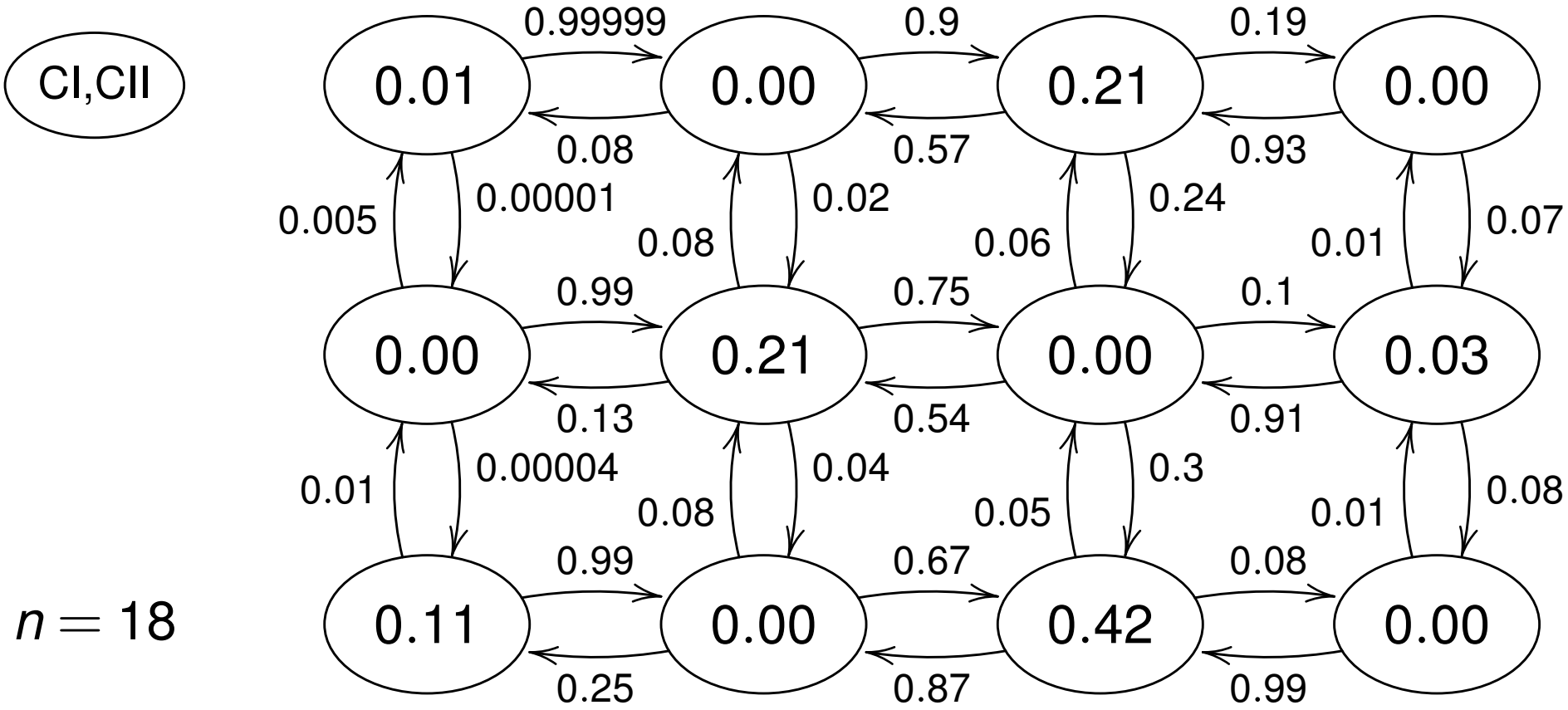
State Probabilities



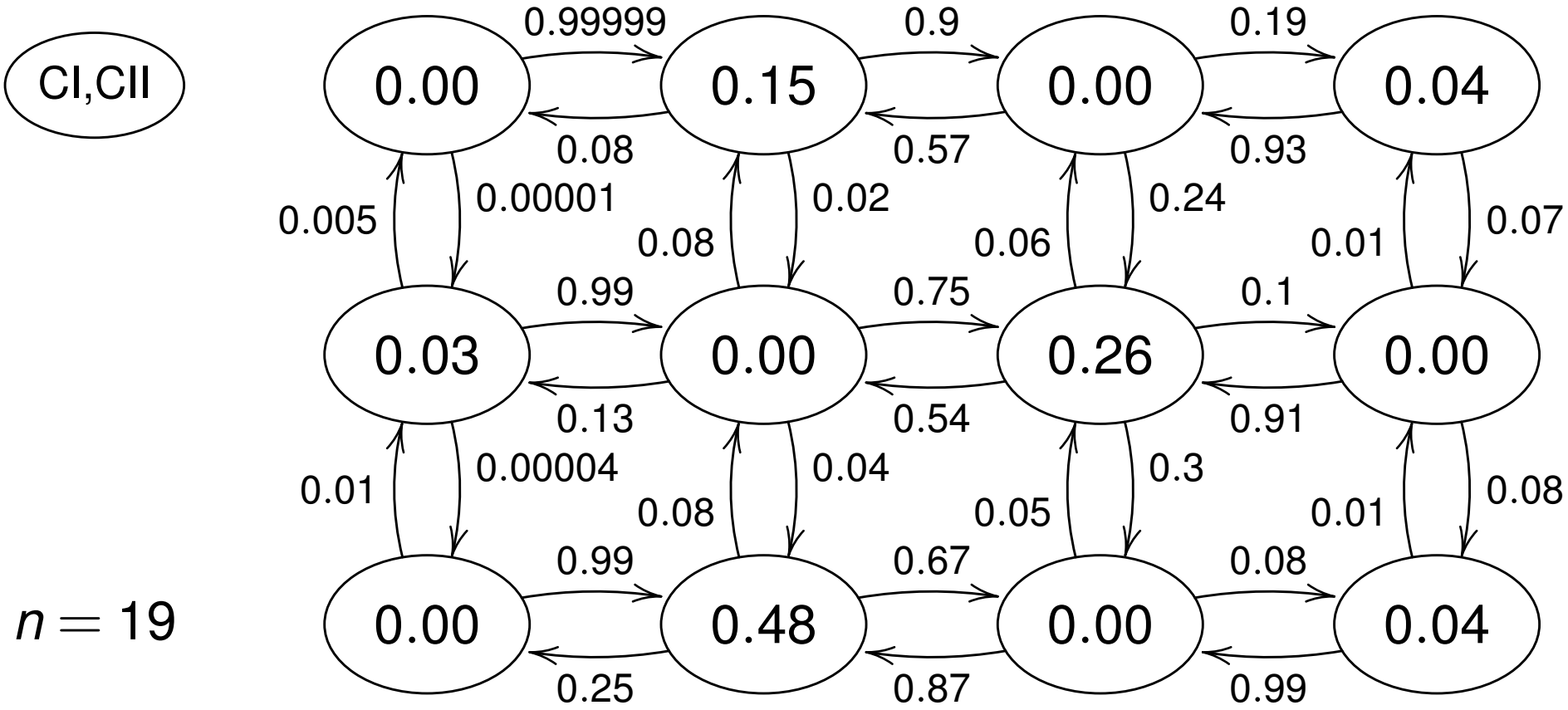
State Probabilities



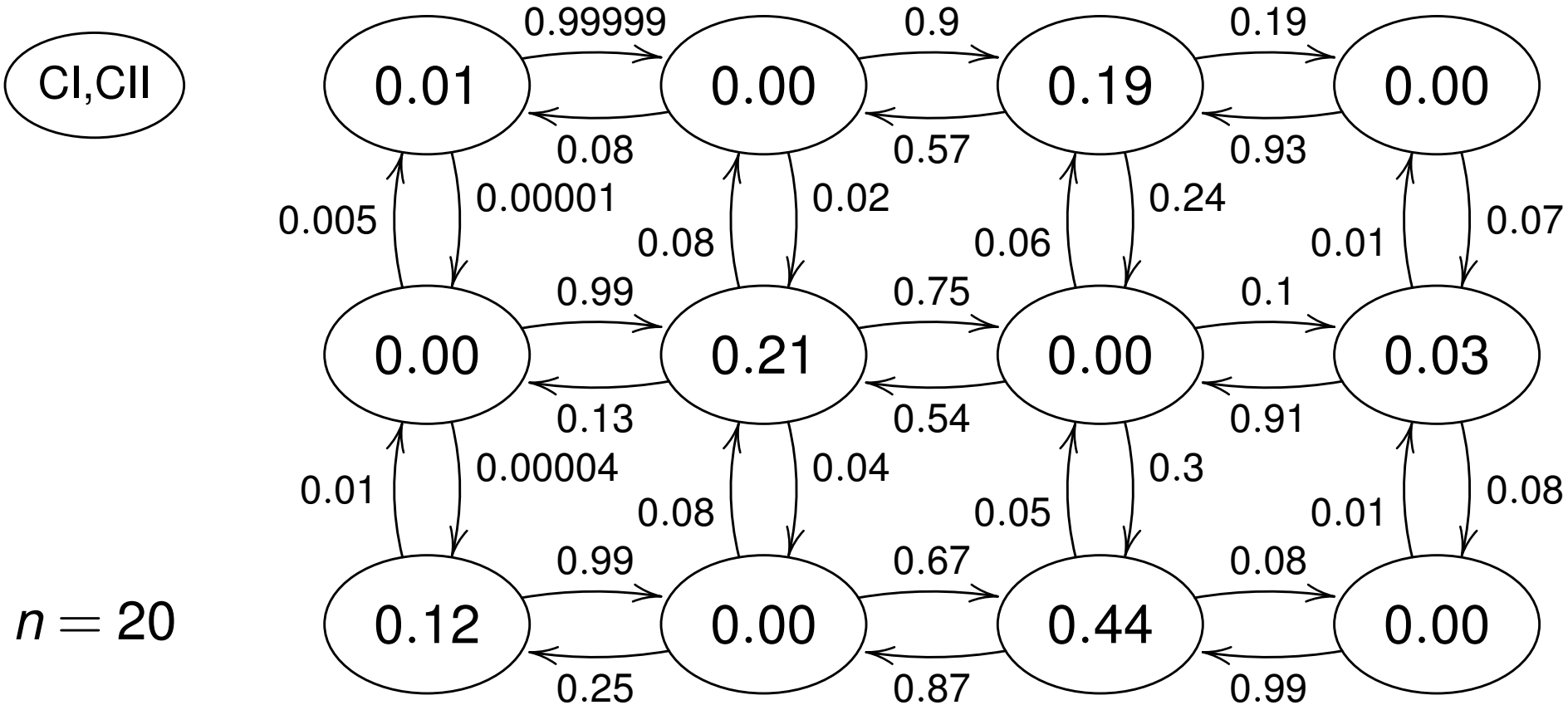
State Probabilities



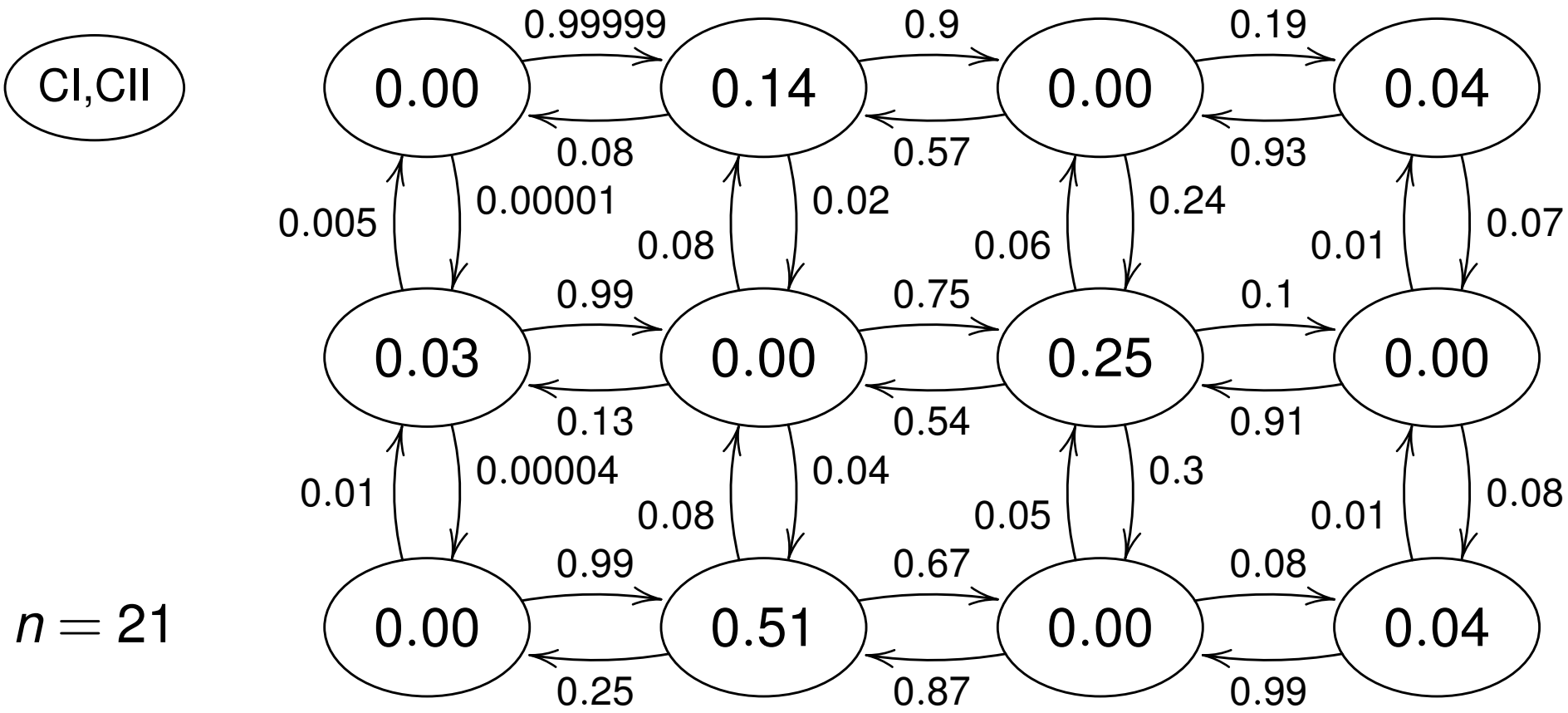
State Probabilities



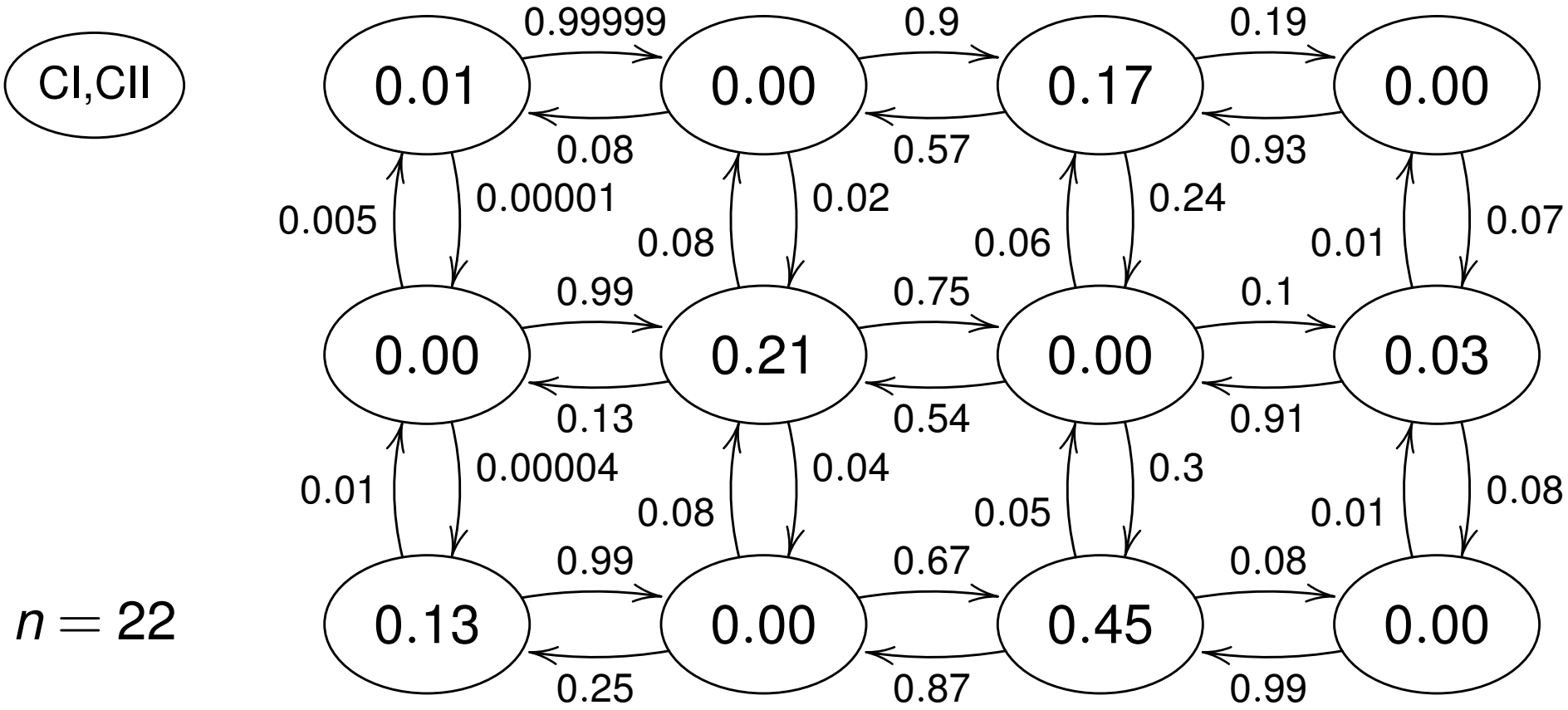
State Probabilities



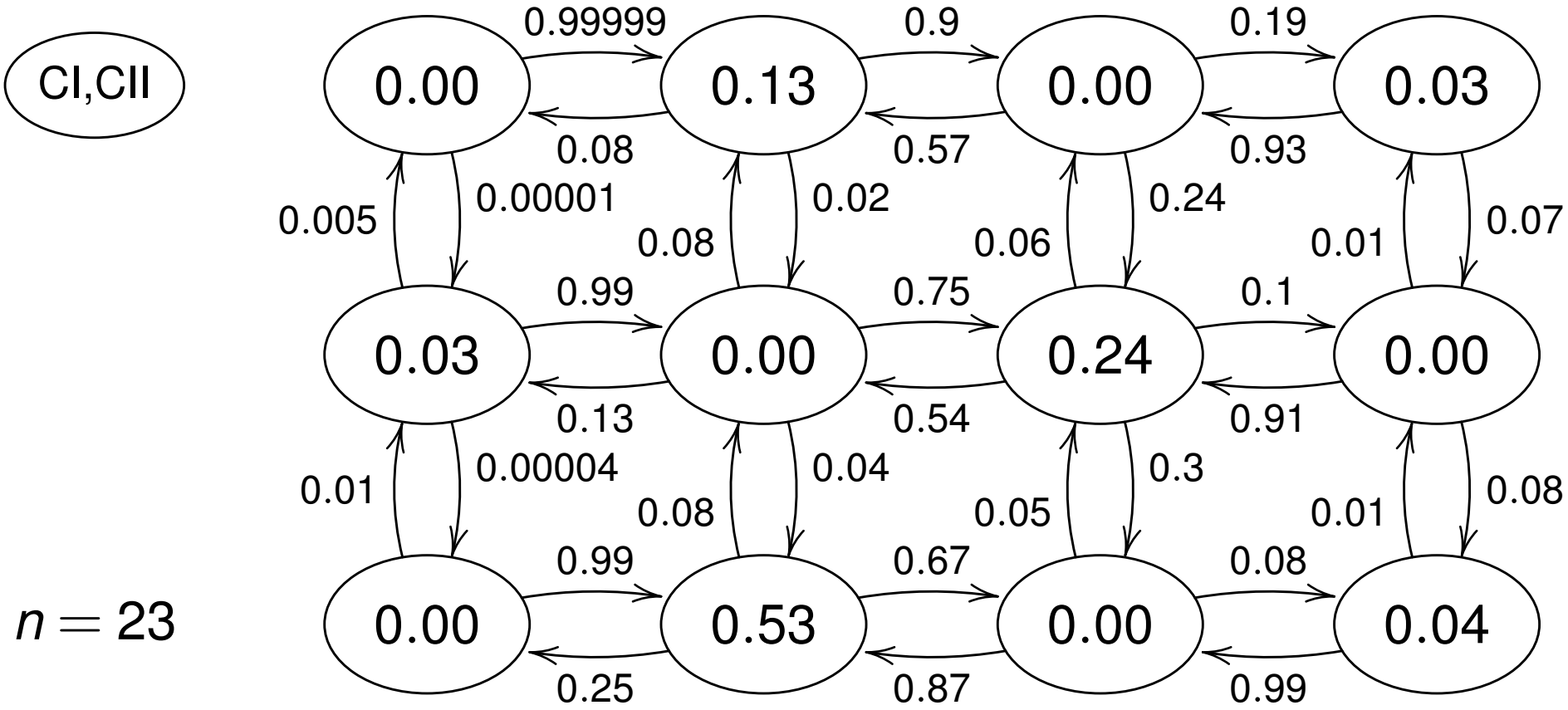
State Probabilities



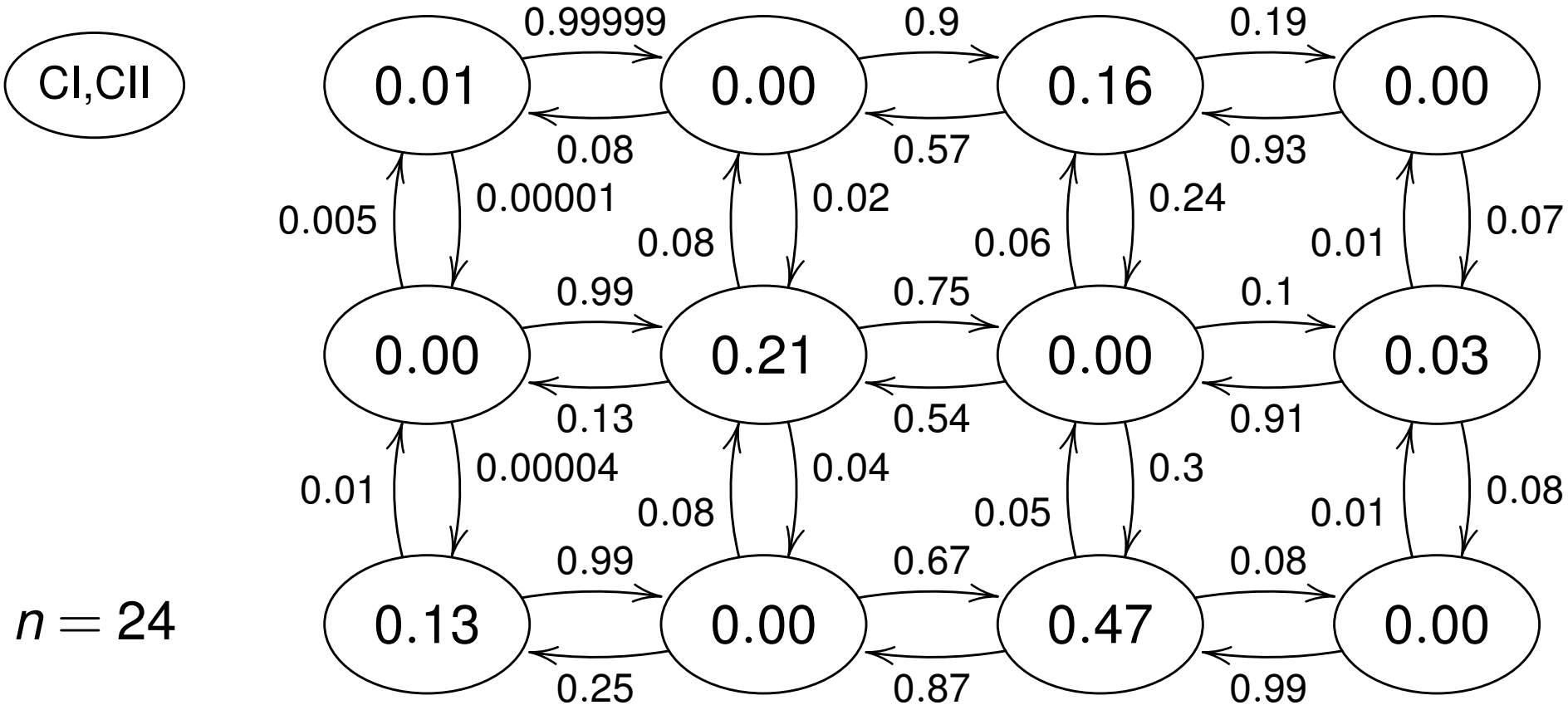
State Probabilities



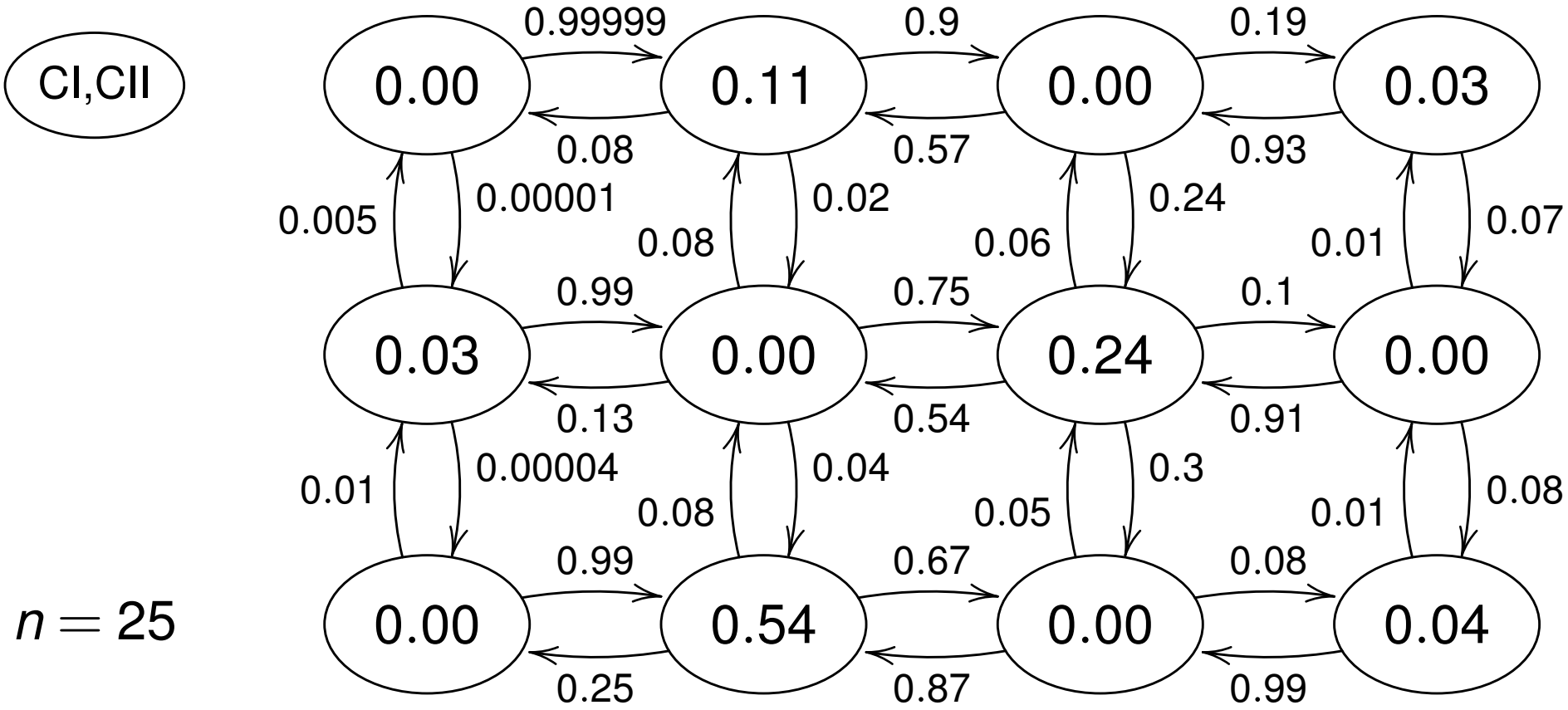
State Probabilities



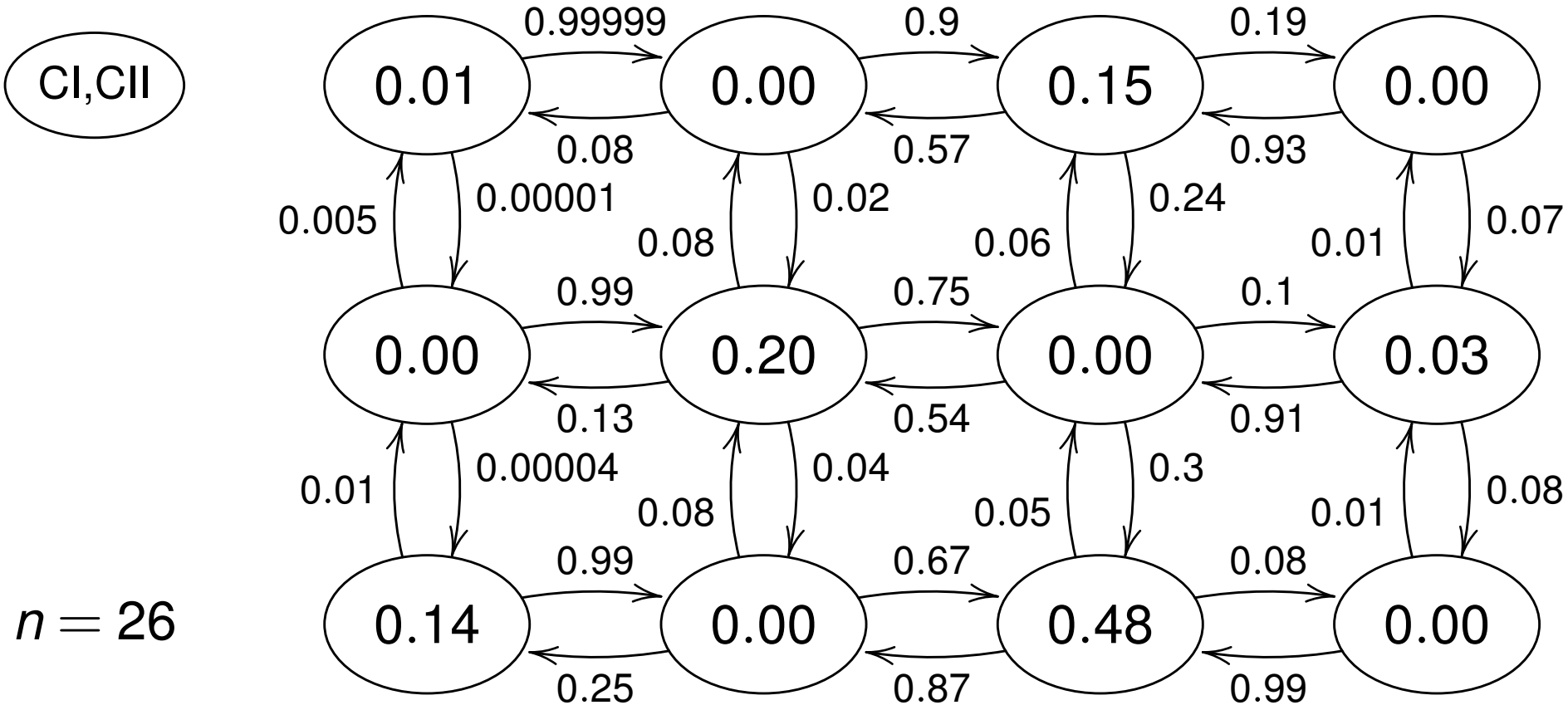
State Probabilities



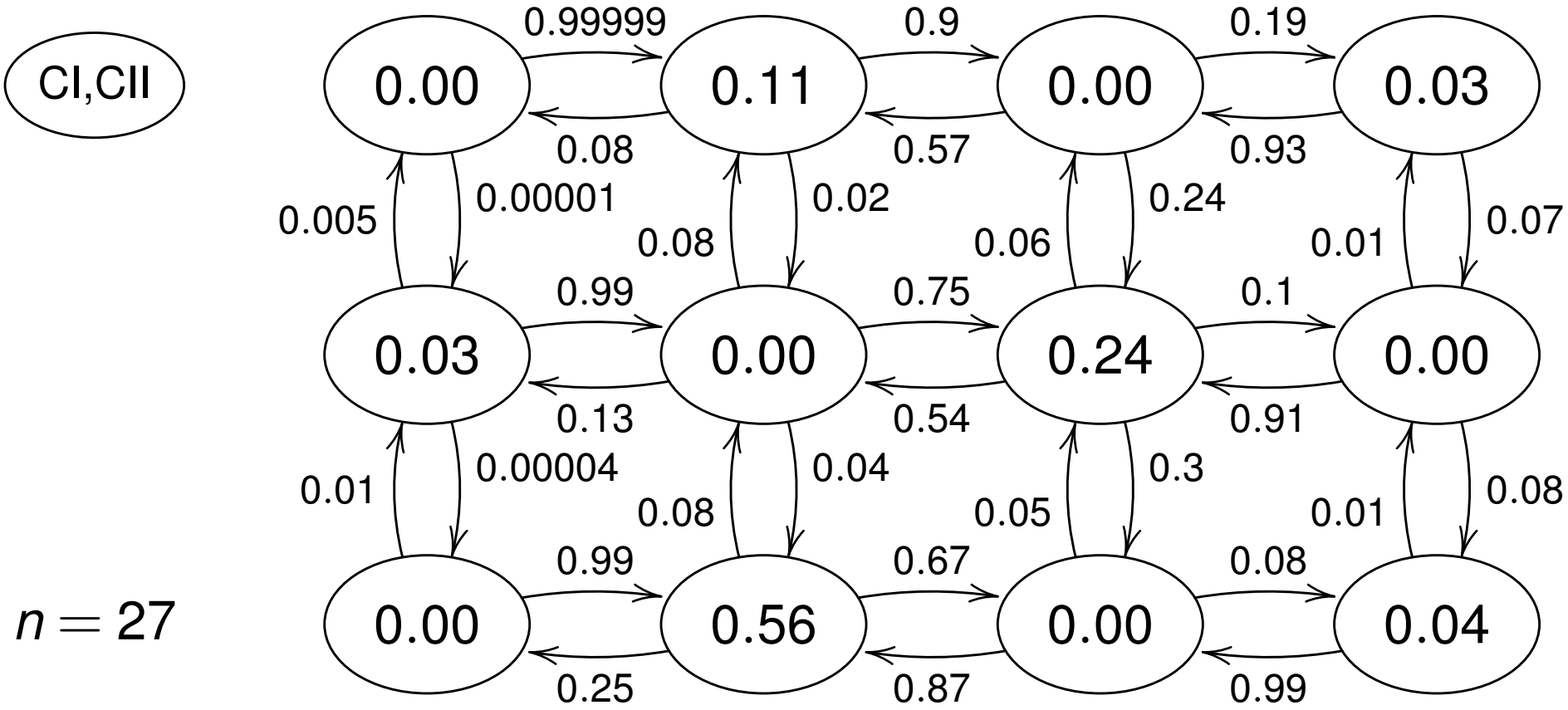
State Probabilities



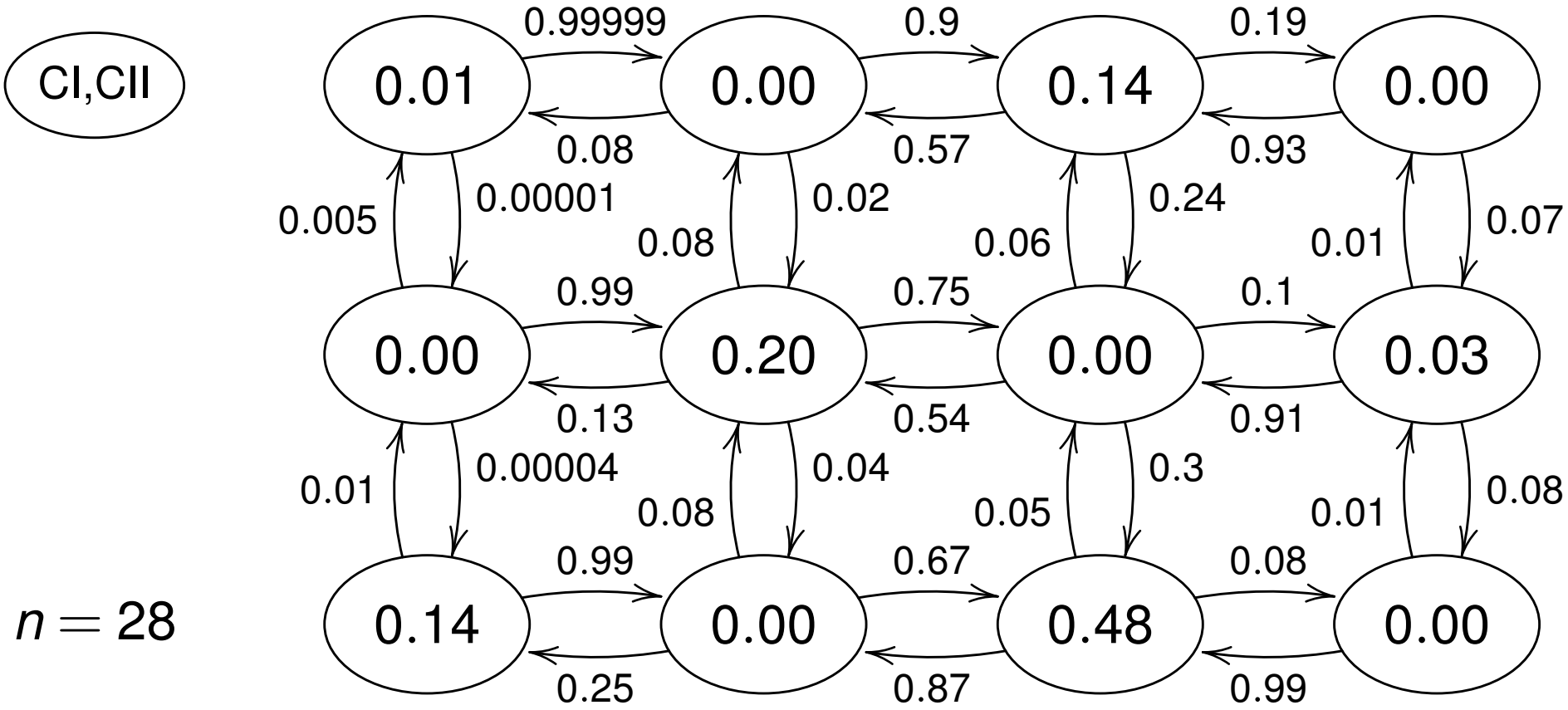
State Probabilities



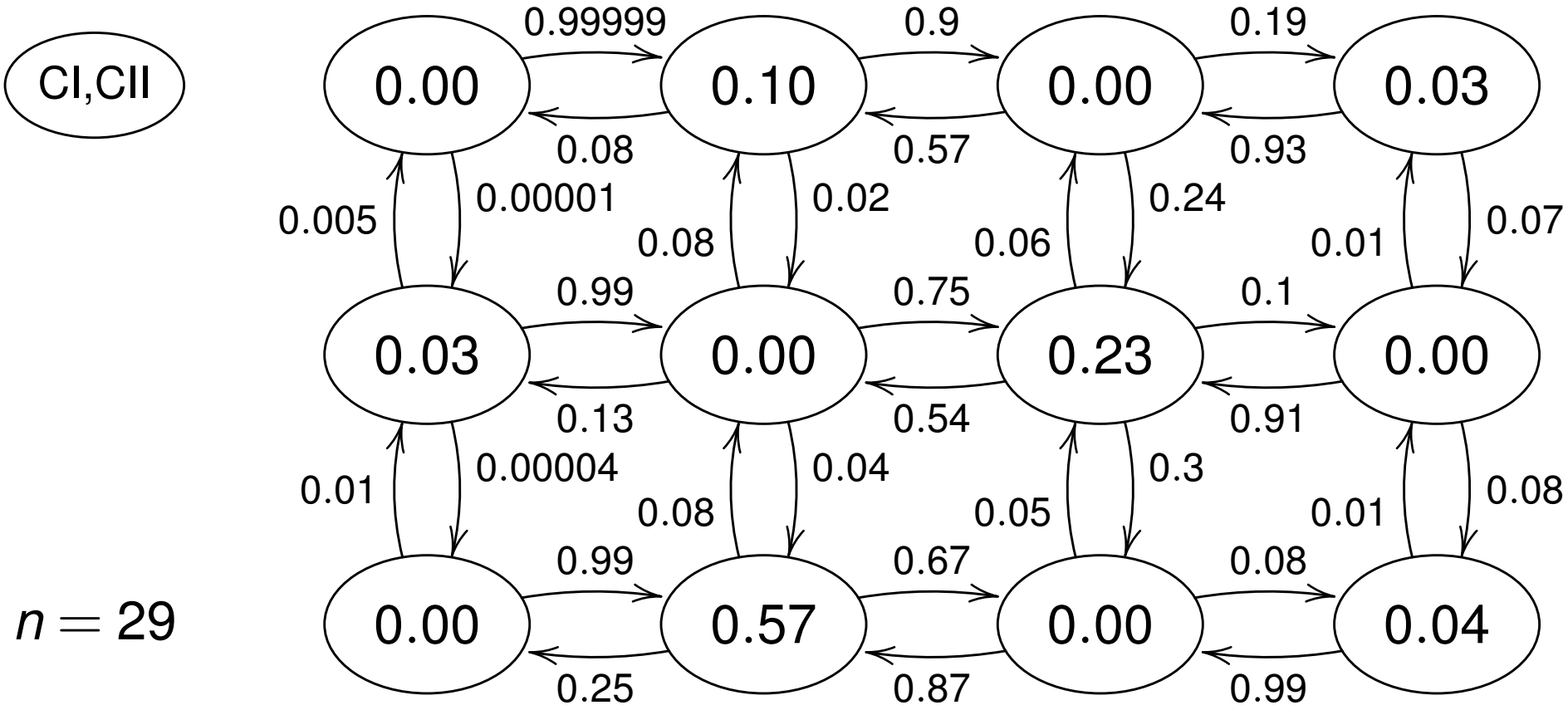
State Probabilities



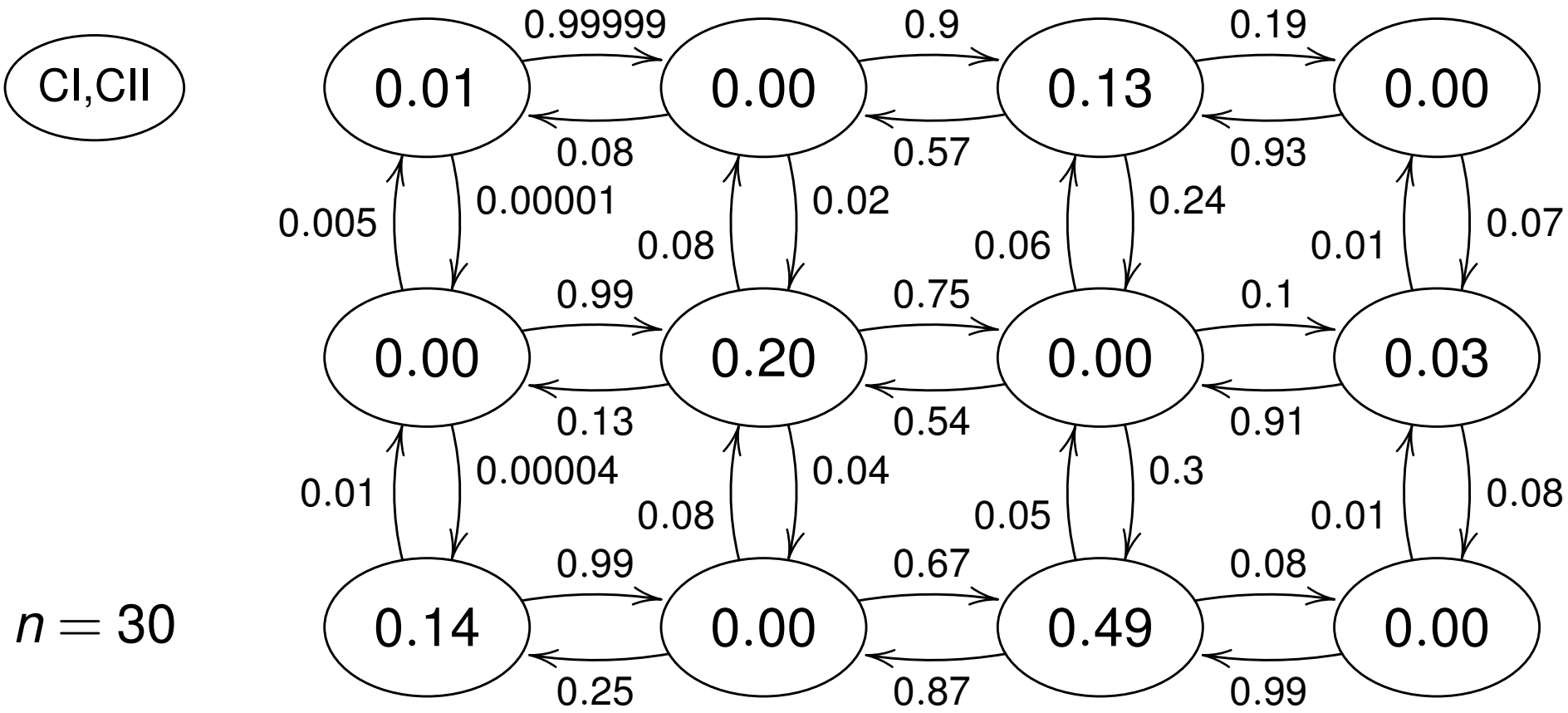
State Probabilities



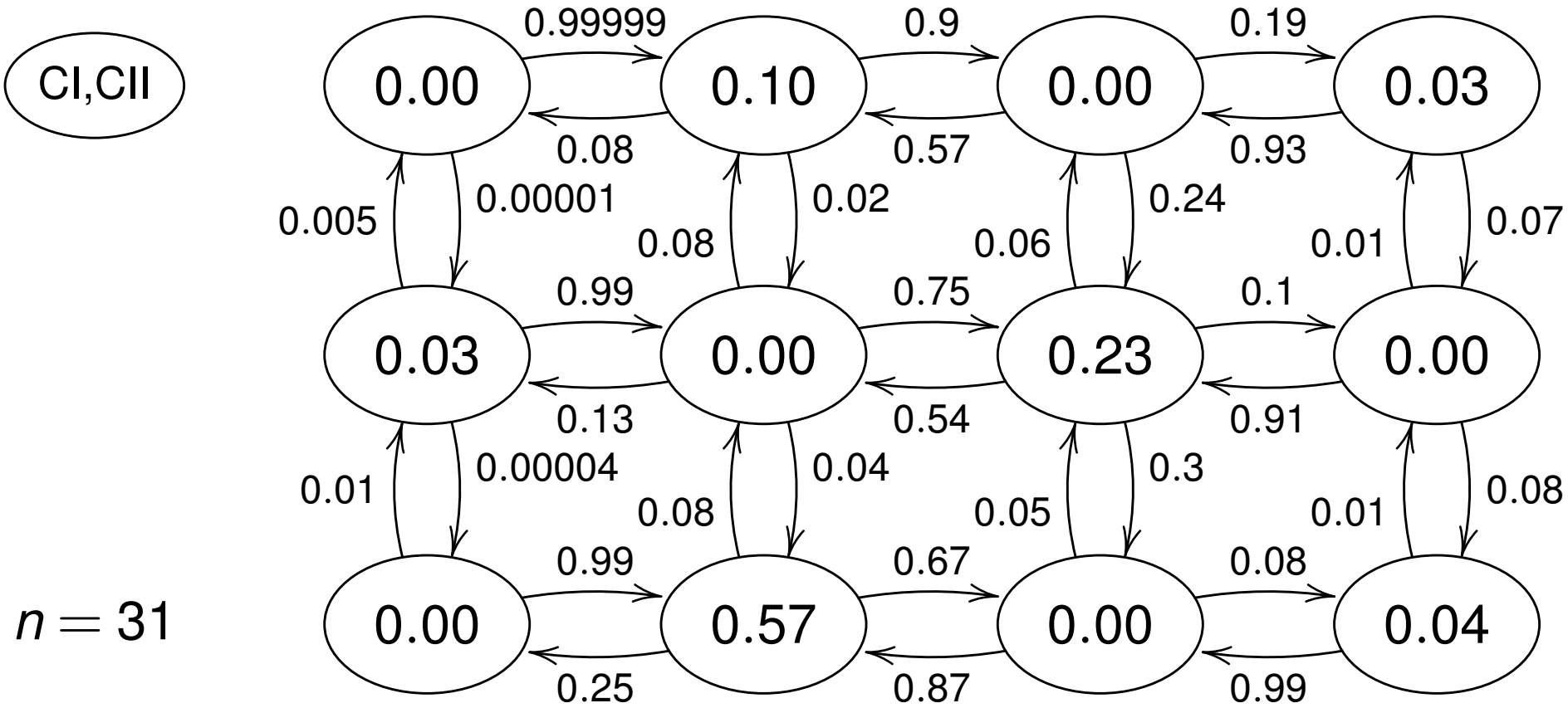
State Probabilities



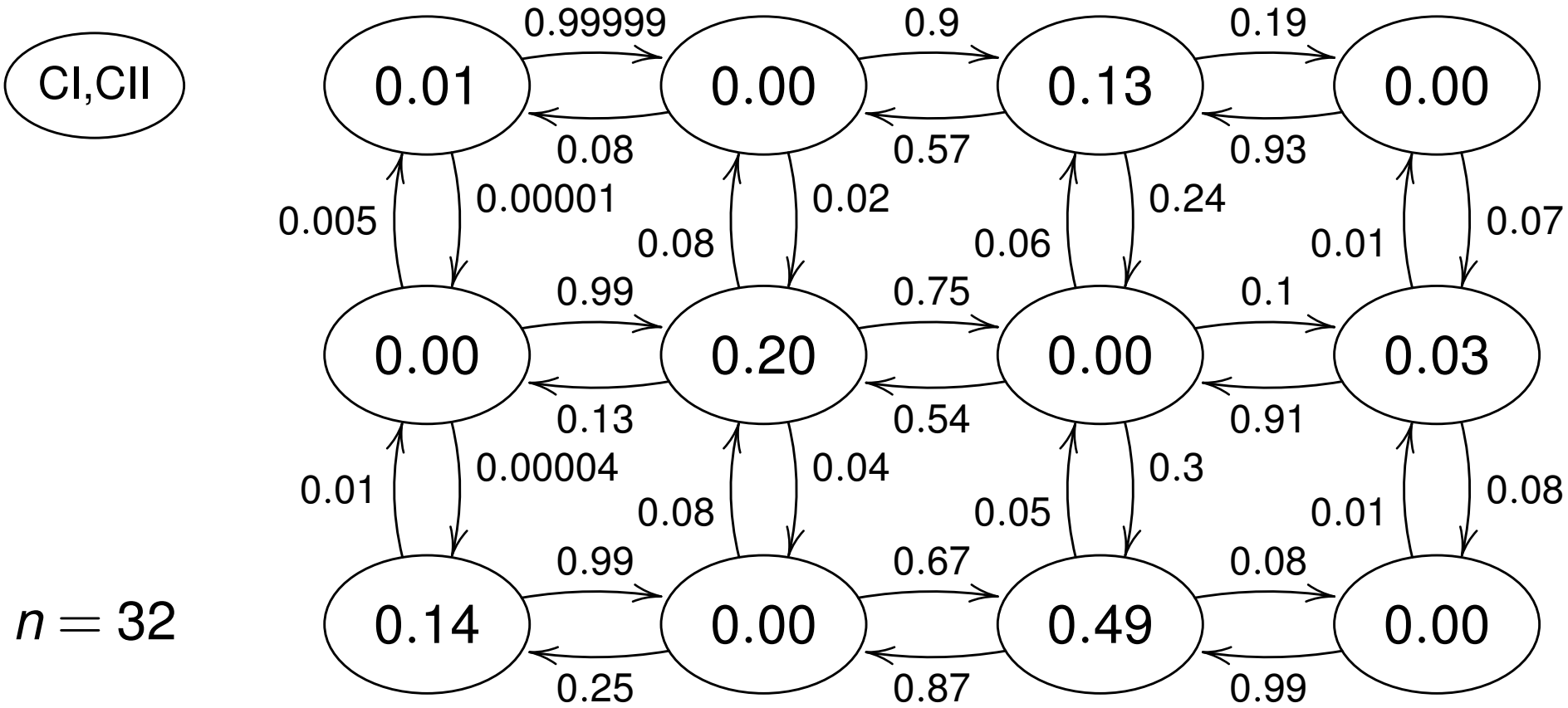
State Probabilities



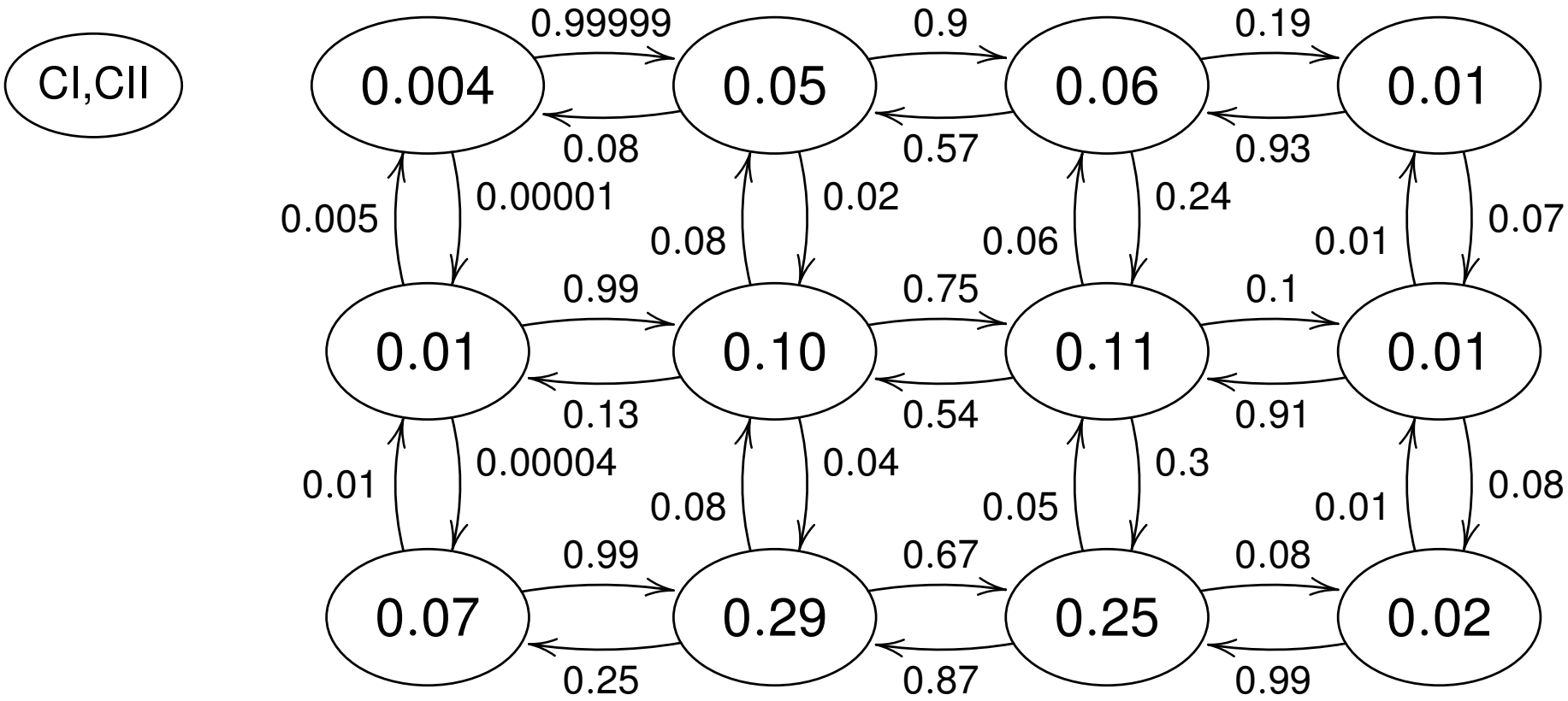
State Probabilities



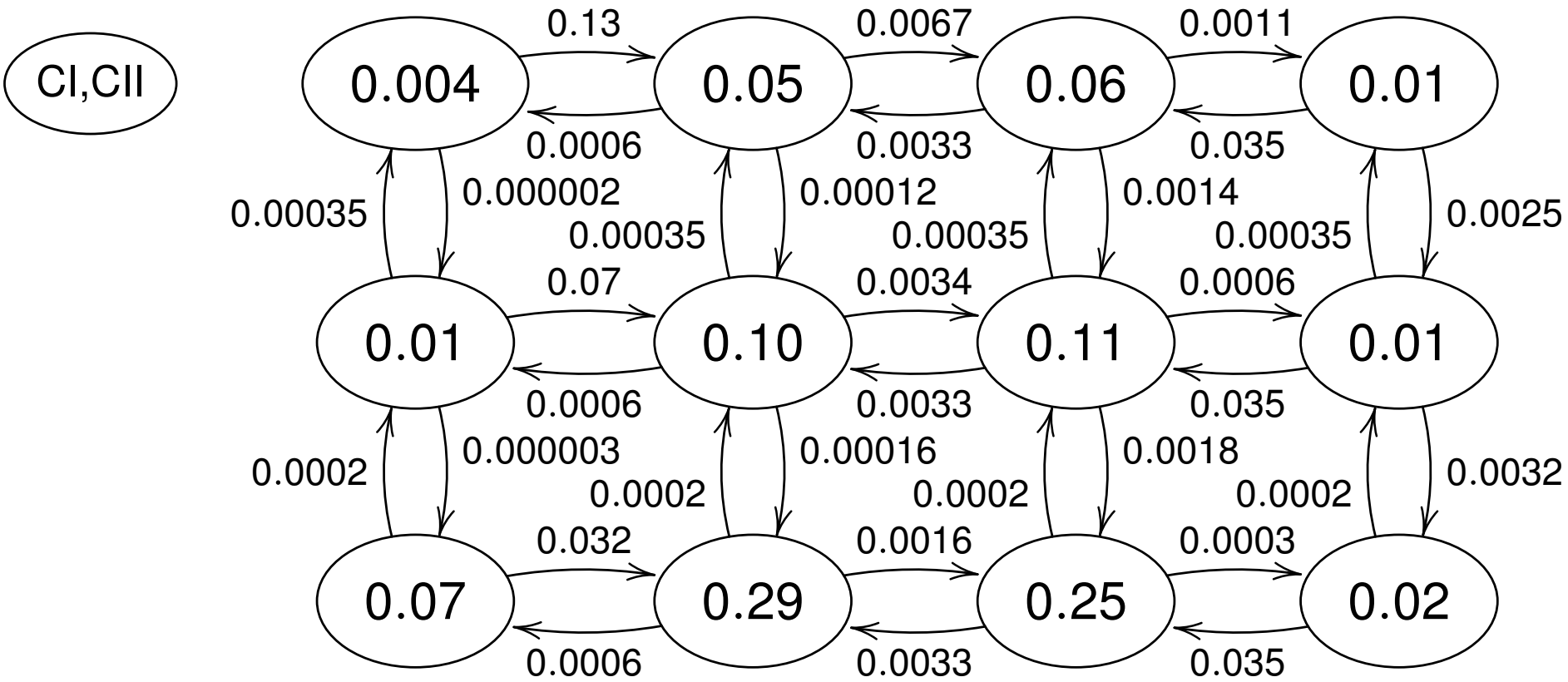
State Probabilities



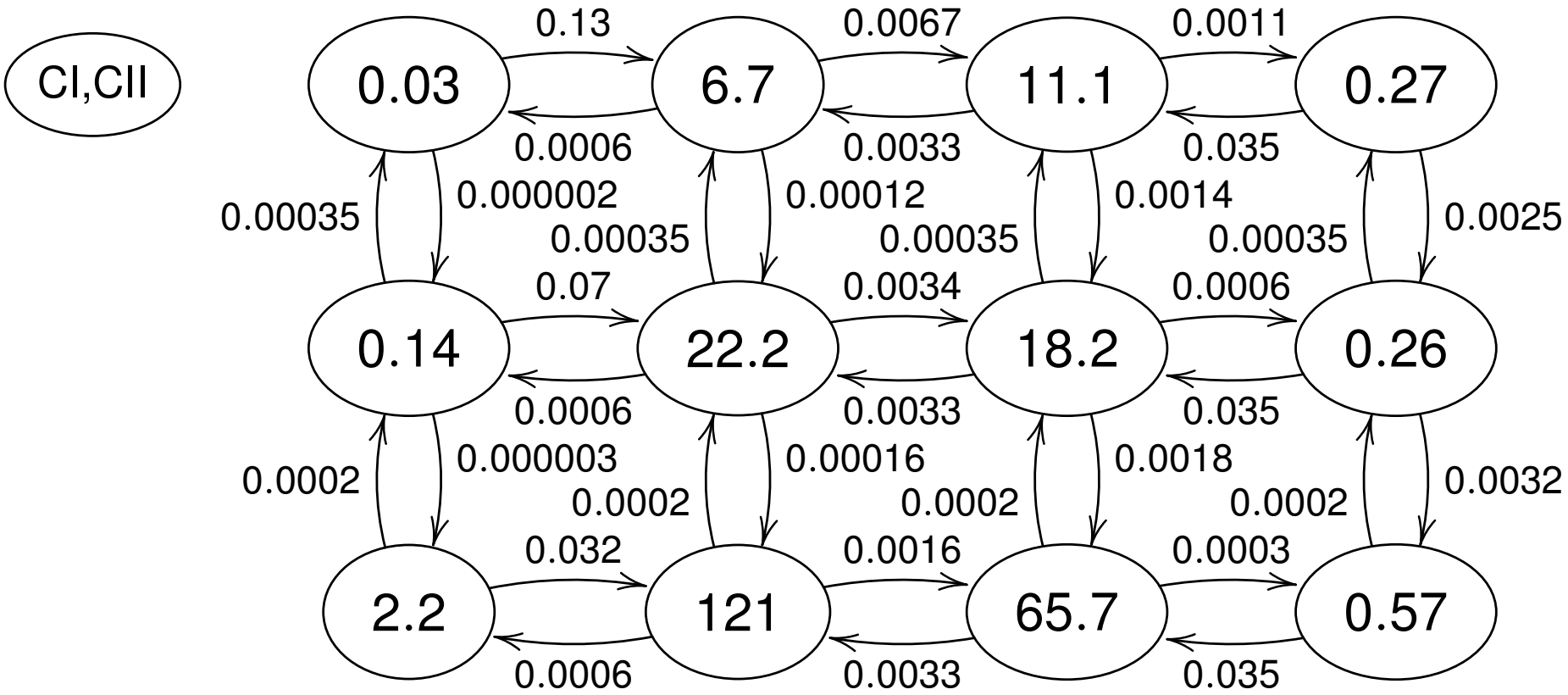
Steady-State Distribution of the EMC



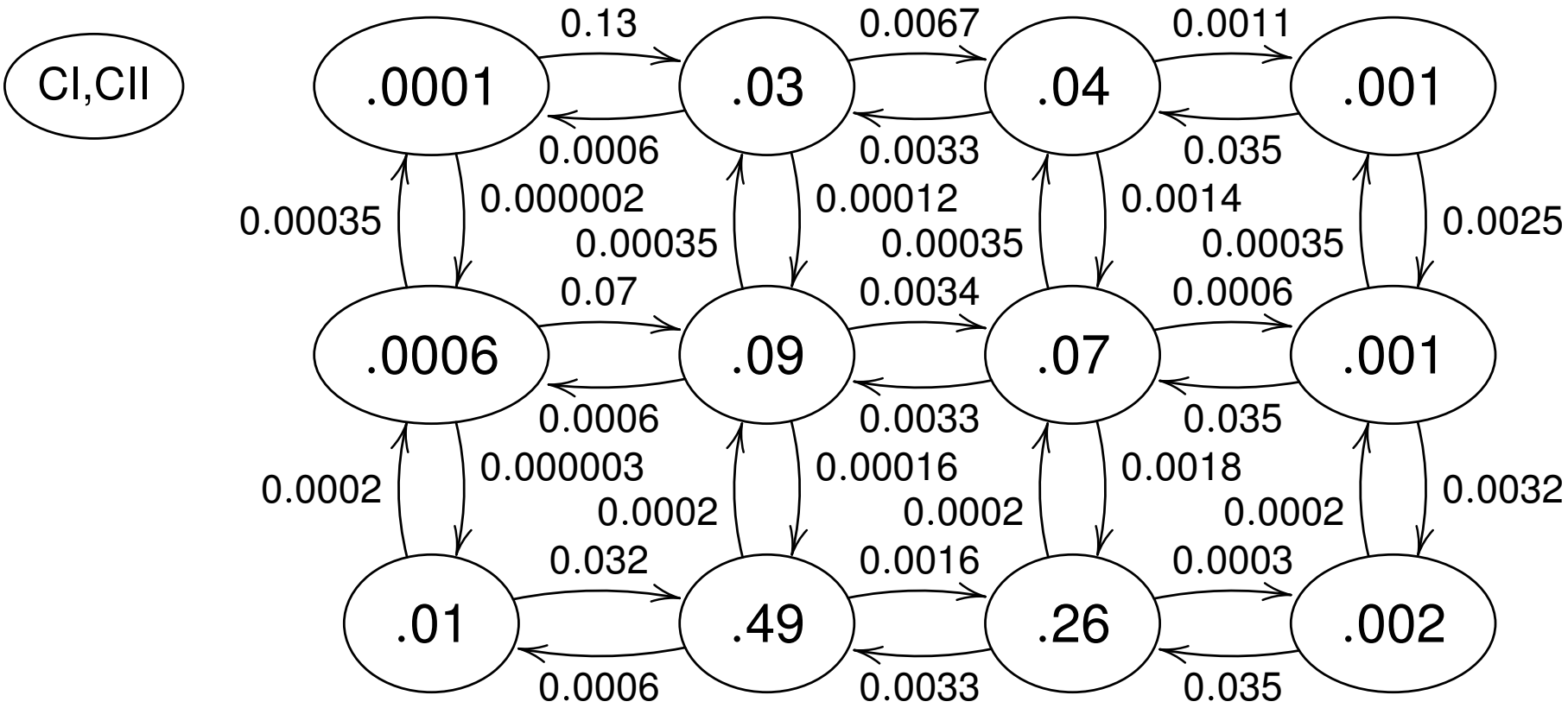
Steady-State Distribution of the CTMC



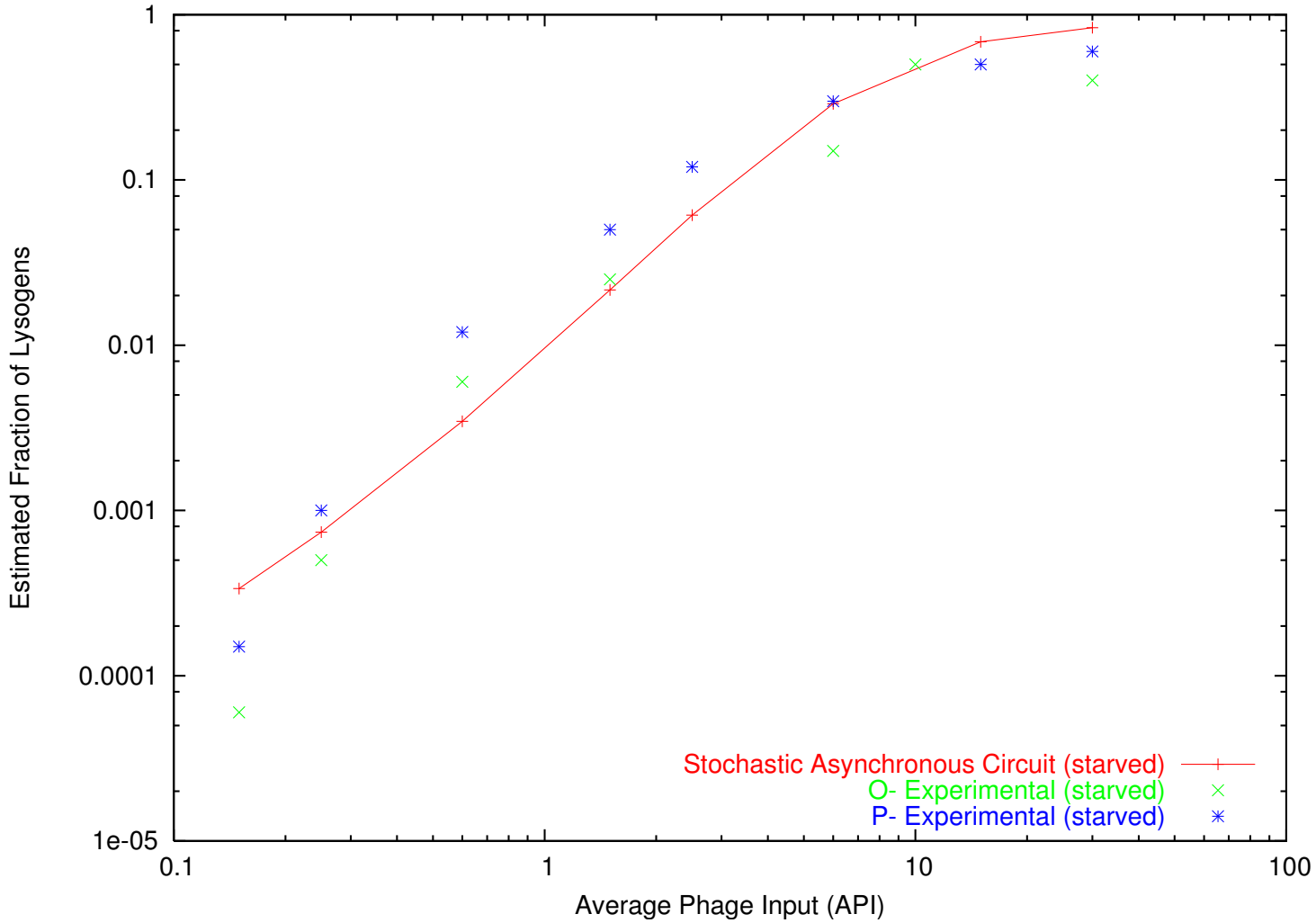
Steady-State Distribution of the CTMC



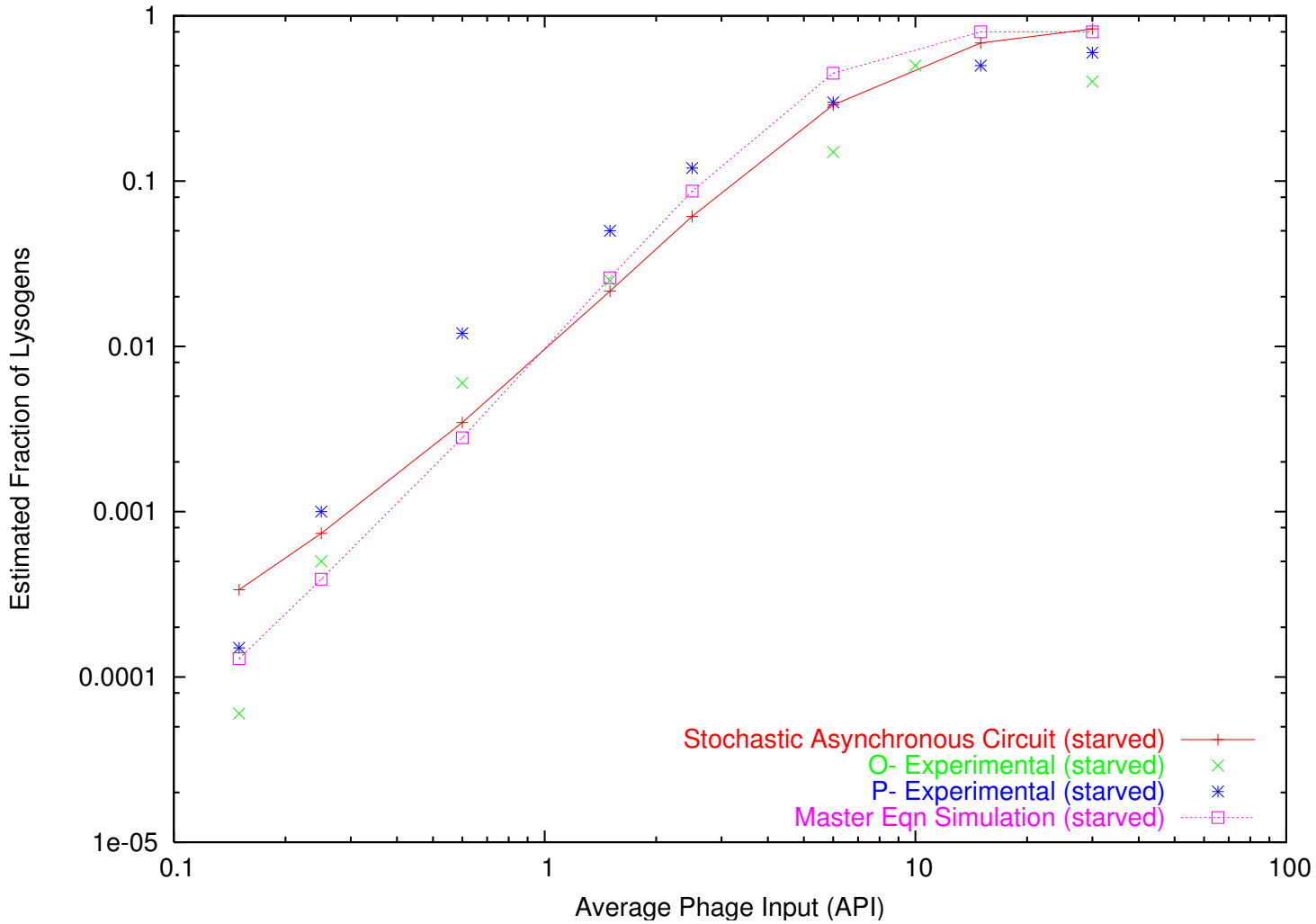
Steady-State Distribution of the CTMC



Fraction of Lysogens vs. API

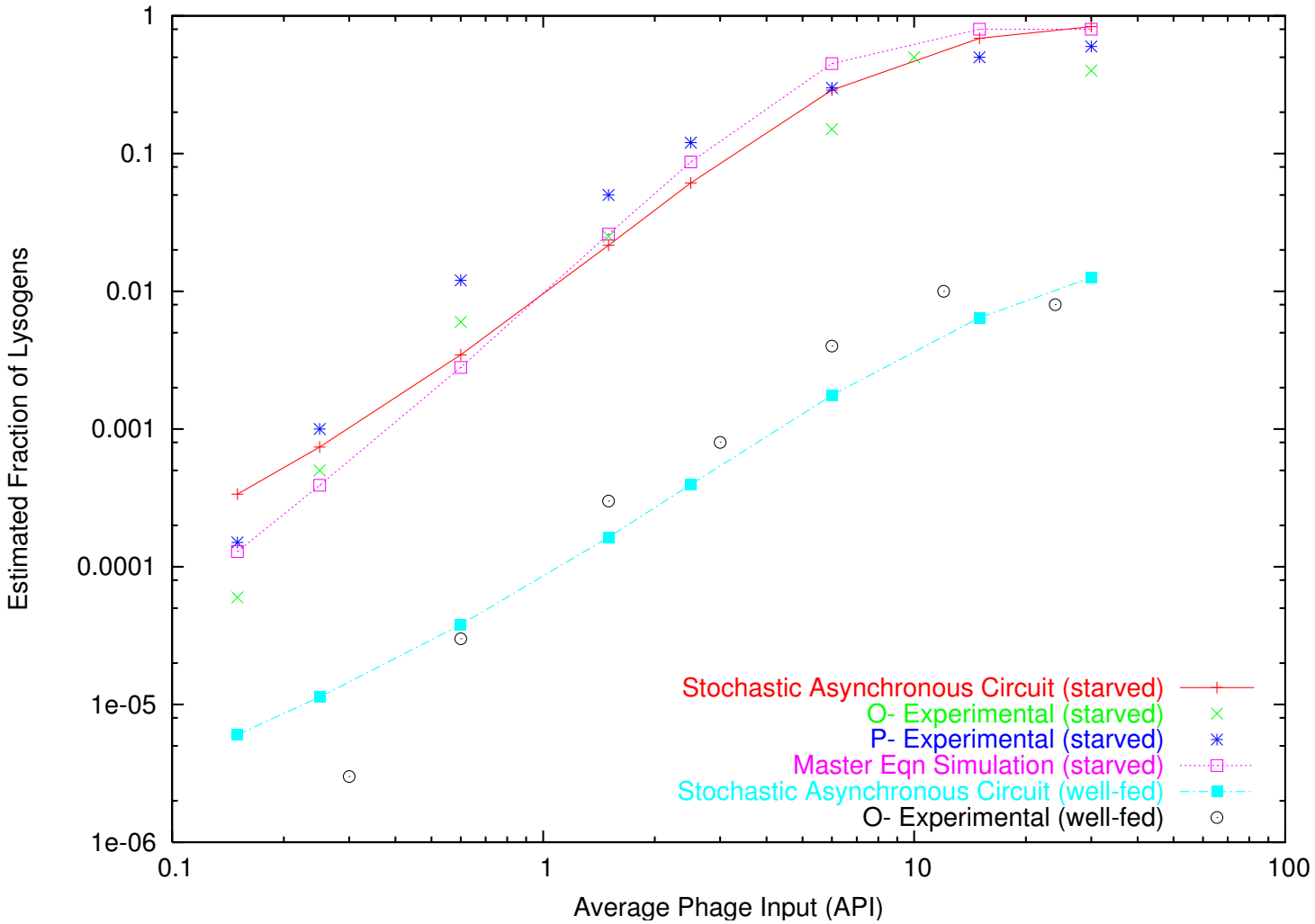


Fraction of Lysogens vs. API



Stochastic FSM results generated in only 7 minutes.

Fraction of Lysogens vs. API



Stochastic FSM results generated in only 7 minutes.

Transient Solutions

- Goal: calculate, $\pi_i(t)$, the probability of being in state i at time t.
- $\pi(t) = (\pi_0(t), \pi_1(t), \ldots)$ is the vector of all such probabilities.

$$\pi(t) \quad = \quad \pi(0)\mathcal{P}(t)$$

where $p_{ij}(t) = Prob\{X(t) = j \mid X(0) = i\}$.

Transient Solutions

- Goal: calculate, $\pi_i(t)$, the probability of being in state i at time t.
- $\pi(t) = (\pi_0(t), \pi_1(t), \ldots)$ is the vector of all such probabilities.

$$\pi(t) \quad = \quad \pi(0)\mathcal{P}(t)$$

where $p_{ij}(t) = Prob\{X(t) = j \mid X(0) = i\}$.

- *Chapman-Kolmogorov Forward Equation:*

$$\frac{\mathcal{P}(t)}{dt} \quad = \quad \mathcal{P}(t)Q$$

Transient Solutions

- Goal: calculate, $\pi_i(t)$, the probability of being in state i at time t.
- $\pi(t) = (\pi_0(t), \pi_1(t), \ldots)$ is the vector of all such probabilities.

$$\pi(t) = \pi(0)\mathcal{P}(t)$$

where $p_{ij}(t) = Prob\{X(t) = j \mid X(0) = i\}$.

- *Chapman-Kolmogorov Forward Equation:*

$$\frac{\mathcal{P}(t)}{dt} = \mathcal{P}(t)Q$$
$$\mathcal{P}(t) = e^{Qt}$$

Transient Solutions

- Goal: calculate, $\pi_i(t)$, the probability of being in state i at time t.
- $\pi(t) = (\pi_0(t), \pi_1(t), \ldots)$ is the vector of all such probabilities.

$$\pi(t) = \pi(0)\mathcal{P}(t)$$

where $p_{ij}(t) = Prob\{X(t) = j \mid X(0) = i\}$.

- *Chapman-Kolmogorov Forward Equation:*

$$\begin{aligned} \frac{\mathcal{P}(t)}{dt} &= \mathcal{P}(t)Q \\ \mathcal{P}(t) &= e^{Qt} \\ \pi(t) &= \pi(0)e^{Qt} \end{aligned}$$

Uniformization

$$\pi(t) \quad = \quad \pi(0)e^{Qt}$$

Uniformization

$$\pi(t) = \pi(0)e^{Qt}$$

$$e^{Qt} = \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!}$$

Uniformization

$$\begin{aligned} \pi(t) &= \pi(0)e^{Qt} \\ e^{Qt} &= \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \\ P &= I + \frac{1}{\Gamma}Q \quad \text{where } \Gamma = max_i|q_{ii}| \end{aligned}$$

Uniformization

$$\begin{aligned} \pi(t) &= \pi(0)e^{Qt} \\ e^{Qt} &= \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \\ Q &= \Gamma(P - I) \quad \text{where } \Gamma = max_i |q_{ii}| \end{aligned}$$

Uniformization

$$
\begin{aligned}
\pi(t) &= \pi(0)e^{Qt} \\
e^{Qt} &= \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \\
Q &= \Gamma(P - I) \quad \text{where } \Gamma = max_i \left| q_{ii} \right| \\
e^{Qt} &= e^{(\Gamma(P-I))t}
\end{aligned}
$$

Uniformization

$$
\begin{aligned}
\pi(t) &= \pi(0)e^{Qt} \\
e^{Qt} &= \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \\
Q &= \Gamma(P - I) \quad \text{where } \Gamma = max_i |q_{ii}| \\
e^{Qt} &= e^{\Gamma P t} e^{-\Gamma t}
\end{aligned}
$$

Uniformization

$$
\begin{aligned}
\pi(t) &= \pi(0)e^{Qt} \\
e^{Qt} &= \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \\
Q &= \Gamma(P - I) \quad \text{where } \Gamma = max_i |q_{ii}| \\
e^{Qt} &= e^{-\Gamma t} e^{\Gamma P t}
\end{aligned}
$$

Uniformization

$$
\begin{aligned}
\pi(t) &= \pi(0)e^{Qt} \\
e^{Qt} &= \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \\
Q &= \Gamma(P - I) \quad \text{where } \Gamma = max_i |q_{ii}| \\
e^{Qt} &= e^{-\Gamma t} \sum_{k=0}^{\infty} \frac{(\Gamma P t)^k}{k!}
\end{aligned}
$$

Uniformization

$$
\begin{aligned}
\pi(t) &= \pi(0)e^{Qt} \\
e^{Qt} &= \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \\
Q &= \Gamma(P - I) \quad \text{where } \Gamma = max_i |q_{ii}| \\
e^{Qt} &= e^{-\Gamma t} \sum_{k=0}^{\infty} P^k \frac{(\Gamma t)^k}{k!}
\end{aligned}
$$

Uniformization

$$
\begin{aligned}
\pi(t) &= \pi(0)e^{Qt} \\
e^{Qt} &= \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \\
Q &= \Gamma(P - I) \quad \text{where } \Gamma = max_i |q_{ii}| \\
e^{Qt} &= e^{-\Gamma t} \sum_{k=0}^{\infty} P^k \frac{(\Gamma t)^k}{k!} \\
\pi(t) &= \pi(0)e^{-\Gamma t} \sum_{k=0}^{\infty} P^k \frac{(\Gamma t)^k}{k!}
\end{aligned}
$$

Uniformization

$$
\begin{aligned}
\pi(t) &= \pi(0)e^{Qt} \\
e^{Qt} &= \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \\
Q &= \Gamma(P - I) \quad \text{where } \Gamma = max_i |q_{ii}| \\
e^{Qt} &= e^{-\Gamma t} \sum_{k=0}^{\infty} P^k \frac{(\Gamma t)^k}{k!} \\
\pi(t) &= e^{-\Gamma t} \sum_{k=0}^{\infty} \pi(0) P^k \frac{(\Gamma t)^k}{k!}
\end{aligned}
$$

Uniformization

$$
\begin{aligned}
\pi(t) &= \pi(0)e^{Qt} \\
e^{Qt} &= \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \\
Q &= \Gamma(P - I) \quad \text{where } \Gamma = max_i |q_{ii}| \\
e^{Qt} &= e^{-\Gamma t} \sum_{k=0}^{\infty} P^k \frac{(\Gamma t)^k}{k!} \\
\pi(t) &= e^{-\Gamma t} \sum_{k=0}^{\infty} \pi(0) P^k \frac{(\Gamma t)^k}{k!} \\
\pi(0)P^k &= (\pi(0)P^{k-1})P
\end{aligned}
$$

Truncation Error

$$\pi(t) = e^{-\Gamma t} \sum_{k=0}^{\infty} \pi(0) P^k \frac{(\Gamma t)^k}{k!}$$

Truncation Error

$$\pi^*(t) = e^{-\Gamma t} \sum_{k=0}^{K} \pi(0) P^k \frac{(\Gamma t)^k}{k!}$$

Truncation Error

$$\pi^*(t) = e^{-\Gamma t} \sum_{k=0}^{K} \pi(0) P^k \frac{(\Gamma t)^k}{k!}$$

$$\| \pi(t) - \pi^*(t) \|_\infty \leq 1 - e^{-\Gamma t} \sum_{k=0}^{K} \frac{(\Gamma t)^k}{k!}$$

Truncation Error

$$\begin{aligned}
\pi^*(t) &= e^{-\Gamma t} \sum_{k=0}^{K} \pi(0) P^k \frac{(\Gamma t)^k}{k!} \\
\| \pi(t) - \pi^*(t) \|_\infty &\leq 1 - e^{-\Gamma t} \sum_{k=0}^{K} \frac{(\Gamma t)^k}{k!} \\
1 - e^{-\Gamma t} \sum_{k=0}^{K} \frac{(\Gamma t)^k}{k!} &\leq \varepsilon
\end{aligned}$$

Truncation Error

$$\begin{aligned}
\pi^*(t) &= e^{-\Gamma t} \sum_{k=0}^{K} \pi(0) P^k \frac{(\Gamma t)^k}{k!} \\
\| \pi(t) - \pi^*(t) \|_\infty &\leq 1 - e^{-\Gamma t} \sum_{k=0}^{K} \frac{(\Gamma t)^k}{k!} \\
1 - e^{-\Gamma t} \sum_{k=0}^{K} \frac{(\Gamma t)^k}{k!} &\leq \varepsilon \\
\sum_{k=0}^{K} \frac{(\Gamma t)^k}{k!} &\geq \frac{1-\varepsilon}{e^{-\Gamma t}}
\end{aligned}$$

Transient Analysis Algorithm

1. $K = 0, \xi = 1, \sigma = 1$
2. $\eta = (1 - \varepsilon)/e^{-\Gamma t}$.
3. While $\sigma < \eta$ do
   1. $K = K + 1$
   2. $\xi = \xi \times (\Gamma t)/K$
   3. $\sigma = \sigma + \xi$
4. $\pi = y = \pi(0)$
5. For $k = 1$ to K do
   1. $y = yP \times (\Gamma t)/k$
   2. $\pi = \pi + y$
6. $\pi(t) = e^{-\Gamma t}\pi$

Transient Analysis Example

$$Q = \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix}$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0]
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 0.01
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 0.01 \\
\eta &= 1.1
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 0.01 \\
\eta &= 1.1 \\
K &= 1
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 0.01 \\
\eta &= 1.1 \\
K &= 1 \\
\pi(0.01) &= [0.94 \;\; 0.04 \;\; 0.02]
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 0.1
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 0.1 \\
\eta &= 2.2
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 0.1 \\
\eta &= 2.2 \\
K &= 3
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 0.1 \\
\eta &= 2.2 \\
K &= 3 \\
\pi(0.1) &= [0.62 \;\; 0.27 \;\; 0.10]
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 1.0
\end{aligned}
$$

Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 1.0 \\
\eta &= 2951
\end{aligned}
$$

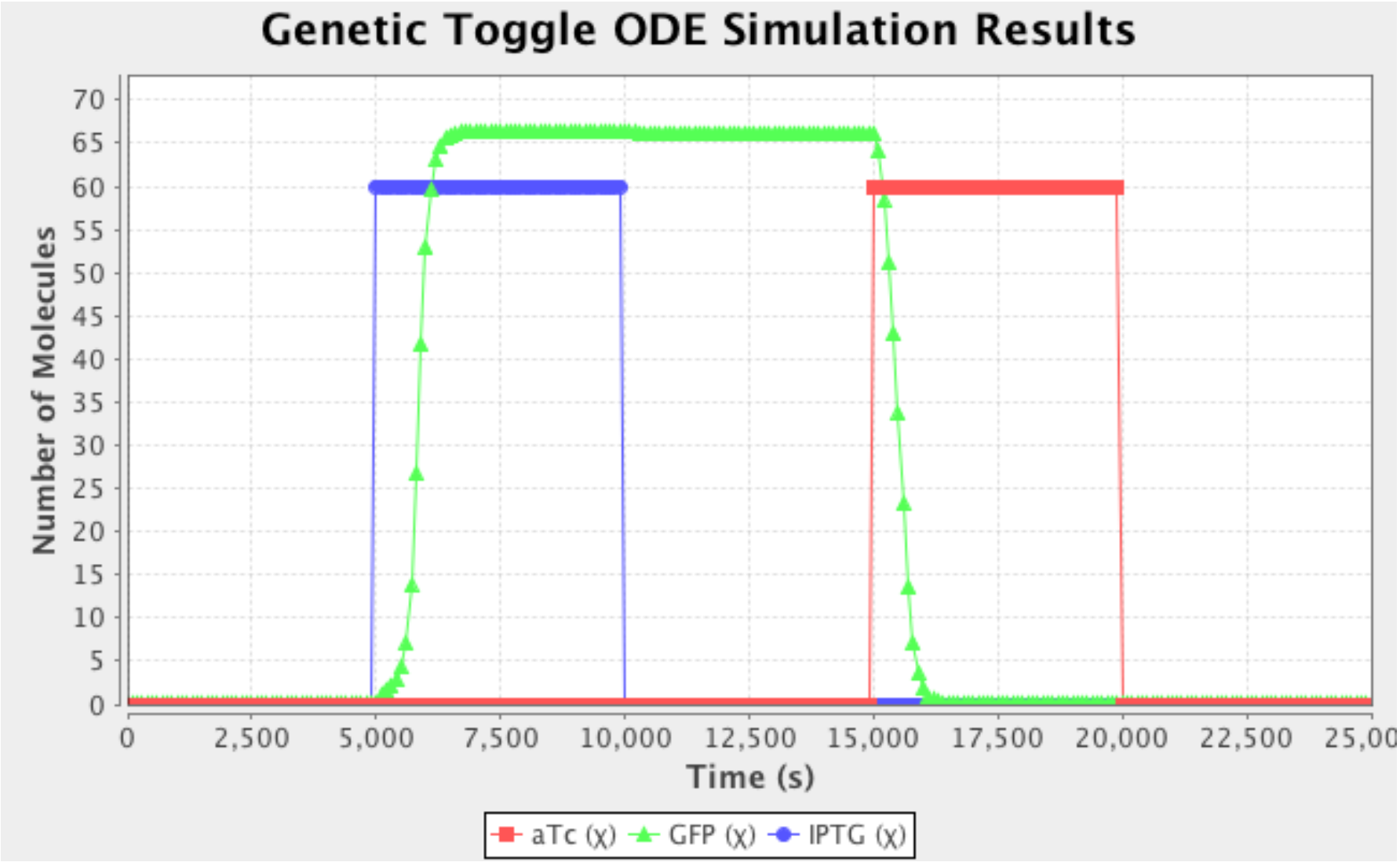
Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 1.0 \\
\eta &= 2951 \\
K &= 15
\end{aligned}
$$

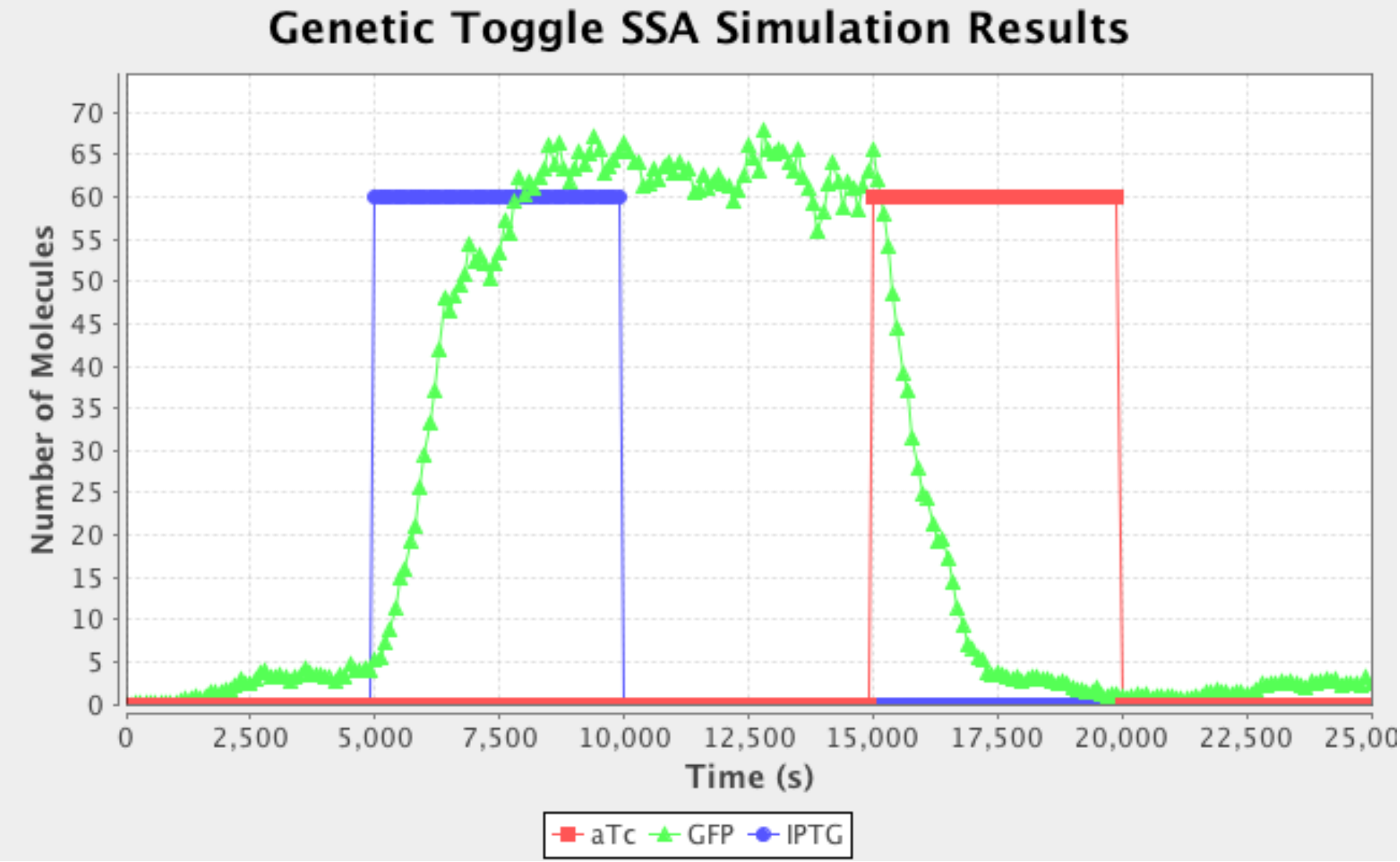
Transient Analysis Example

$$
\begin{aligned}
Q &= \begin{pmatrix} -6 & 4 & 2 \\ 4 & -4 & 0 \\ 4 & 4 & -8 \end{pmatrix} \\
\Gamma &= 8 \\
\varepsilon &= 0.01 \\
\pi(0) &= [1.0 \;\; 0.0 \;\; 0.0] \\
t &= 1.0 \\
\eta &= 2951 \\
K &= 15 \\
\pi(1.0) &= [0.4 \;\; 0.5 \;\; 0.1]
\end{aligned}
$$

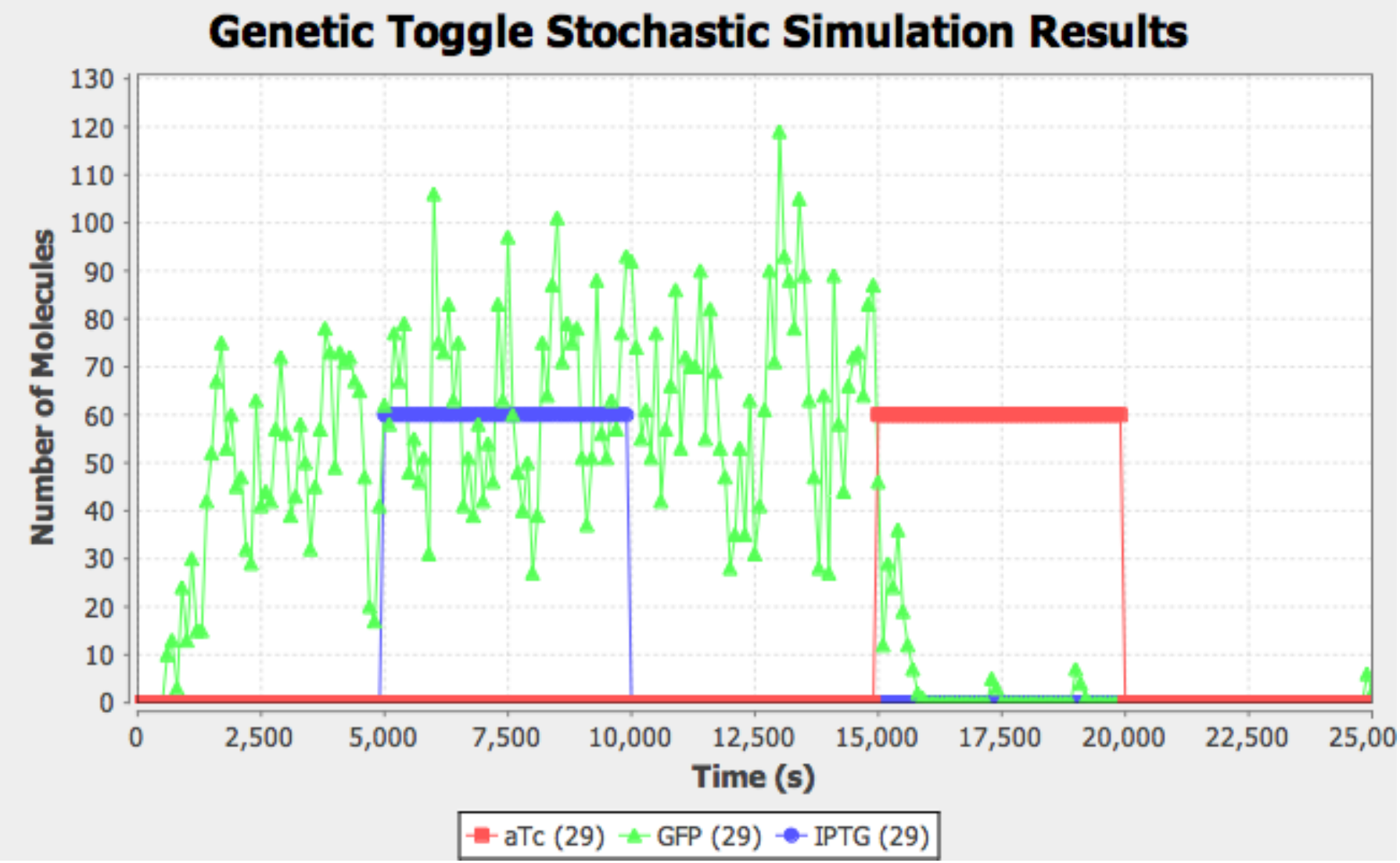
Classical Chemical Kinetics: ODE Analysis



Stochastic Simulation of the Genetic Toggle Switch



Stochastic Simulation of the Genetic Toggle Switch



Population Simulation for Toggle Switch Failure

(Loading ToggleFailSim.mov)

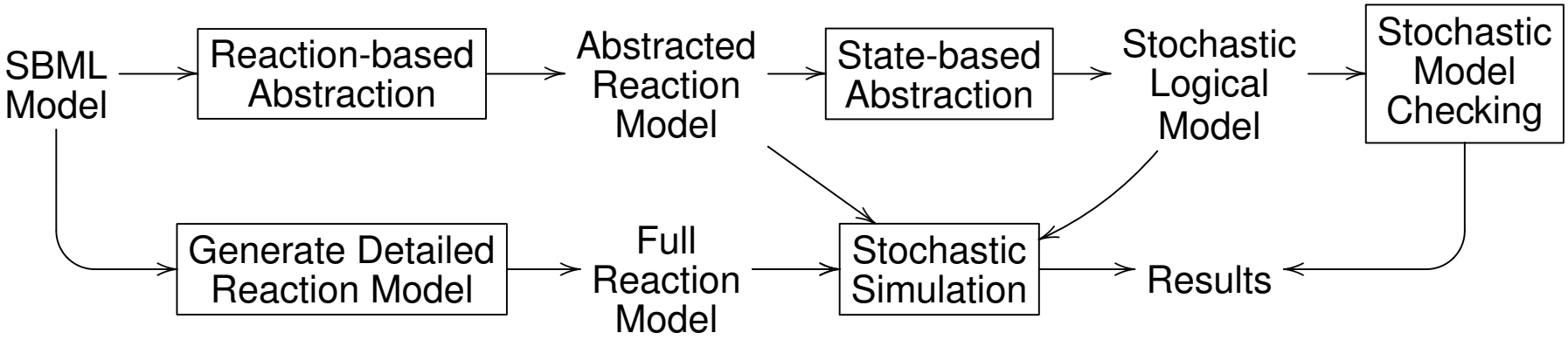
Stevens et al., ACS Synthetic Biology (2013)
Watanabe et al., Frontiers (2014)
Watanabe et al., ACS Synthetic Biology (2016)

Population Simulation for Toggle Switch Response Time

(Loading ToggleResponseSim.mov)

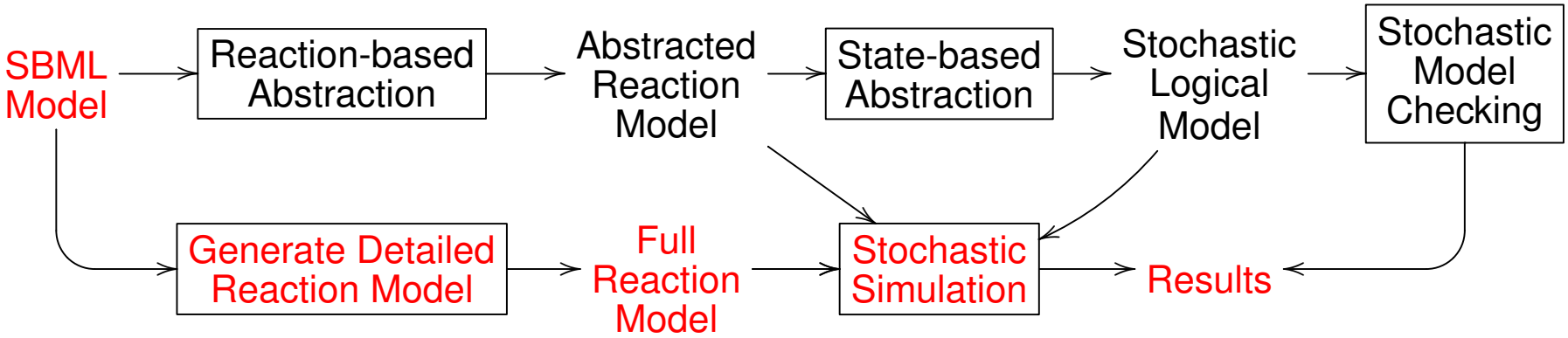
Stevens et al., ACS Synthetic Biology (2013)
Watanabe et al., Frontiers (2014)
Watanabe et al., ACS Synthetic Biology (2016)

Abstraction for Genetic Circuits



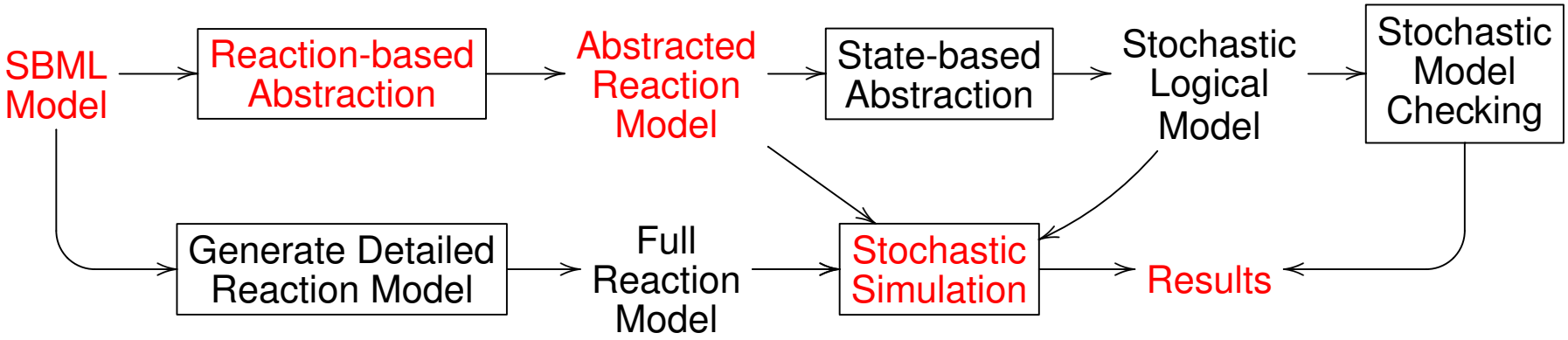
Kuwahara et al., Trans. on Computational Systems Biology (2006).
Kuwahara et al., Journal of Computational Biology (2008).
Kuwahara et al., PLoS Computational Biology (2010).
Madsen, Zhang et al., Comp. Int. in Bioinfo./CompBio., 2012 (**best student paper**).
Madsen, Zhang et al., ACM Jour. on Emerging Tech. in Comp. Sys. (2014).

Abstraction for Genetic Circuits



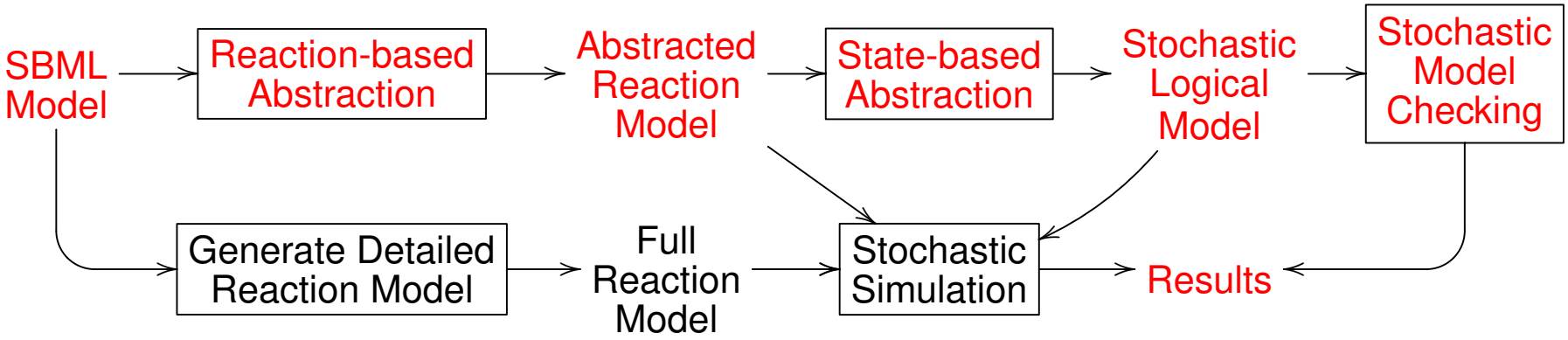
Kuwahara et al., Trans. on Computational Systems Biology (2006).
Kuwahara et al., Journal of Computational Biology (2008).
Kuwahara et al., PLoS Computational Biology (2010).
Madsen, Zhang et al., Comp. Int. in Bioinfo./CompBio., 2012 (**best student paper**).
Madsen, Zhang et al., ACM Jour. on Emerging Tech. in Comp. Sys. (2014).

Abstraction for Genetic Circuits



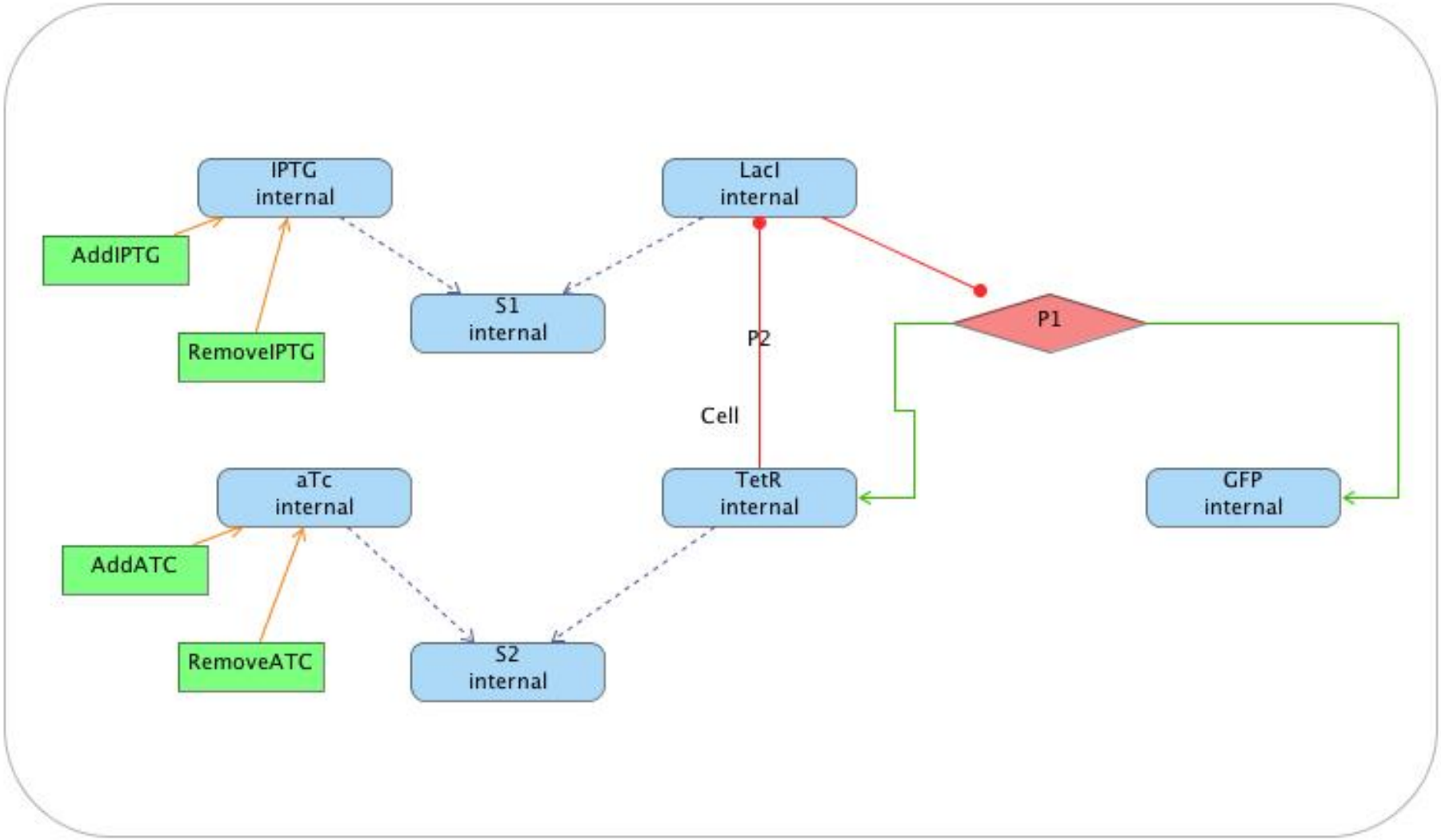
Kuwahara et al., Trans. on Computational Systems Biology (2006).
Kuwahara et al., Journal of Computational Biology (2008).
Kuwahara et al., PLoS Computational Biology (2010).
Madsen, Zhang et al., Comp. Int. in Bioinfo./CompBio., 2012 (**best student paper**).
Madsen, Zhang et al., ACM Jour. on Emerging Tech. in Comp. Sys. (2014).

Abstraction for Genetic Circuits

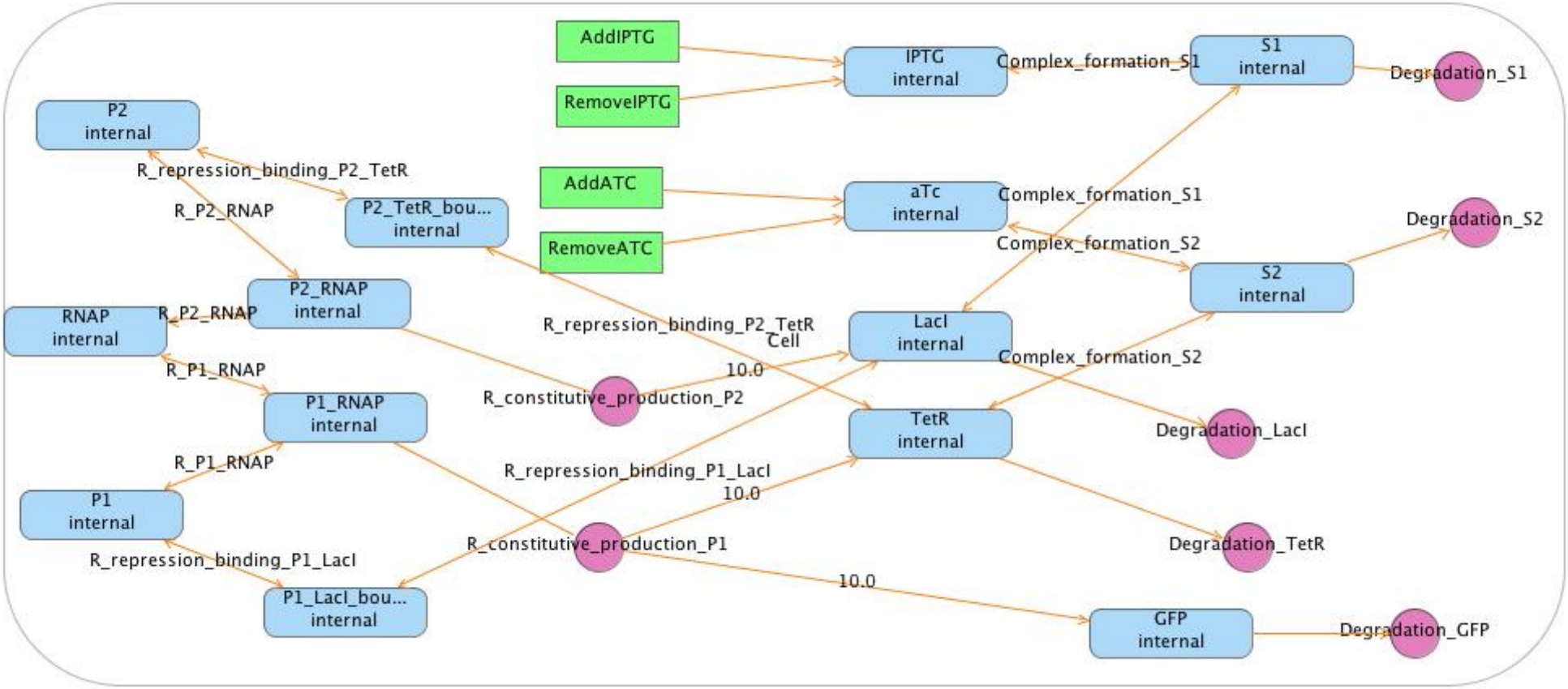


Kuwahara et al., Trans. on Computational Systems Biology (2006).
Kuwahara et al., Journal of Computational Biology (2008).
Kuwahara et al., PLoS Computational Biology (2010).
Madsen, Zhang et al., Comp. Int. in Bioinfo./CompBio., 2012 (**best student paper**).
Madsen, Zhang et al., ACM Jour. on Emerging Tech. in Comp. Sys. (2014).

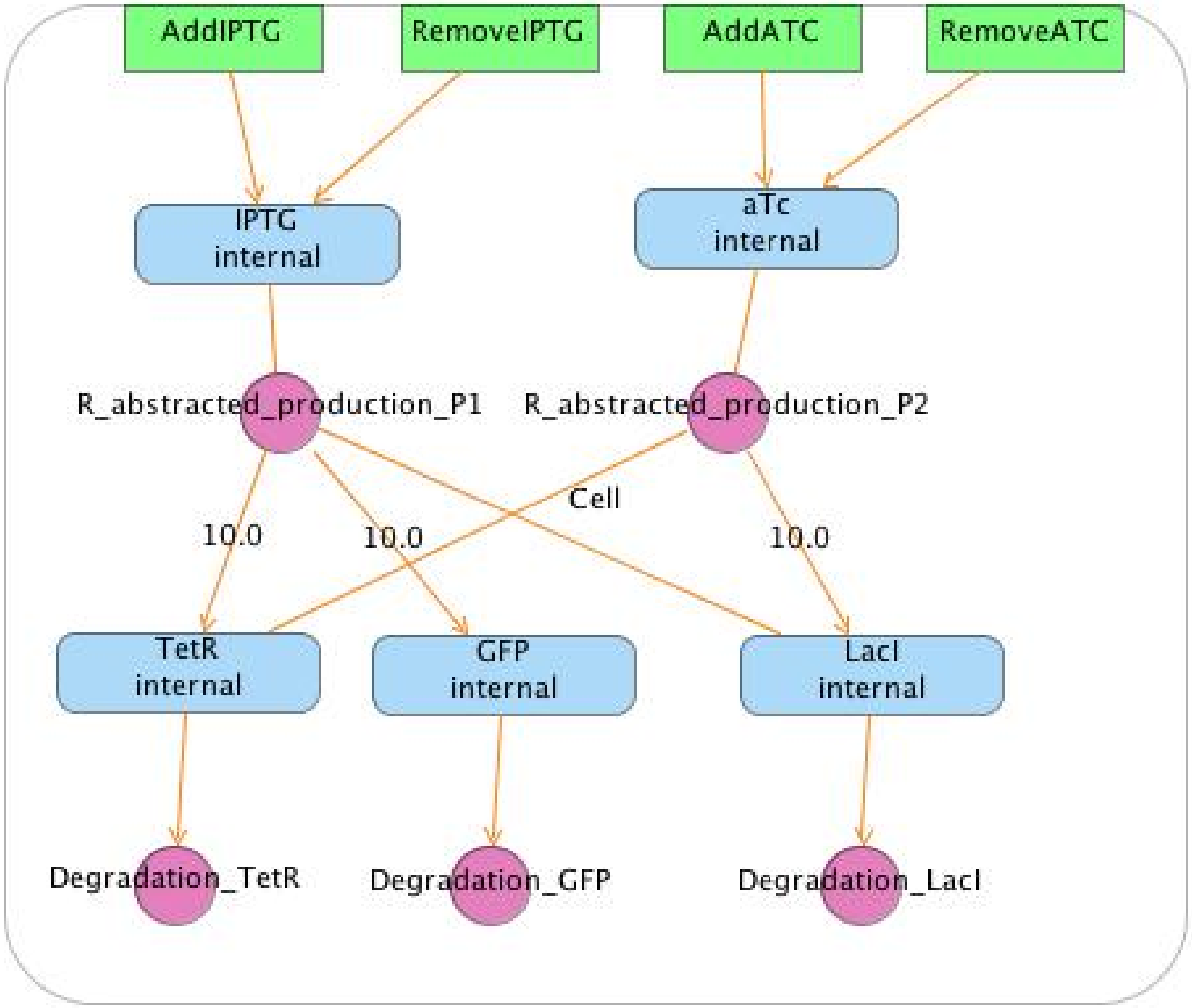
Genetic Toggle Switch: Original Model



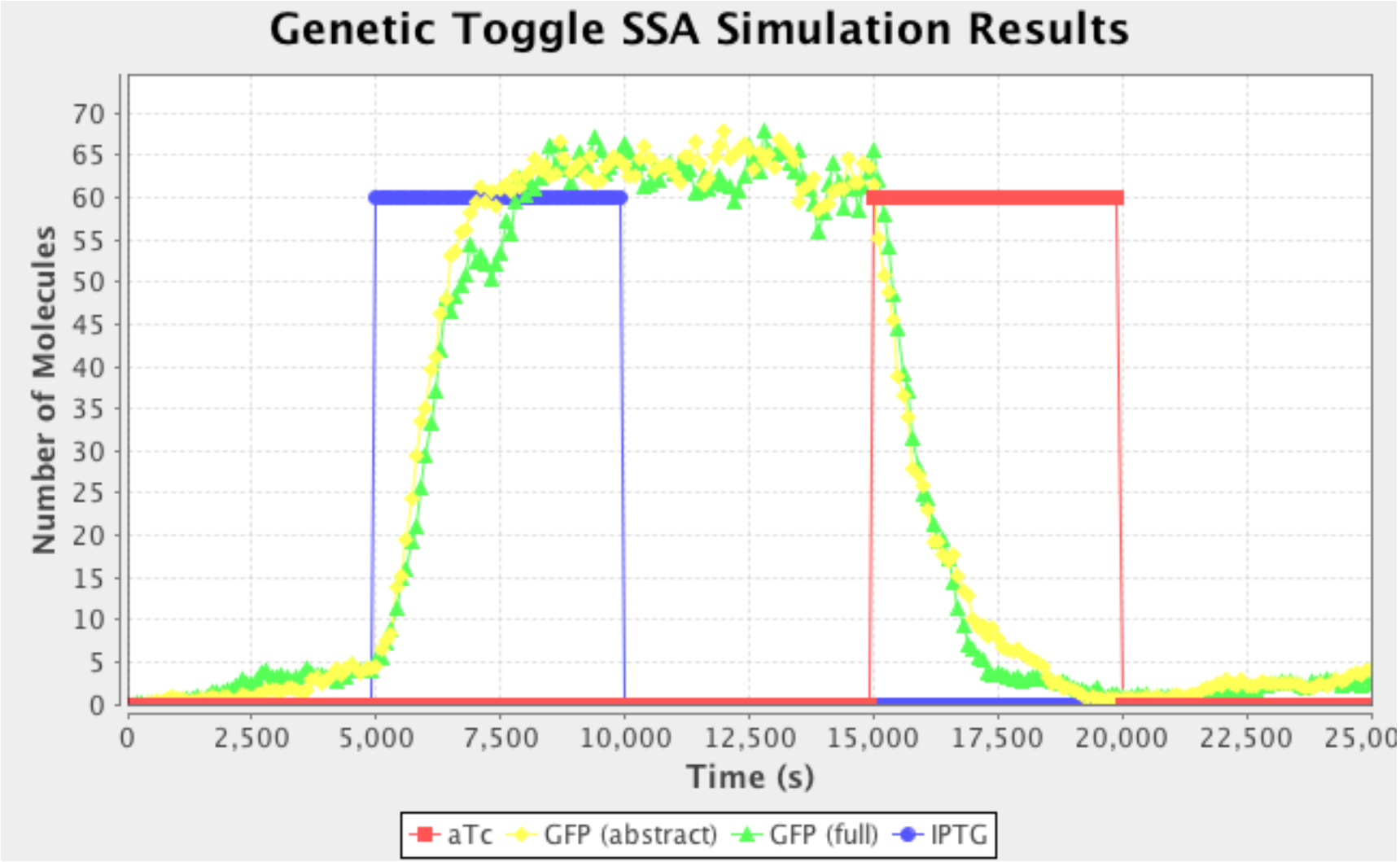
Genetic Toggle Switch: Full Reaction Model



Genetic Toggle Switch: Abstracted Reaction Model



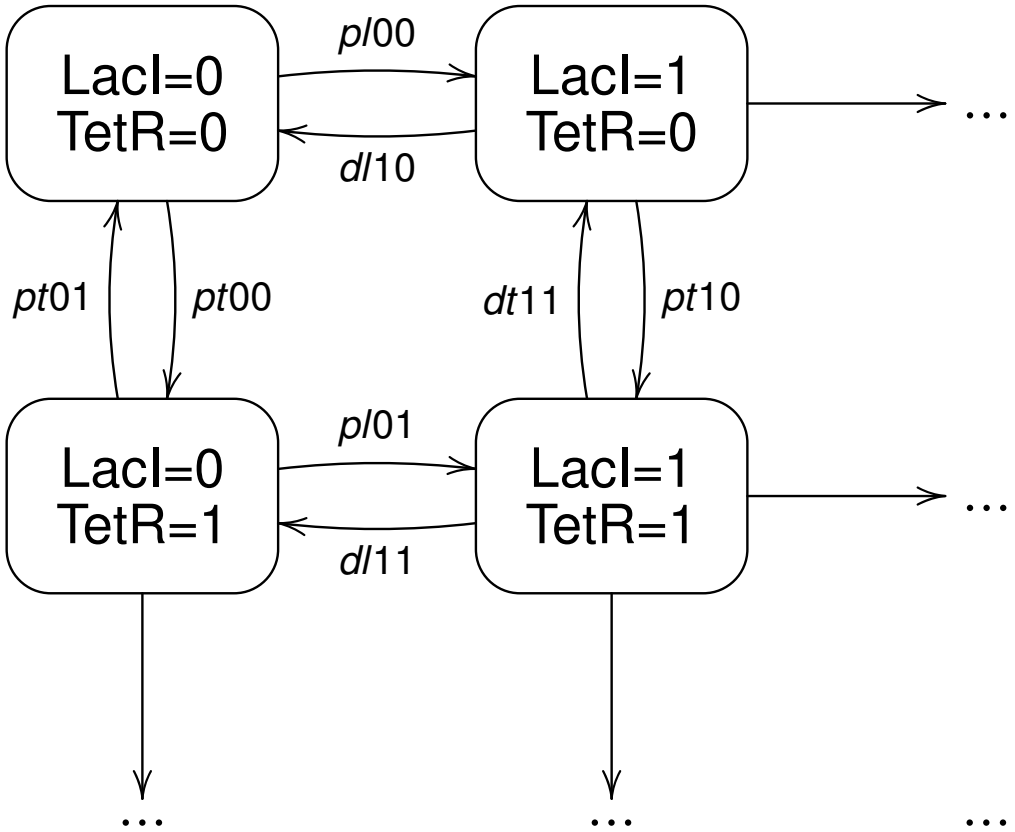
Abstraction Results for the Genetic Toggle Switch



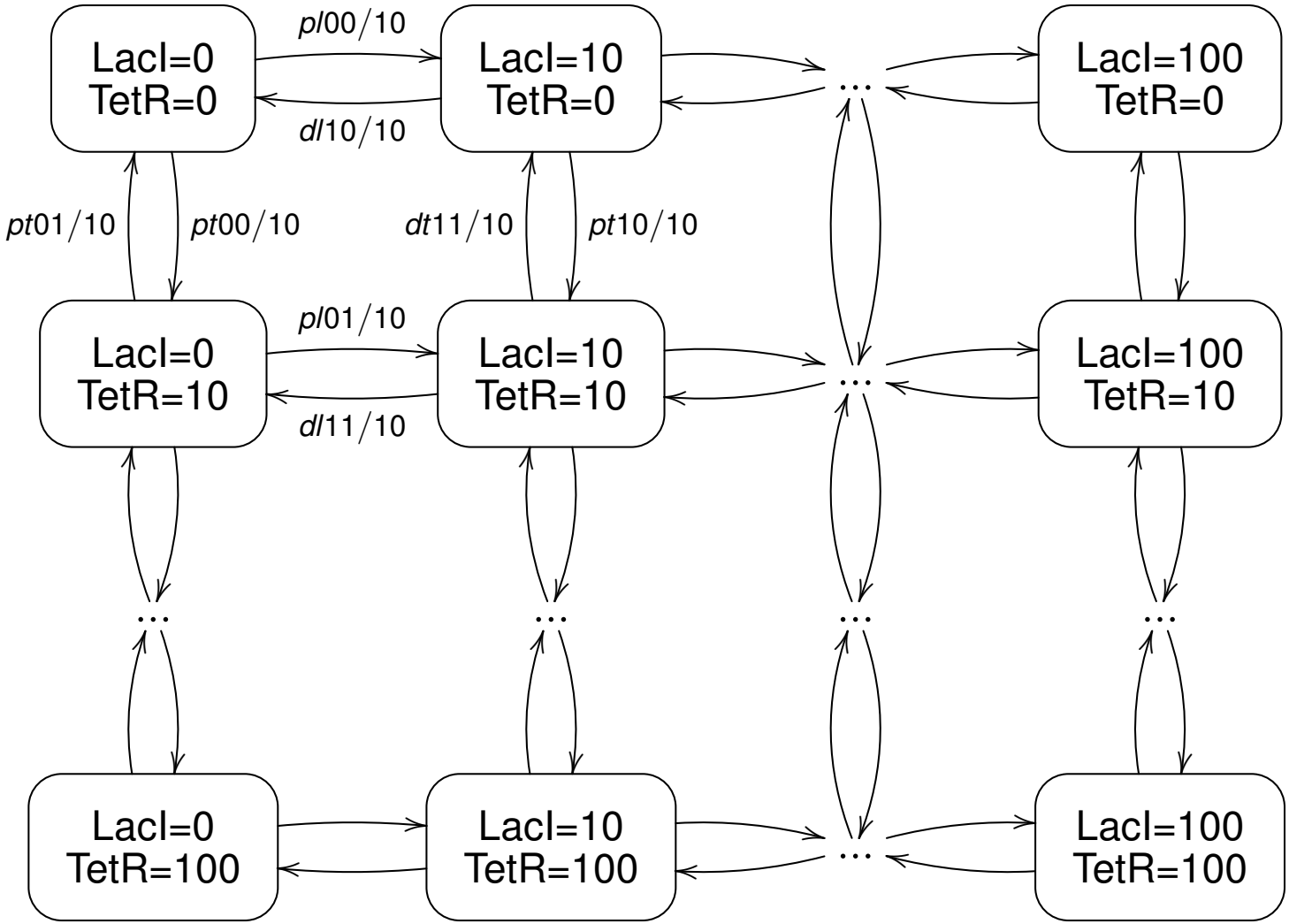
Average of 100 simulation runs.

Full results in about 4 minutes while abstracted results in 15 seconds.

State-based Abstraction



State-based Abstraction

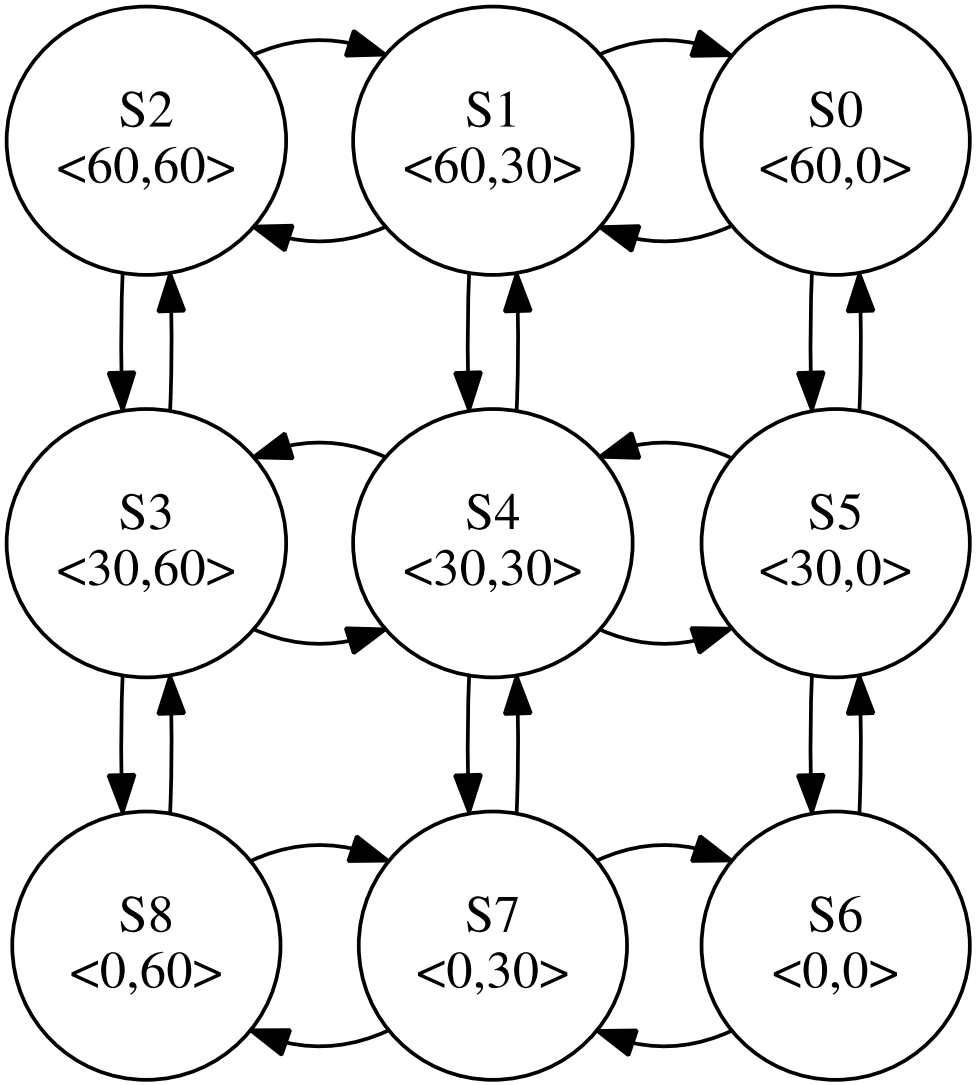


State-based Abstraction

- Our method partitions the state space into equivalence classes given a level set for each species.
- It begins by creating a sparse matrix where each entry, $p_{i,j}$, represents the rate of moving from state i to state j.
- A state is then created with an encoding of the initial values of the species in the genetic circuit.
- Next, a depth first search is performed by changing species values to find the state space.

State Graph for Toggle Switch

- Levels selected at 0, 30, and 60 for both LacI and TetR.



<LacI,TetR>

Transition Rate Calculations

- The transition rates of moving between states are computed using either an operator site reduction approximation which uses a quasi-steady state assumption or an amplified degradation rate.

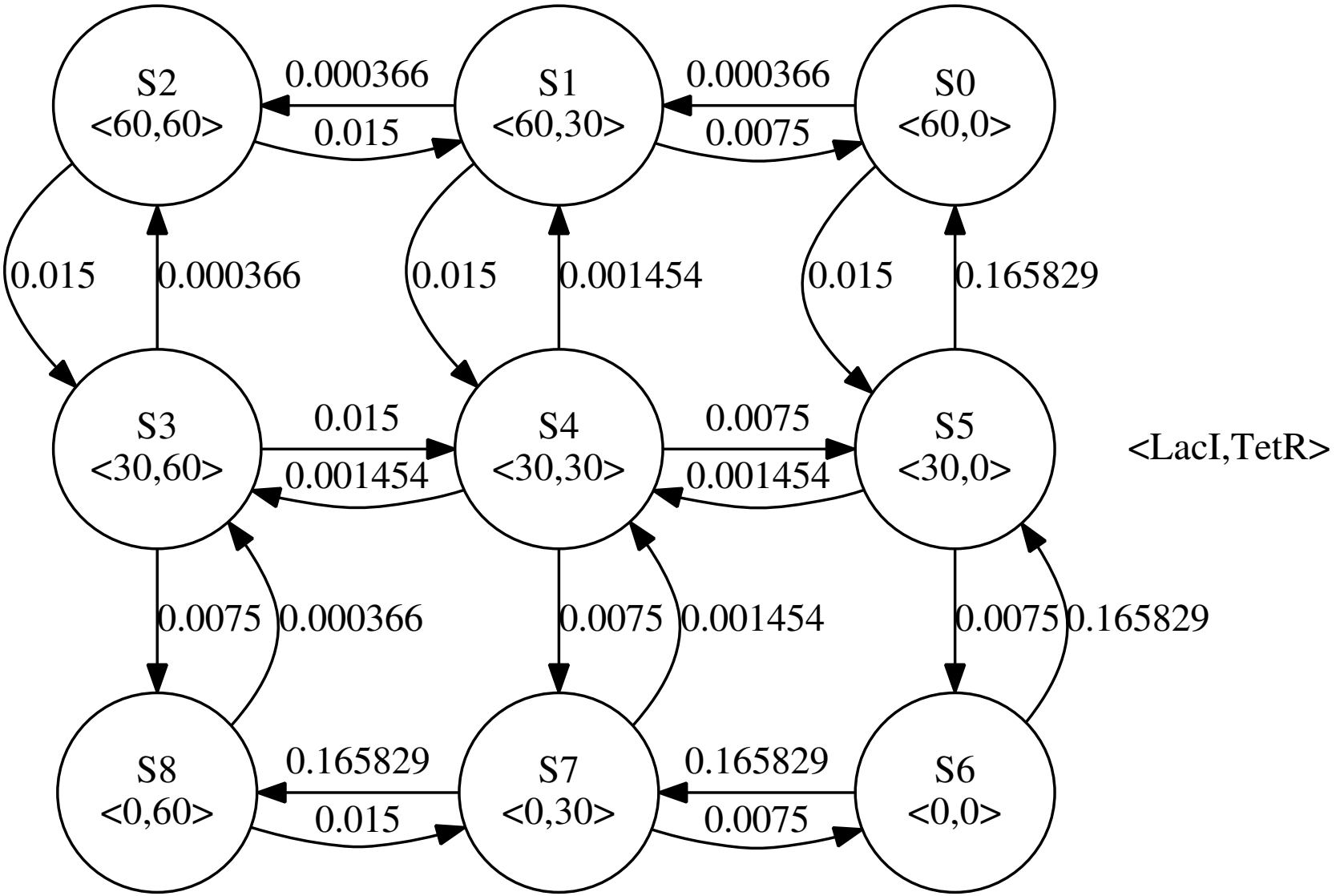
$$\text{production}(s, l, l') = \frac{\sum_{p \in Pro(s)} n_p \cdot \text{rate}(p)}{(l' - l)}$$

$$\text{degradation}(s, l, l') = \frac{k_d l'}{(l' - l)}$$

$$\text{rate}(p) = \begin{cases} \dfrac{n_p k_o n_g K_o n_r}{1 + K_o n_r + \sum_{s_r \in \text{Rep}(p)} (K_r v_{s_r})^{n_c}} & \text{if } |\text{Act}(p)| = 0 \\ \dfrac{n_p k_b n_g K_o n_r + \sum_{s_a \in \text{Act}(p)} n_p k_a n_g K_{oa} n_r (K_a v_{s_a})^{n_c}}{1 + K_o n_r + \sum_{s_r \in \text{Rep}(p)} (K_r v_{s_r})^{n_c} + \sum_{s_a \in \text{Act}(p)} K_{oa} n_r (K_a v_{s_a})^{n_c}} & \text{otherwise} \end{cases}$$

- These rates are added to the transition matrix resulting in a *continuous time Markov chain* (CTMC).

CTMC for Toggle Switch



Properties

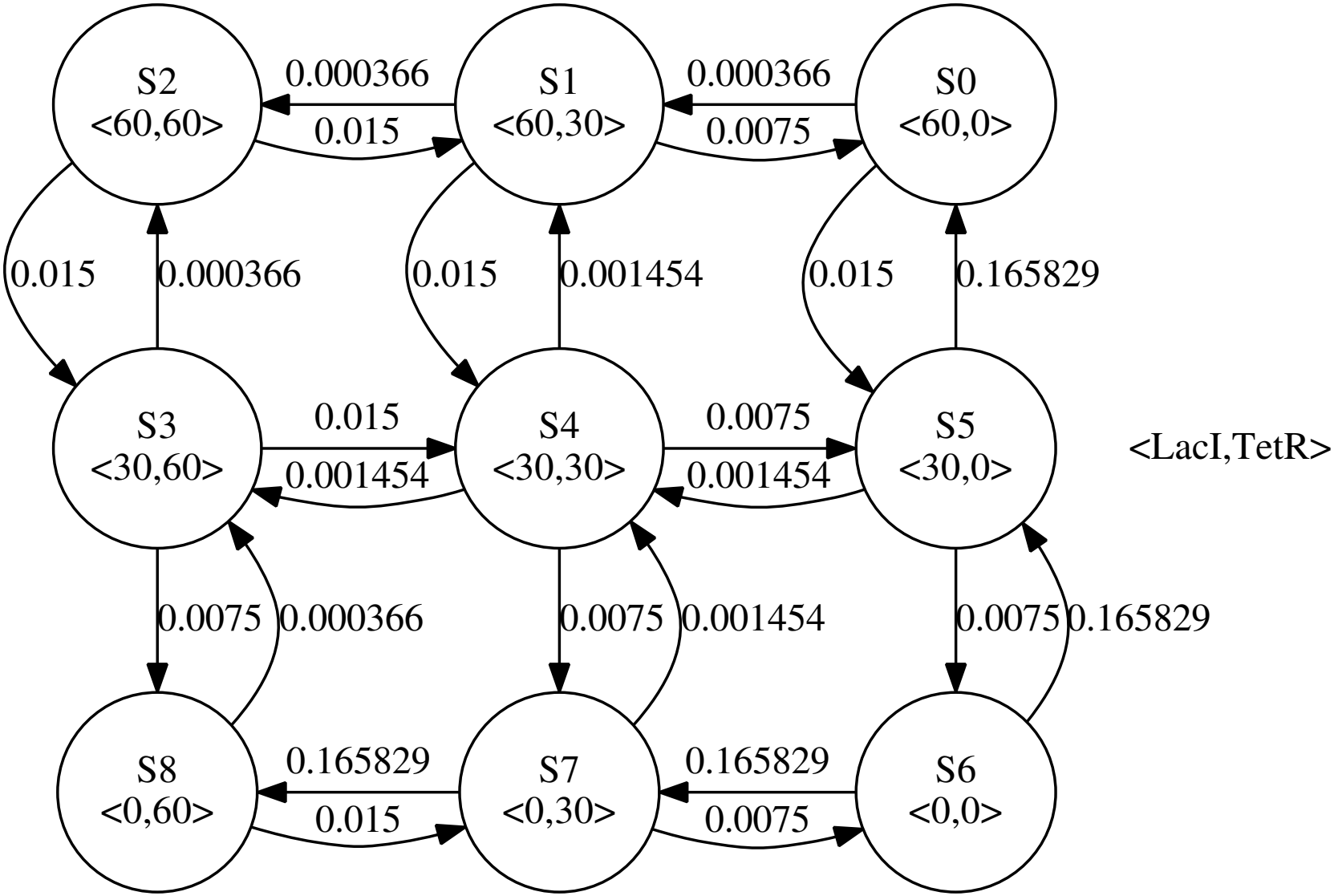
- In order to determine the probability of an interesting event, a property that represents an event occurring in the system must be specified.
- This is accomplished with a subset of *continuous stochastic logic* (CSL):

$$
\begin{aligned}
\textit{Prop} &::= \mathtt{U}(\mathsf{T},\Psi,\Psi) \mid \mathtt{F}(\mathsf{T},\Psi) \mid \mathtt{G}(\mathsf{T},\Psi) \mid \mathtt{St}(\Psi) \\
\Psi &::= \mathtt{true} \mid \Psi \wedge \Psi \mid \neg\Psi \mid \phi \geq \phi \mid \phi > \phi \mid \phi = \phi \\
\phi &::= v_i \mid c_i \mid \phi + \phi \mid \phi - \phi \mid \phi * \phi \mid \phi / \phi \mid \textit{Prop} \\
\mathsf{T} &::= \mathtt{true} \mid \mathsf{T} \wedge \mathsf{T} \mid \neg\mathsf{T} \mid t \geq c_i \mid t > c_i \mid t = c_i
\end{aligned}
$$

- For example, the property, $\mathtt{F}(t \leq 100, \mathrm{LacI} = 0)$, can be used to determine the probability that LacI goes to 0 within 100 seconds.

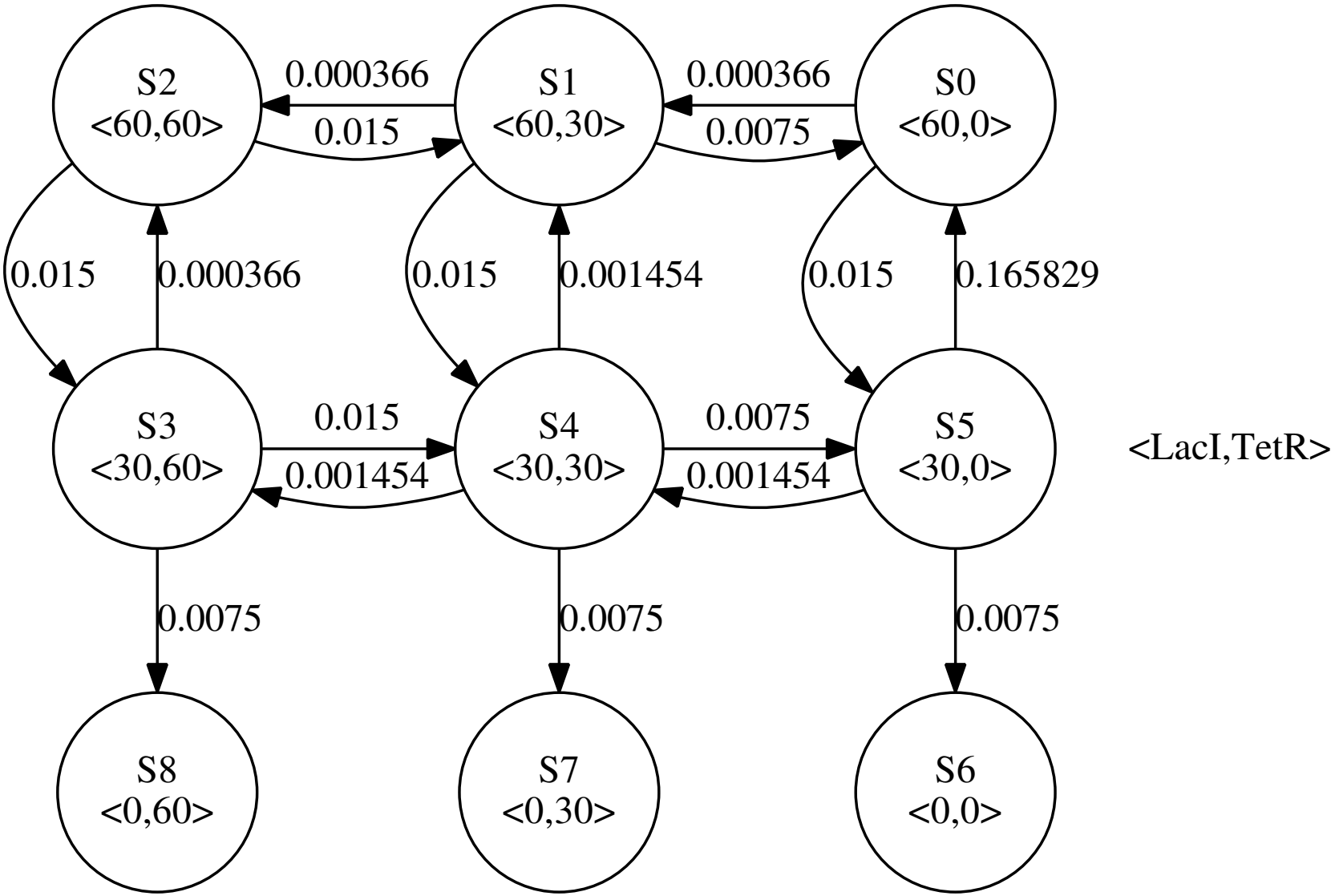
Pruned CTMC for Toggle Switch

$$\mathbb{F}(t \leq 100, \mathsf{LacI} = 0)$$



Pruned CTMC for Toggle Switch

$$\mathbb{F}(t \leq 100, \mathsf{LacI} = 0)$$

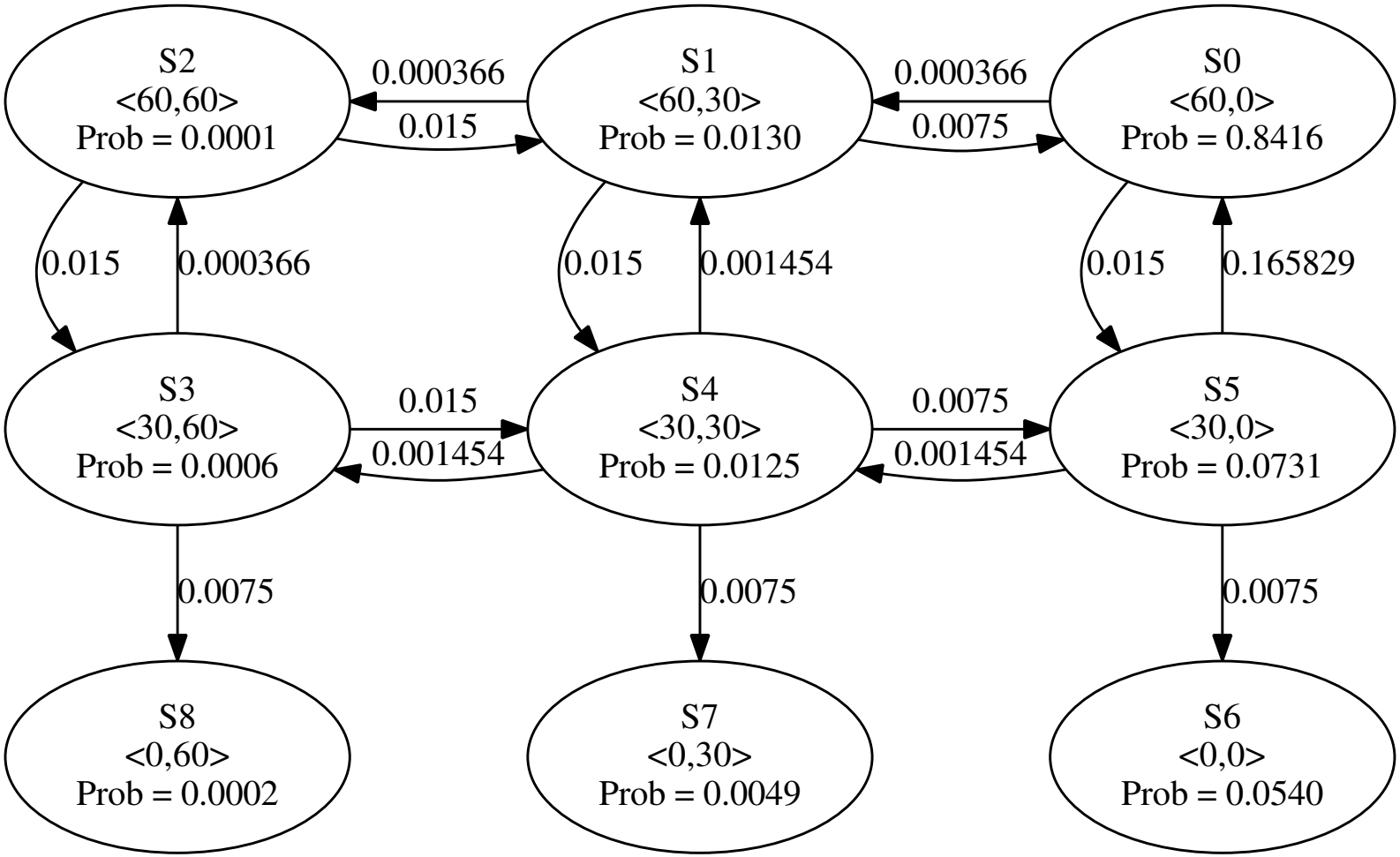


Markov Chain Analysis

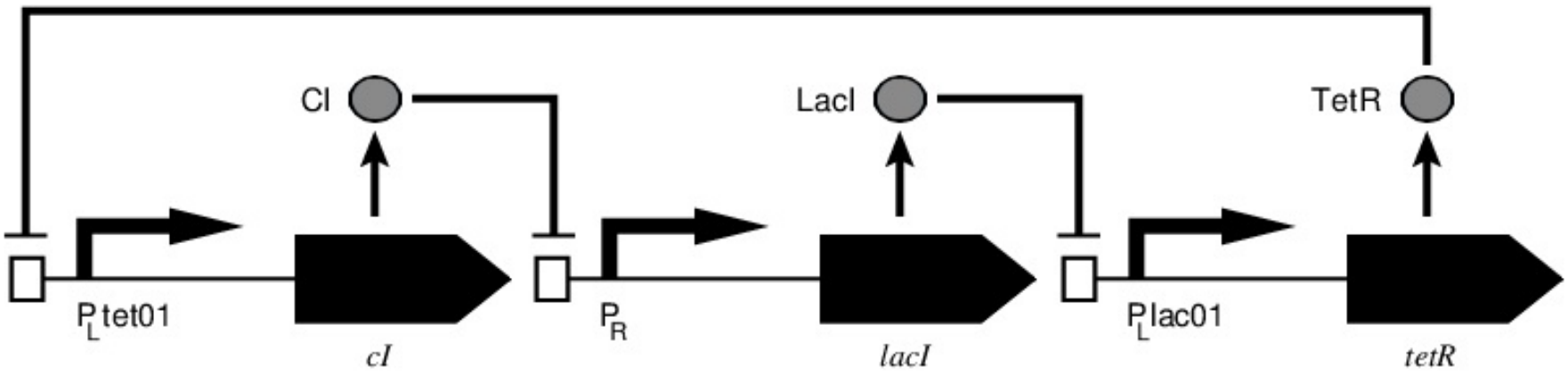
- The CTMC can be analyzed using *steady state Markov chain analysis* or *transient Markov chain analysis*.
- Steady state Markov chain analysis uses the *power iteration* method to compute the invariant distribution of the CTMC.
- Transient Markov chain analysis uses the *uniformization* method to determine the probability of being in each state at a specified time.

Probability of Satisfying the Property

$$\mathbb{F}(t \leq 100, \mathsf{LacI} = 0) \approx 0.0591$$



Repressilator

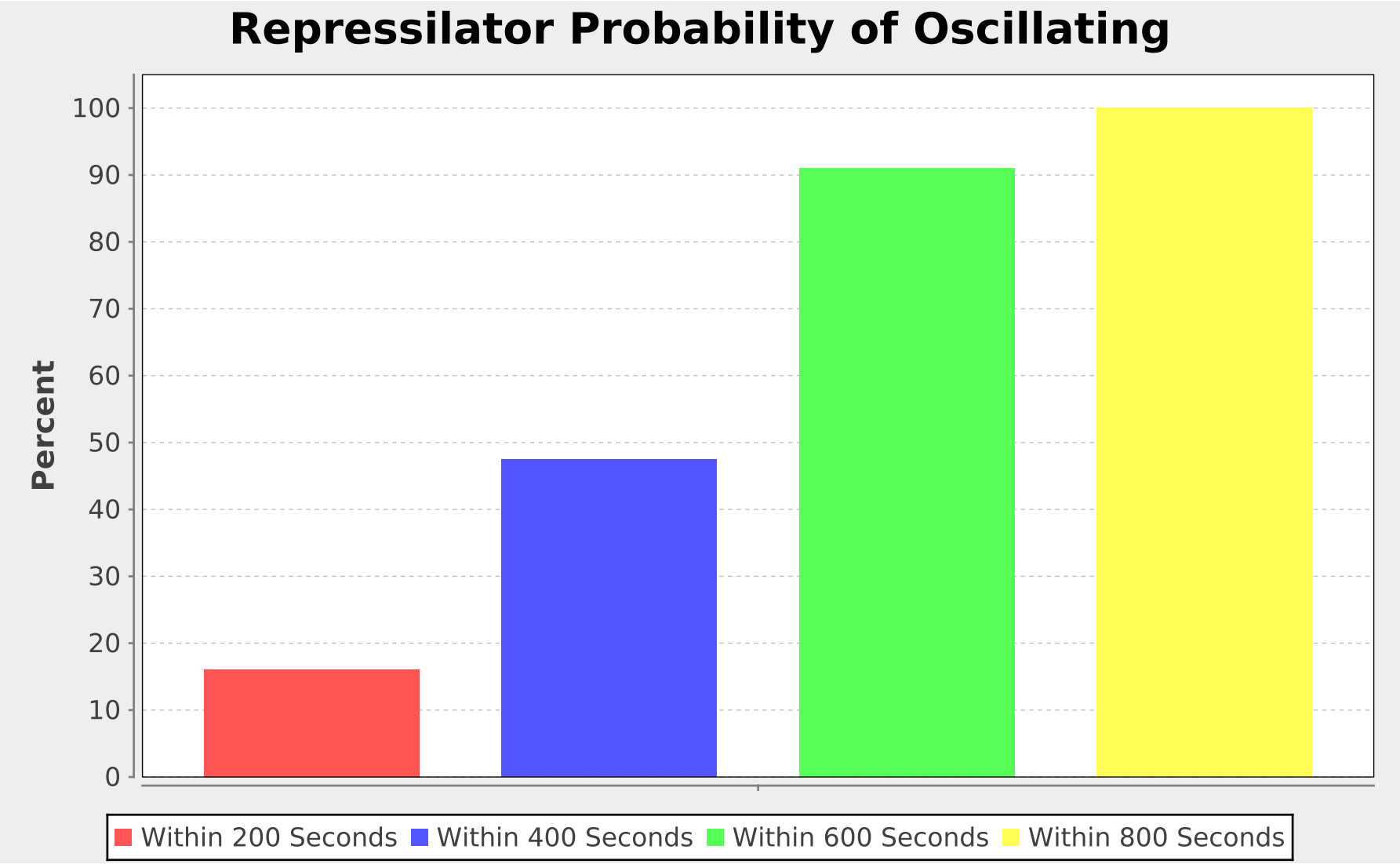


- Applied steady-state analysis to the repressilator using 9 levels evenly spaced between 0 and 80 for CI, LacI, and TetR.
- Determine the likelihood that the value of the CI species is low and goes high or is high and goes low within a predetermined amount of time.

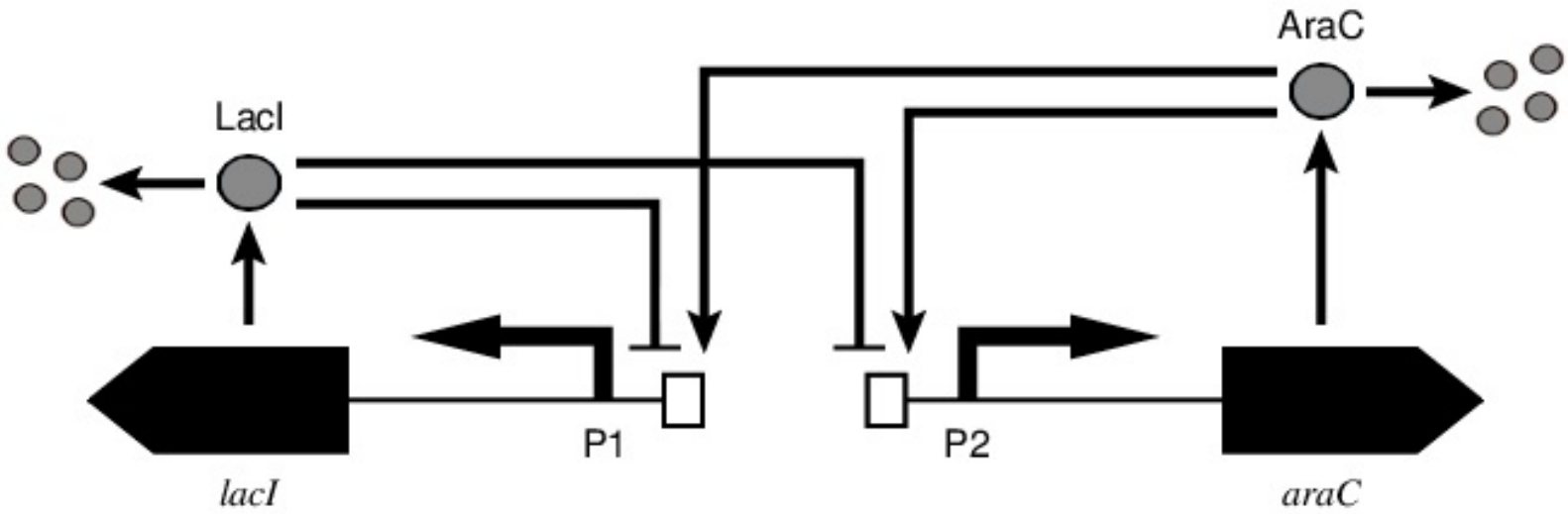
$$\mathtt{St}((\mathsf{CI} \geq 30 \wedge \mathtt{F}(t \leq \text{limit}, \mathsf{CI} < 30) \geq 0.95) \vee$$
$$(\mathsf{CI} < 30 \wedge \mathtt{F}(t \leq \text{limit}, \mathsf{CI} \geq 30) \geq 0.95))$$

- Verification w/STORM (RWTH Aachen) completes in $<$ 1 second.

Repressilator



Dual-feedback Genetic Oscillator

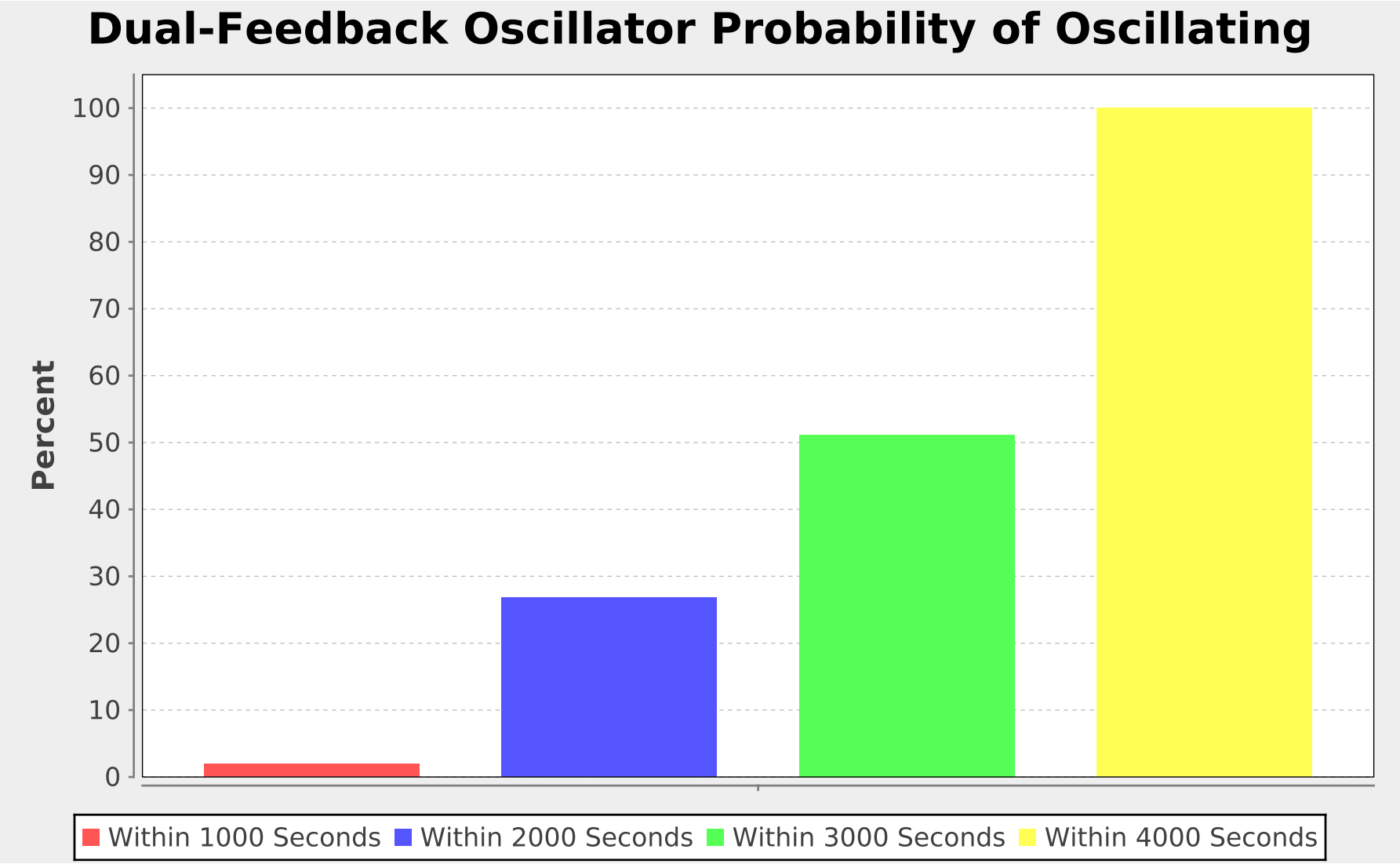


- Applied steady-state analysis to this oscillator using 8 levels evenly spaced between 0 and 120 for AraC and LacI.
- Determine the likelihood that the value of the AraC species is low and goes high or is high and goes low within a predetermined amount of time.

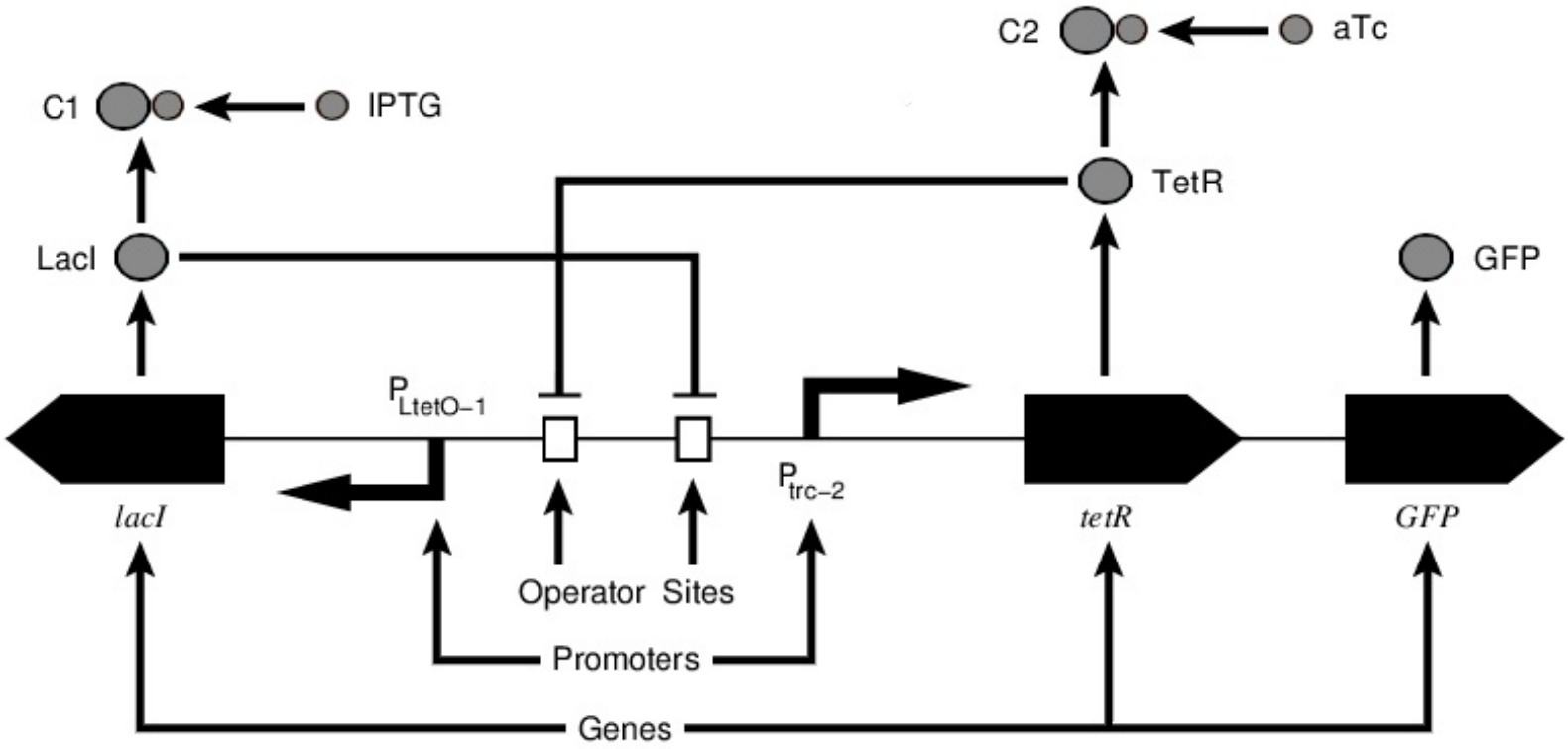
$$\mathtt{St}((\text{AraC} \geq 60 \wedge \mathtt{F}(t \leq \text{limit}, \text{AraC} < 60) \geq 0.95) \vee$$
$$(\text{AraC} < 60 \wedge \mathtt{F}(t \leq \text{limit}, \text{AraC} \geq 60) \geq 0.95))$$

- Verification w/STORM (RWTH Aachen) completes in $<$ 1 second.

Dual-feedback Genetic Oscillator



Genetic Toggle Switch Failure Rate

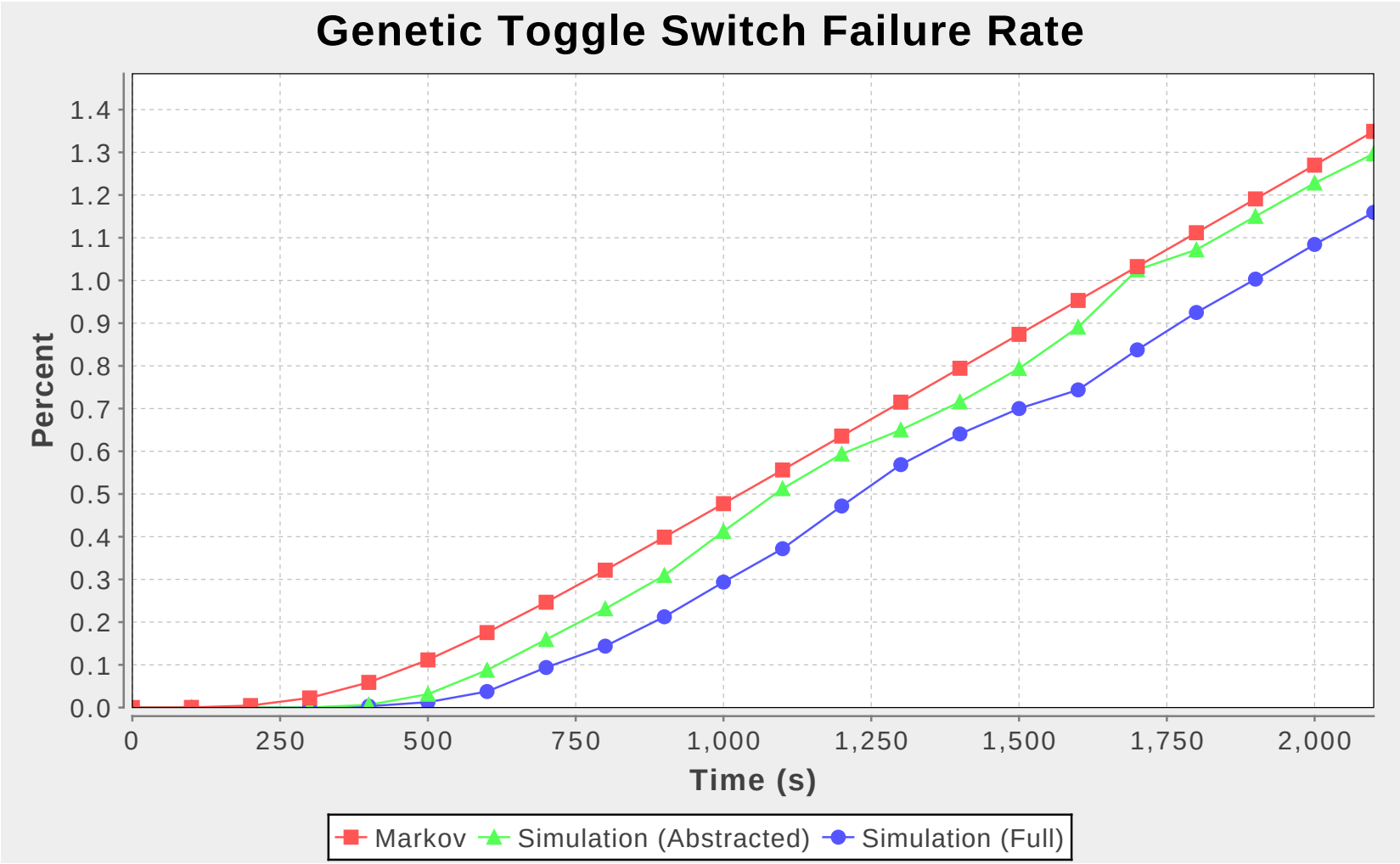


- Applied transient analysis using 9 levels for LacI between 0 and 80, and 11 levels for TetR between 0 and 50.
- LacI set to 60, TetR set to 0, IPTG set to 0, and aTc set to 0.

$$\mathbb{F}(t \leq 2100, \text{LacI} < 20 \wedge \text{TetR} > 40)$$

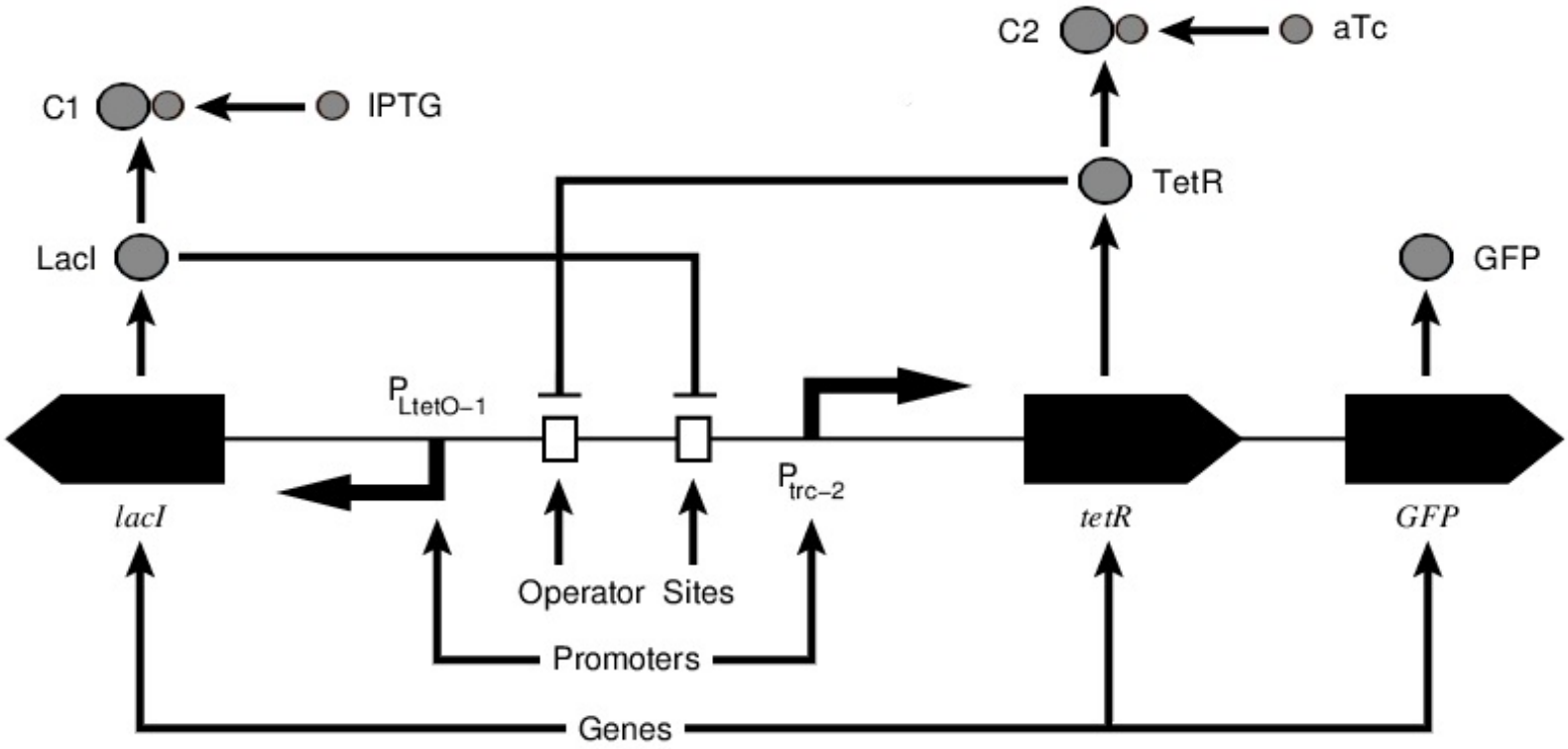
Genetic Toggle Switch Failure Rate

$$\mathbb{F}(t \leq 2100, \mathrm{LacI} < 20 \wedge \mathrm{TetR} > 40)$$



Simulation time: 43 min. (Full), 3 min. 15 sec. (Abstracted), <1 sec. (Markov).

Genetic Toggle Switch Response Time

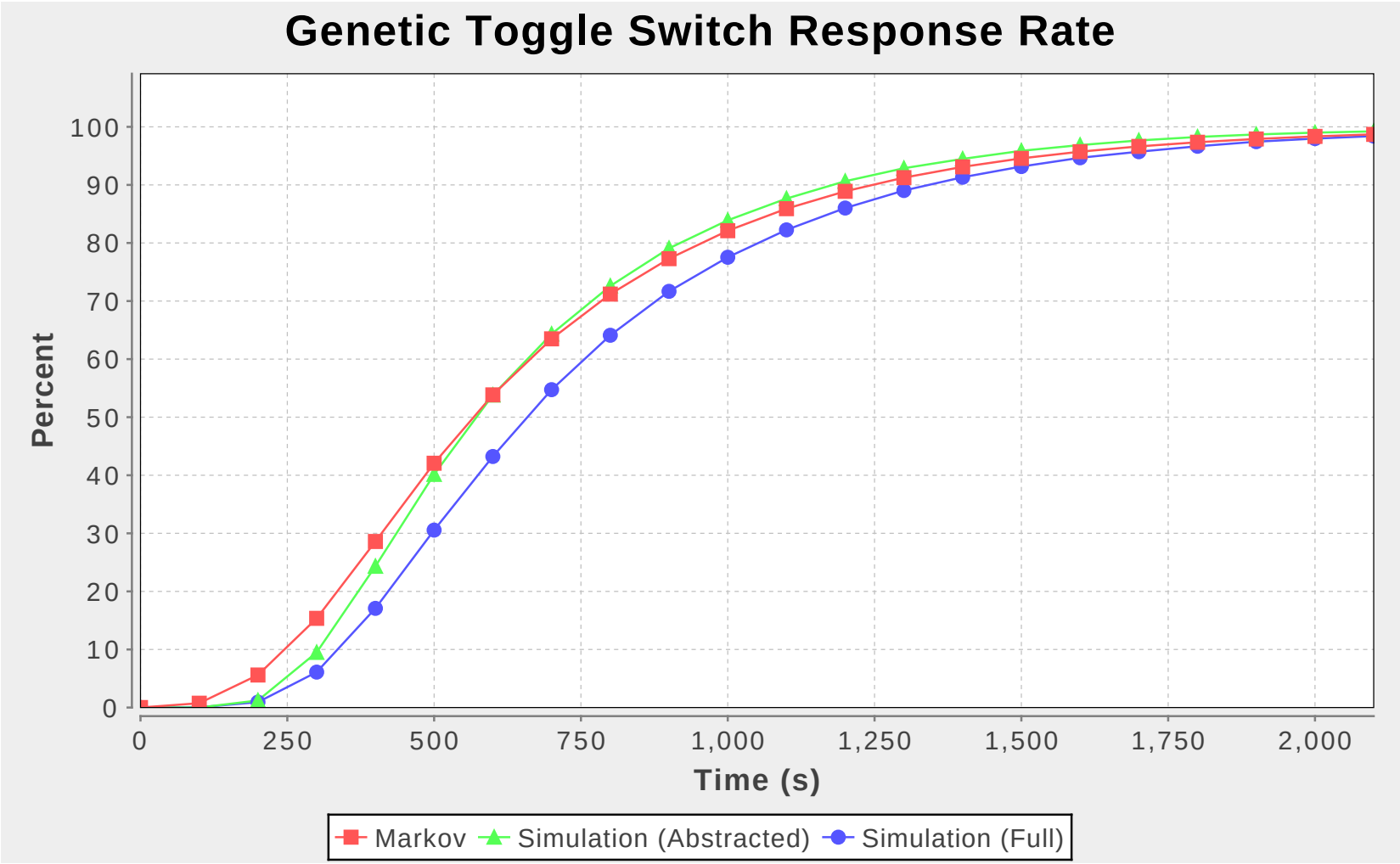


- Applied transient analysis using 14 levels for LacI between 0 and 130, and 5 levels for TetR between 0 and 60.
- LacI set to 60, TetR set to 0, IPTG set to 100, and aTc set to 0.

$$\mathbb{F}(t \leq 2100, \mathrm{LacI} < 20 \wedge \mathrm{TetR} > 40)$$

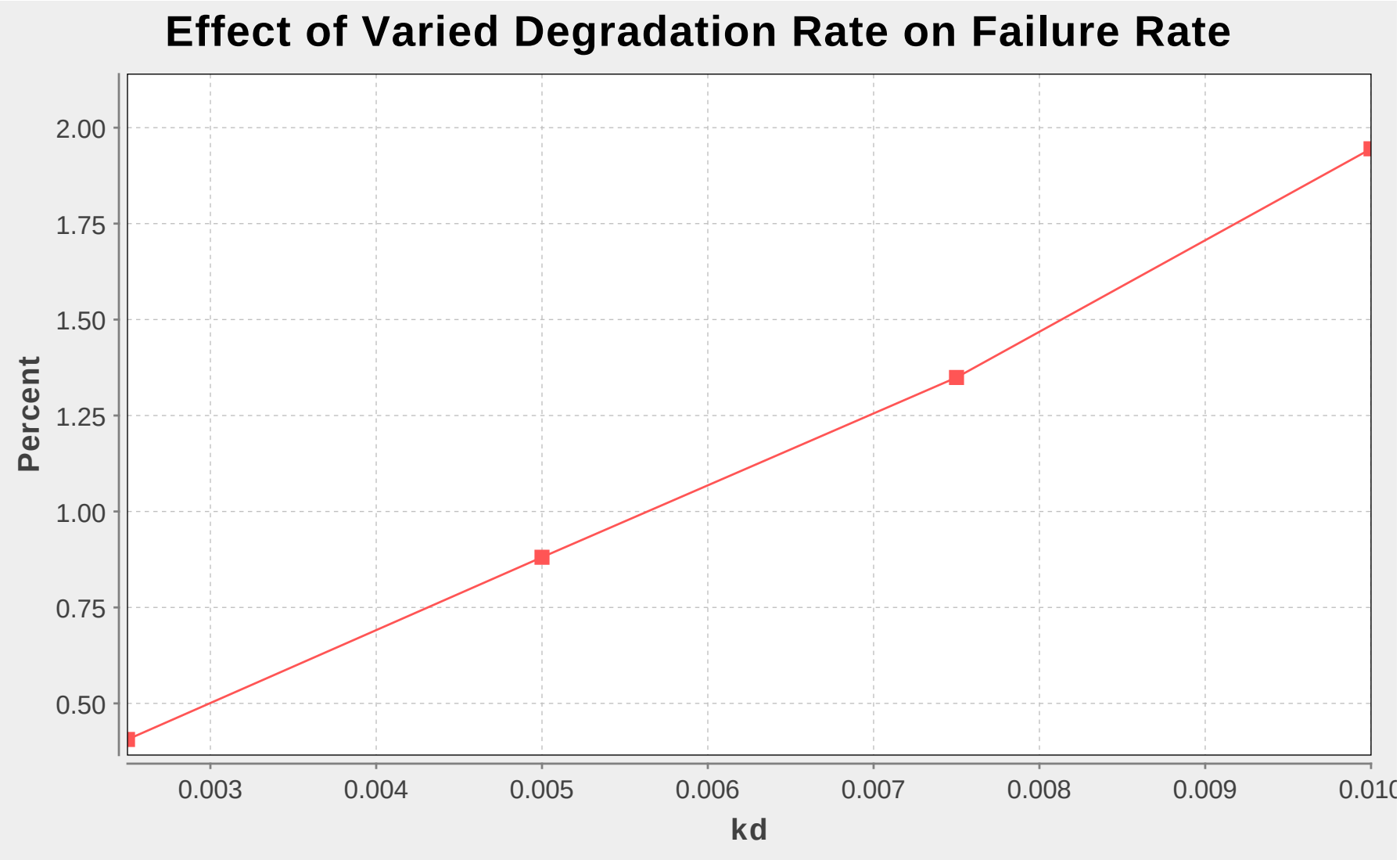
Genetic Toggle Switch Response Time

$$\mathbb{F}(t \leq 2100, \mathrm{LacI} < 20 \wedge \mathrm{TetR} > 40)$$

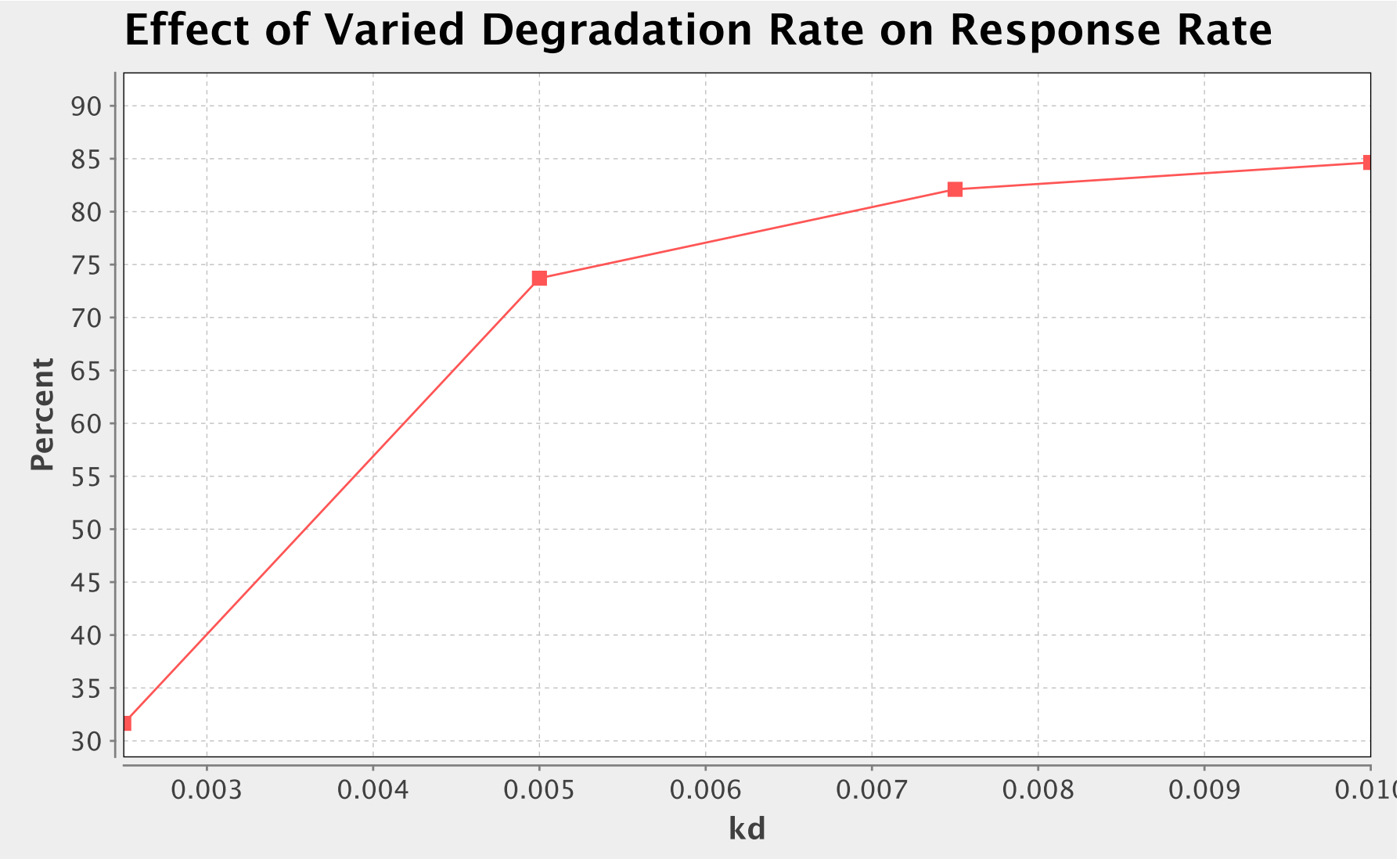


Simulation time: 3 hours 12 min. (Full), 1 min. (Abstracted), 0.5 sec. (Markov).

Failure Rate Versus Degradation Rate



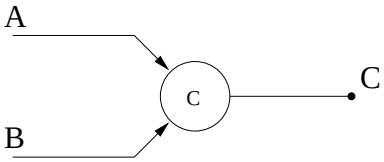
Response Time Versus Degradation Rate



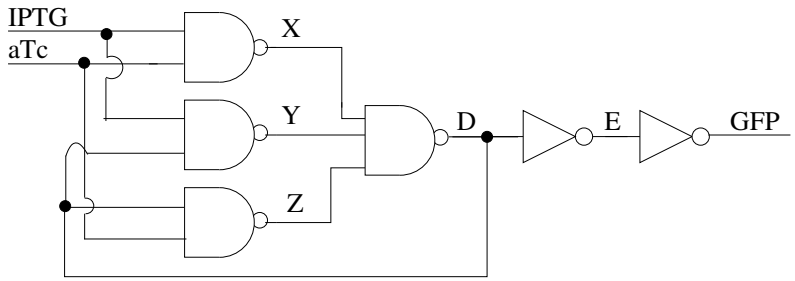
Toggle Switch Failure with Diffusion

(Loading ToggleFailSimDiff.mov)

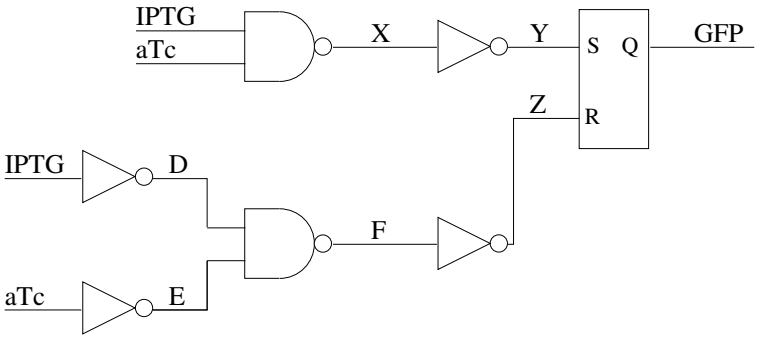
Genetic Muller C-Element



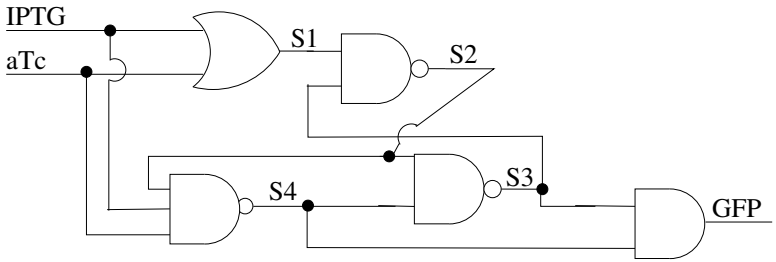
A	B	C
0	0	0
0	1	C
1	0	C
1	1	1



Majority Gate C-element



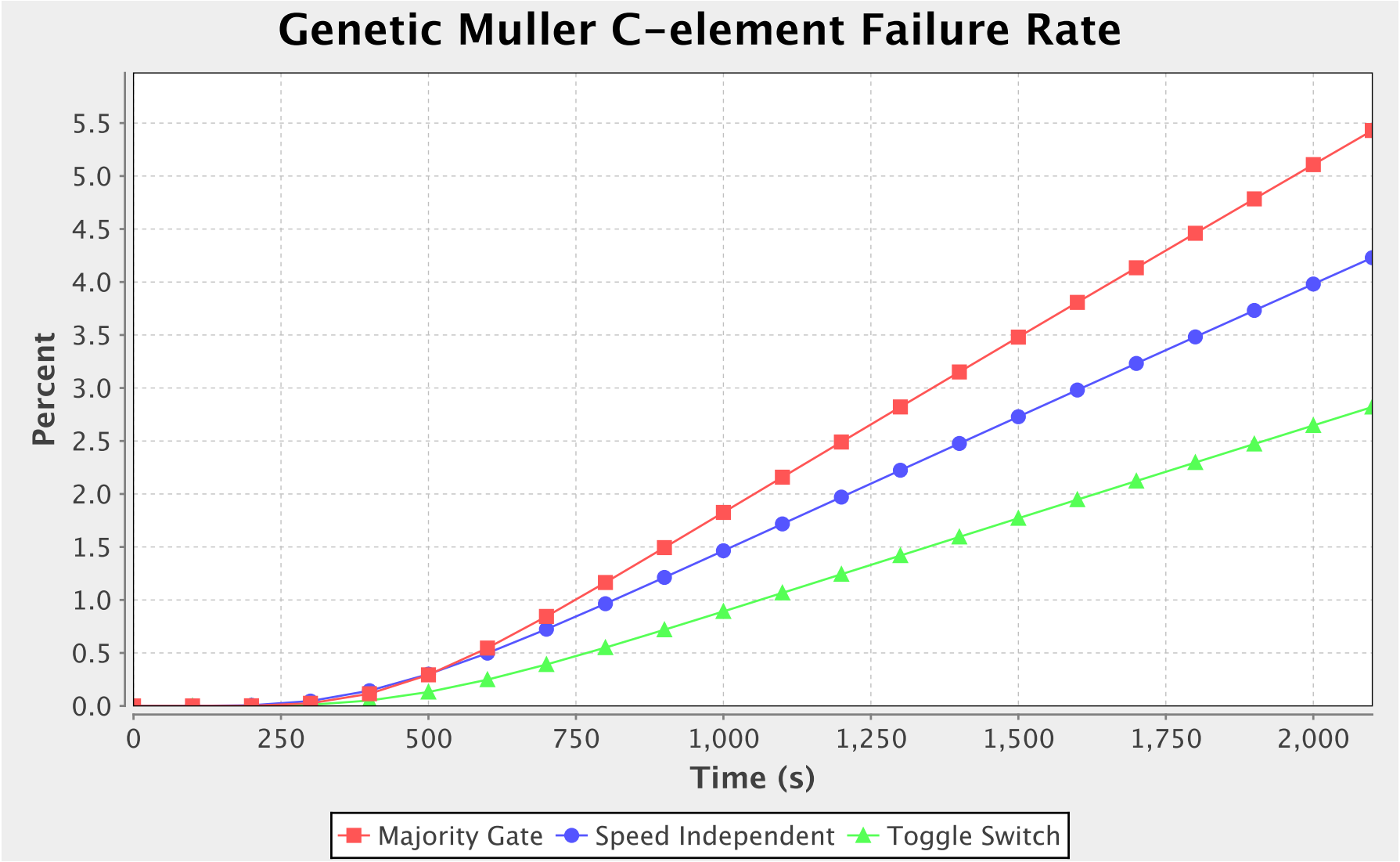
Toggle Switch C-element



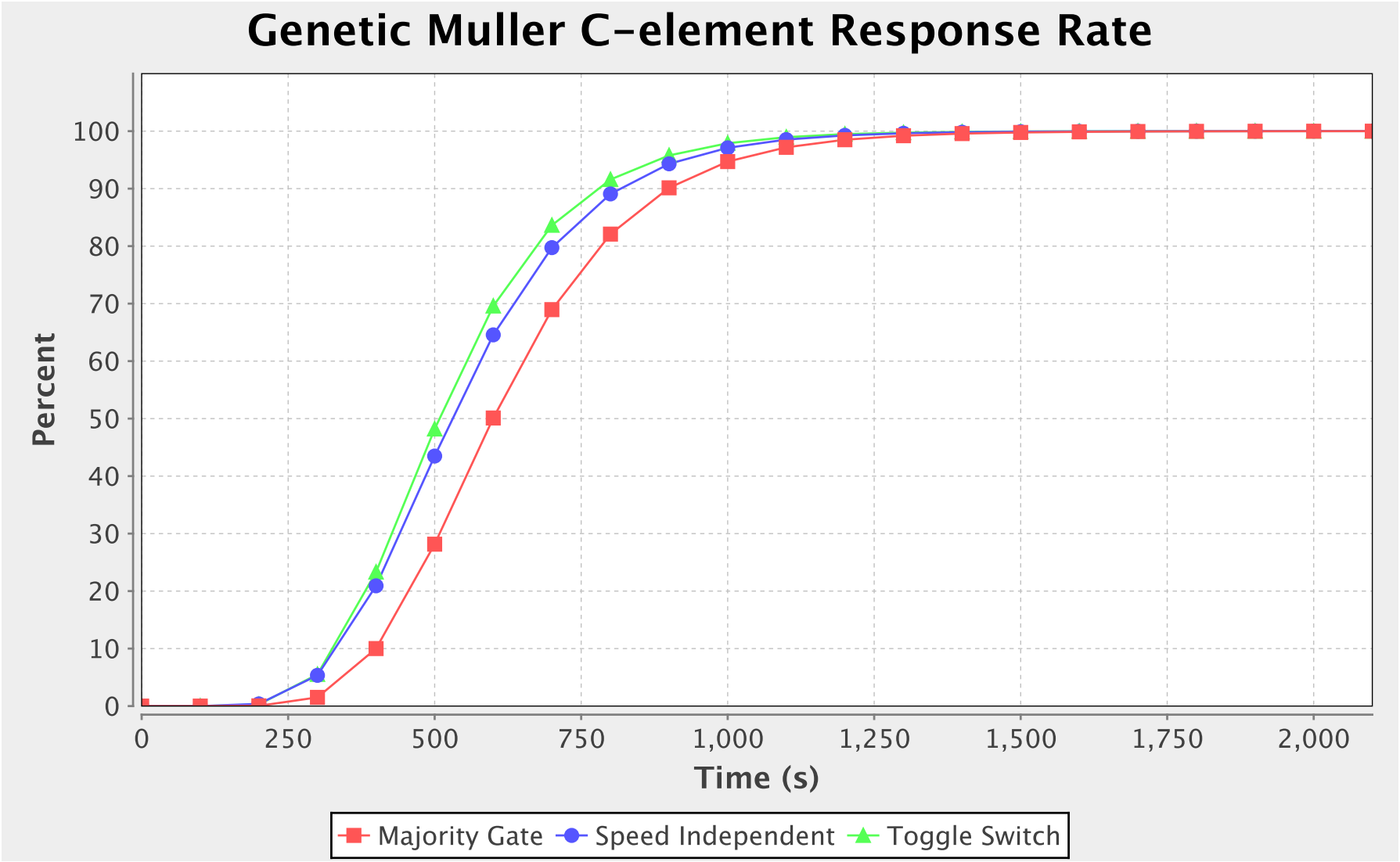
Speed-independent C-element

Nguyen et al., 13th Symposium on Async. Ckts. & Sys., 2007 (**best paper**)
Nguyen et al., Journal of Theoretical Biology (2010)

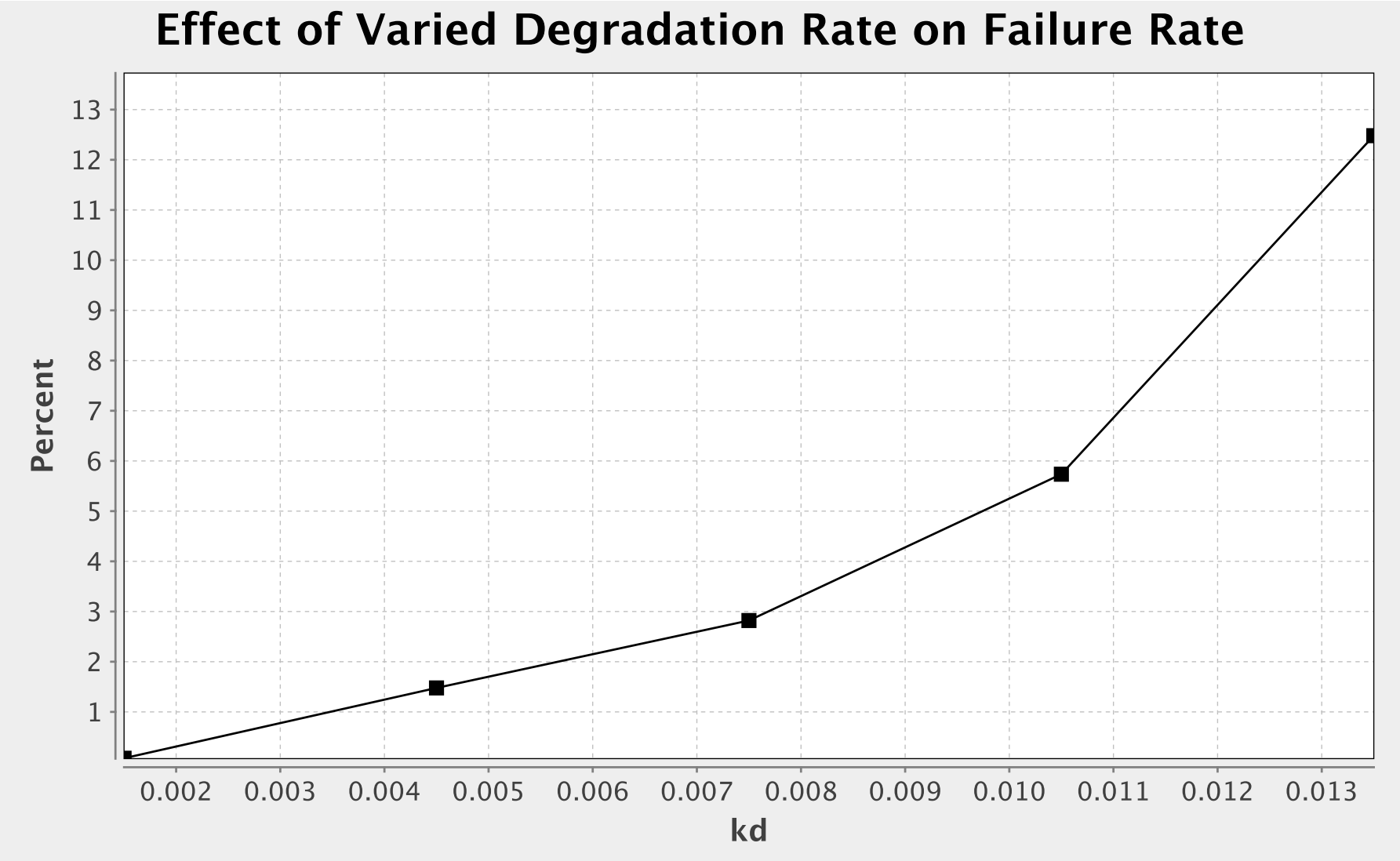
Genetic Muller C-element Failure Rate



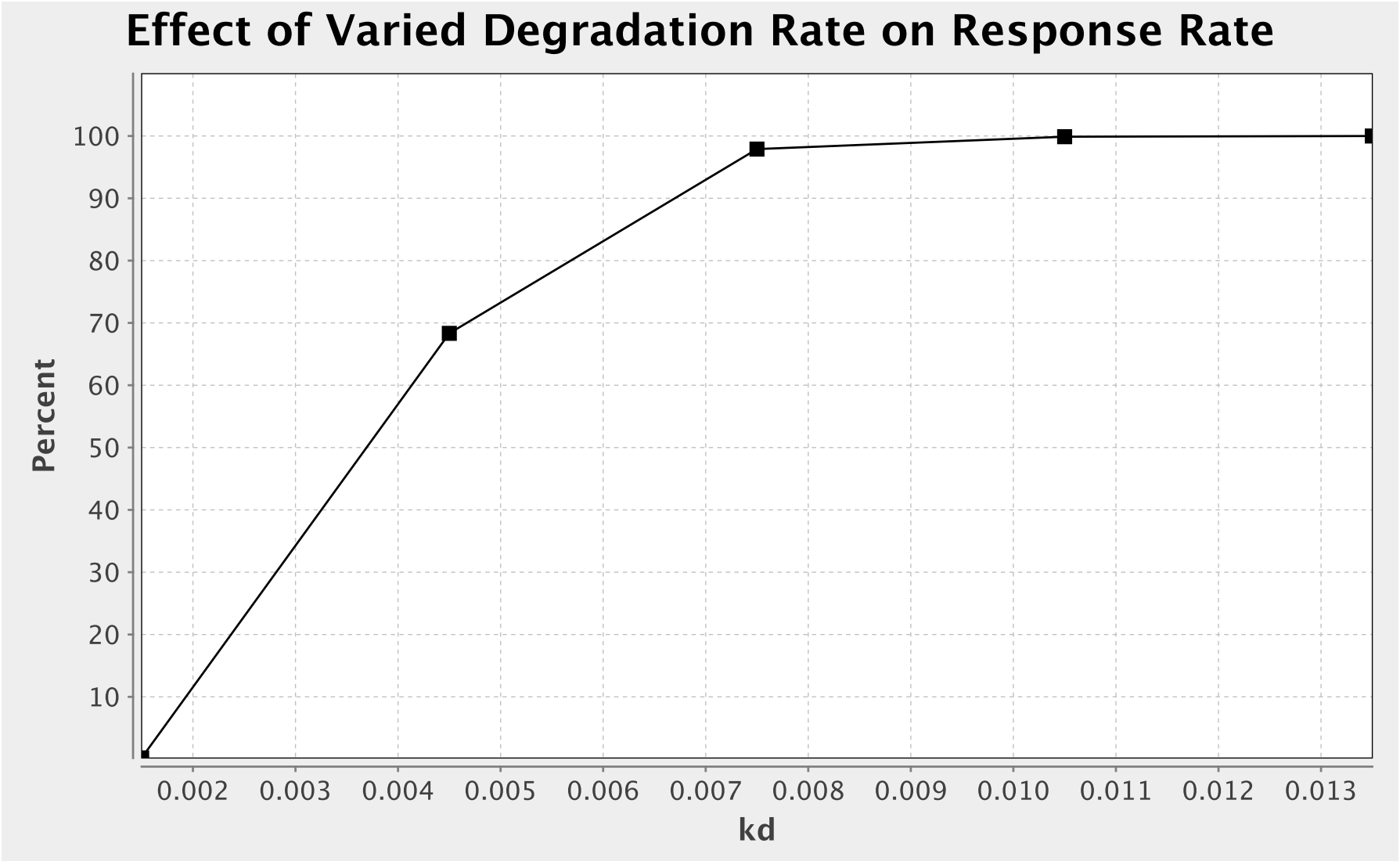
Genetic Muller C-element Response Time



Failure Rate Versus Degradation Rate

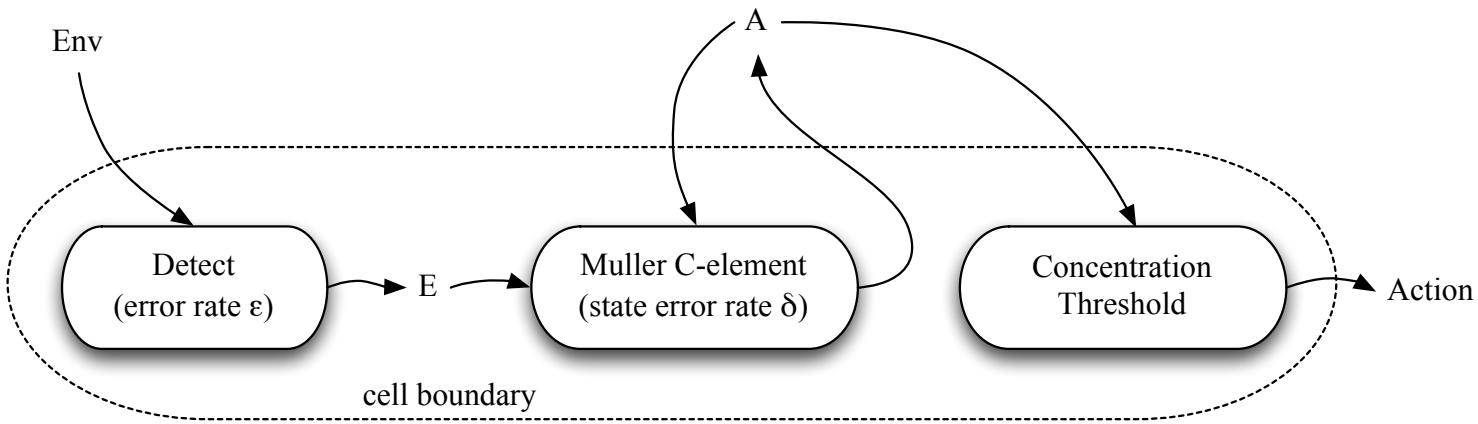


Response Time Versus Degradation Rate



Application: Bacterial Consensus

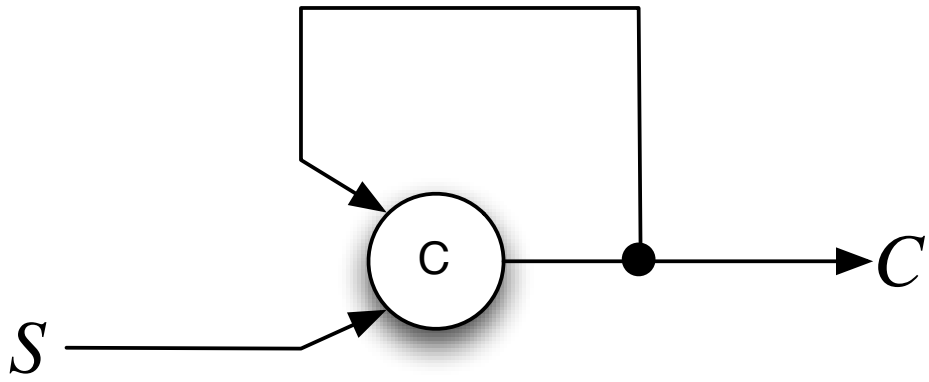
- One interesting application is designing bacteria that can hunt and kill tumor cells (Anderson et al.).
- Care must be taken in determining when to attack potential tumor cells.
- Can use a genetic Muller C-element and a bacterial consensus mechanism known as *quorum sensing*.
- C-element combines a noisy environmental trigger signal and a density dependent quorum sensing signal.
- Activated bacteria signal their neighbors to reach consensus.



Winstead et al., IBE Conference (2008)

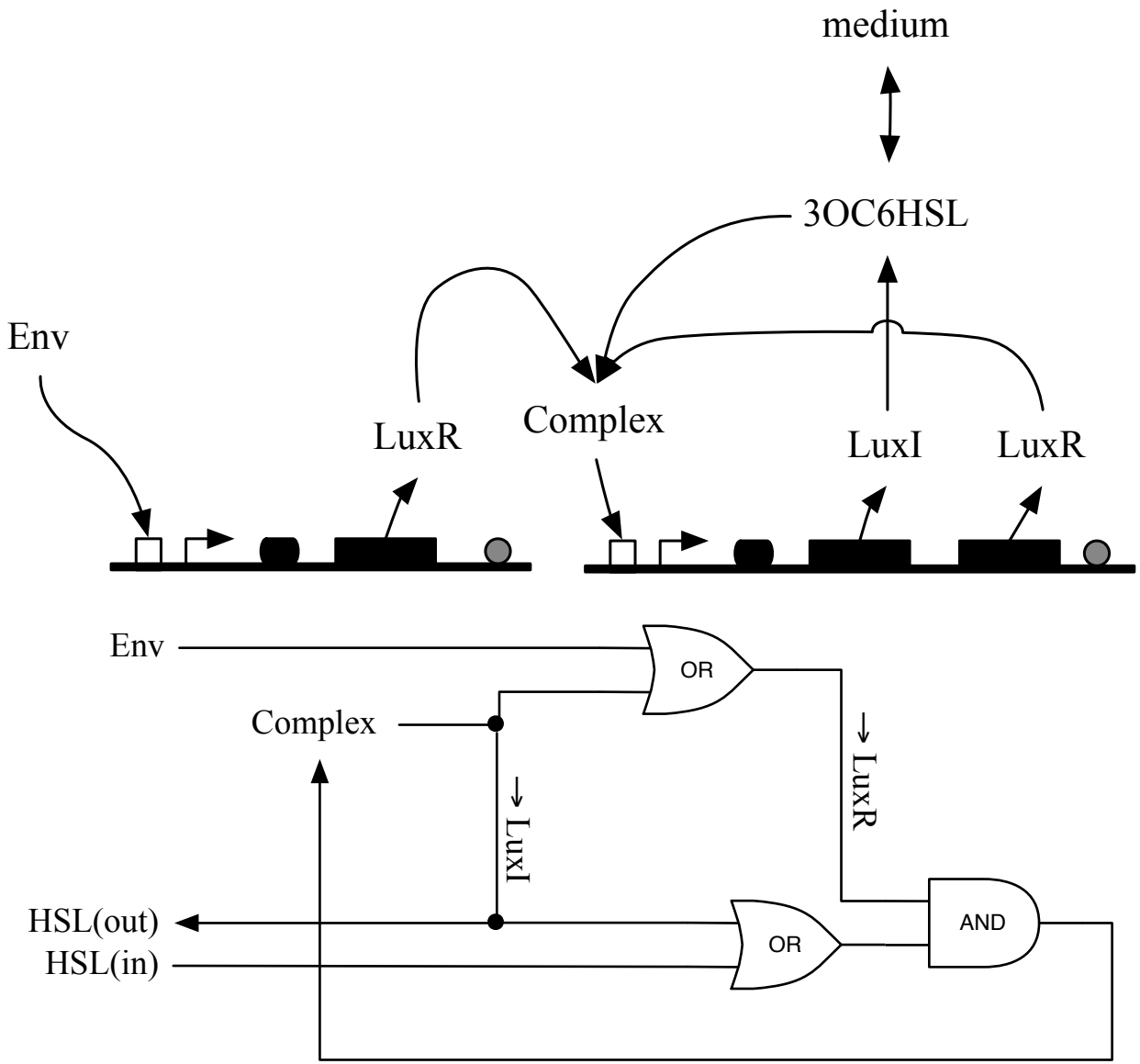
Confidence Amplifier

- A noisy C-element with a confidence-feedback loop:



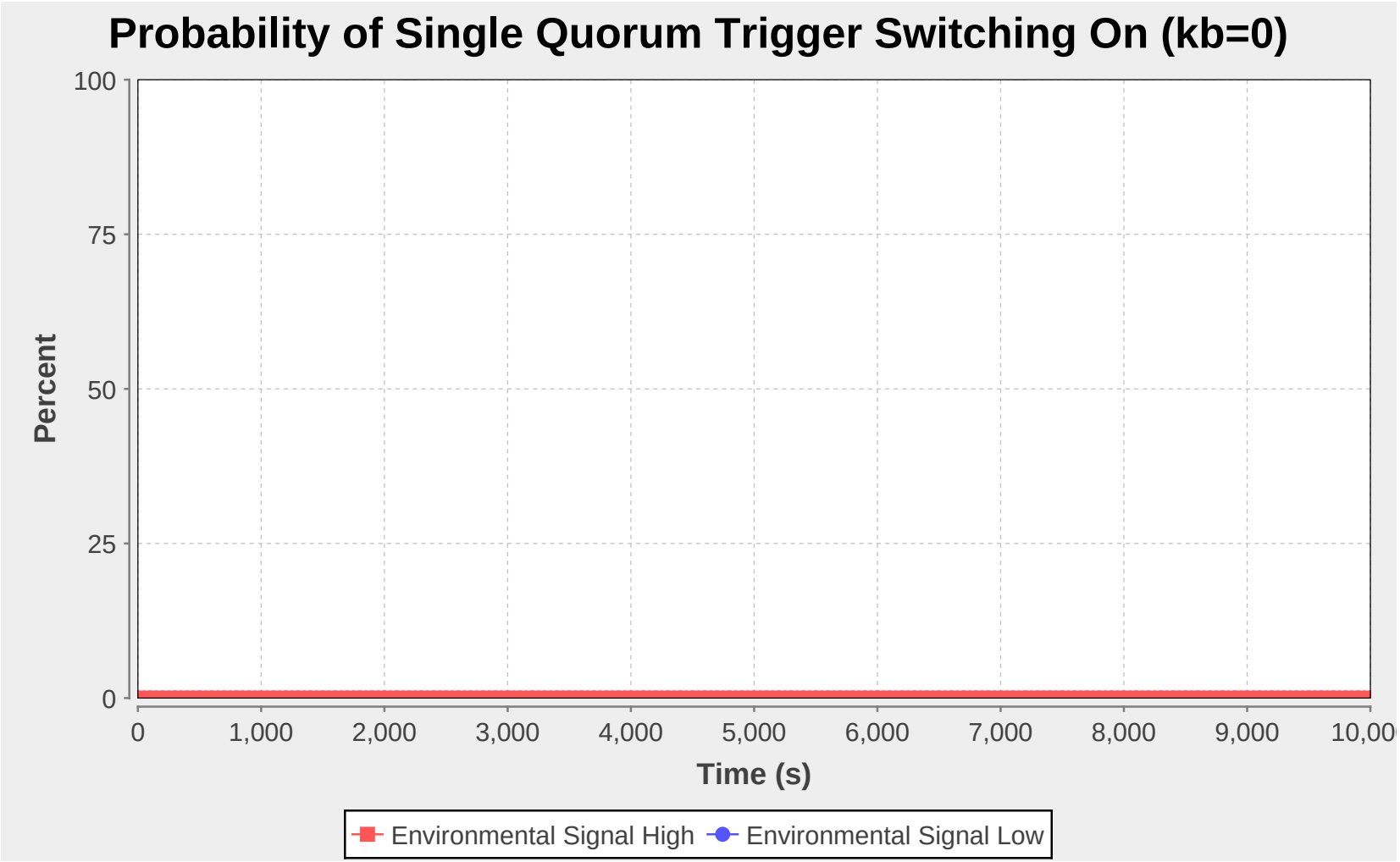
- The output “rails” to maximum confidence, even if S has low confidence.
- This configuration only works if the C-element is “noisy”. Otherwise, the circuit is permanently stuck in its initial state.

Quorum Trigger Circuit



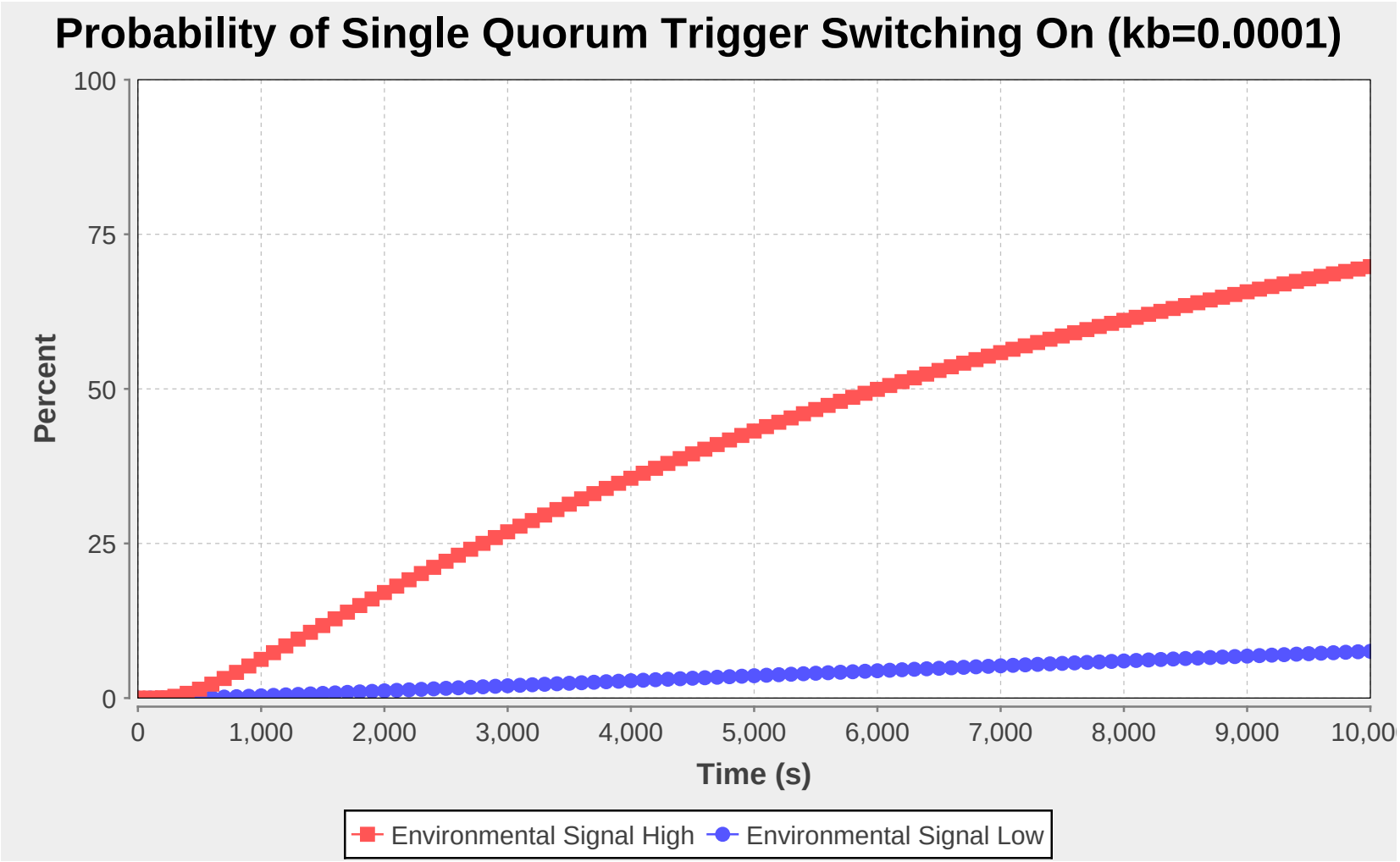
Quorum Trigger Simulation Results

$$\mathbb{F}(t \leq 10000, \mathrm{GFP} > 300)$$



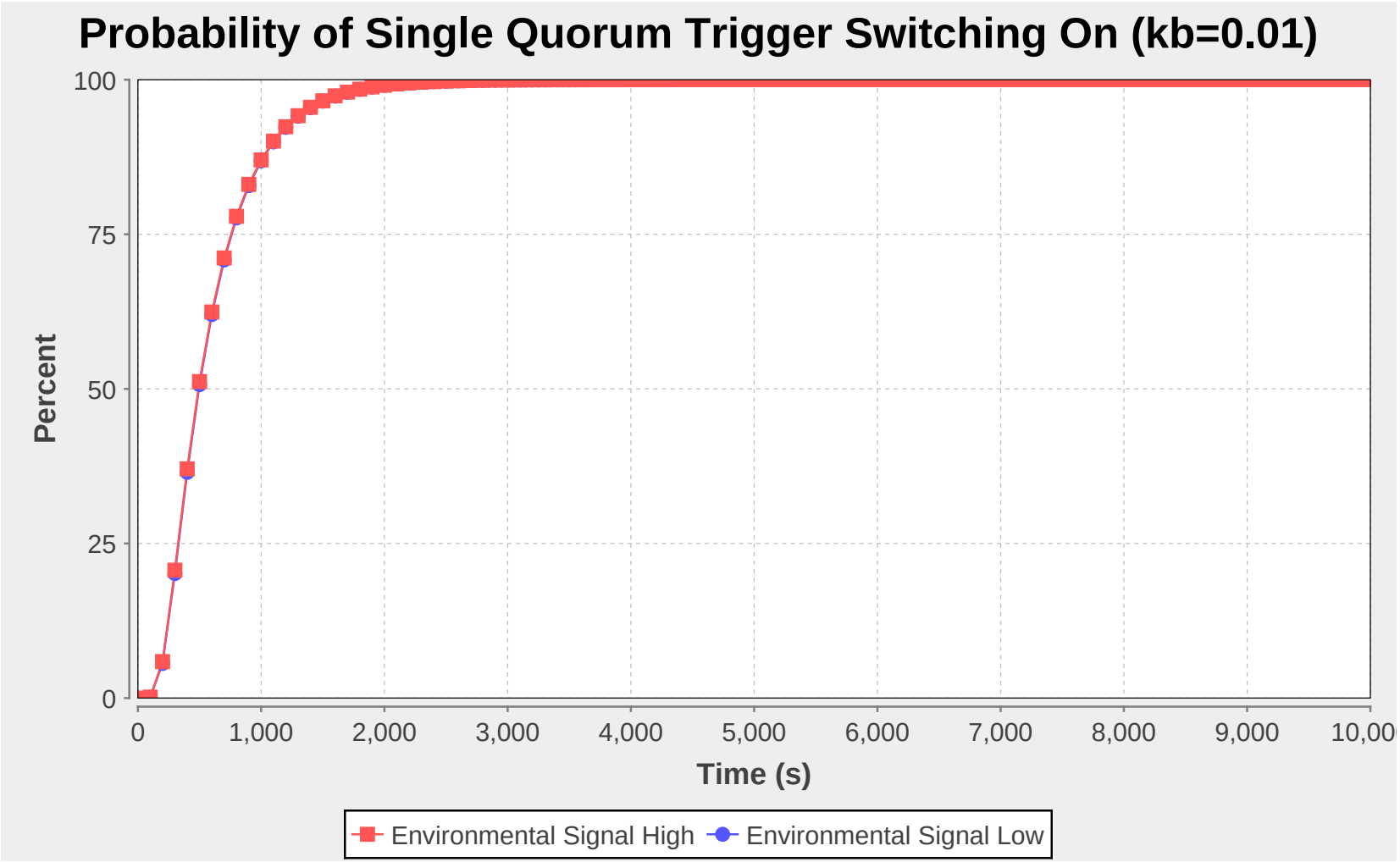
Quorum Trigger Simulation Results

$$\mathbb{F}(t \leq 10000, \mathrm{GFP} > 300)$$



Quorum Trigger Simulation Results

$$\mathbb{F}(t \leq 10000, \text{GFP} > 300)$$



Qualitative Logical Models

- Models typically require the estimation of binding affinities and kinetic parameters which are often difficult to obtain for genetic circuits.
- Systems, however, are often quite robust to parameter variation.
- May be possible to make reasonable behavioral predictions with only qualitative information.
- A *qualitative logical model* is similar to the stochastic FSM model except that no rate parameters are provided.
- Note that while a stochastic FSM may potentially enter any state, it would not be informative if a qualitative logical model also could reach any state.
- Qualitative logical models only describe the most likely states and state transitions.

Qualitative Logical Model for CI/CI Portion of Phage λ

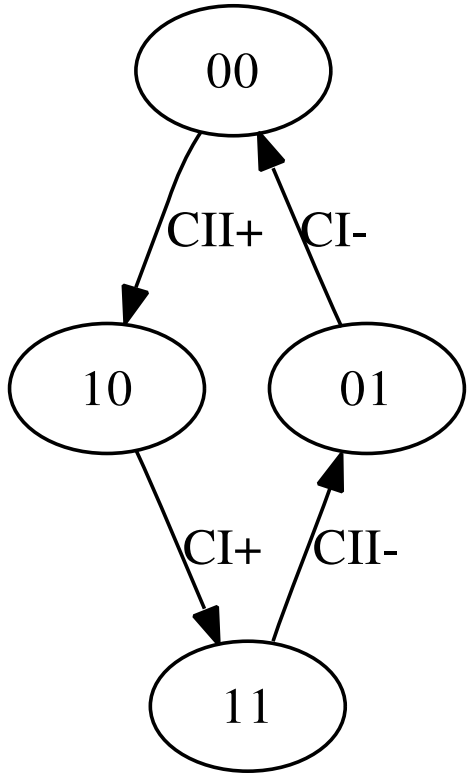
- Qualitative logical model for the CI/CII portion of the phage λ:

$$\begin{aligned} b_{CII} &:= (b_{CI} = 0) \\ b_{CI} &:= (b_{CII} = 1) \end{aligned}$$

  where CI and CII are binary encoded variables.

- In order to analyze a qualitative logical model, one first finds all reachable states using a depth first search assuming some initial state.

State Graph for the CI/CII Portion of the Phage λ Model



- Note that this represents only the most likely scenario as it is potentially possible that through basal production of CI, that CI goes to a high concentration before CII increases.

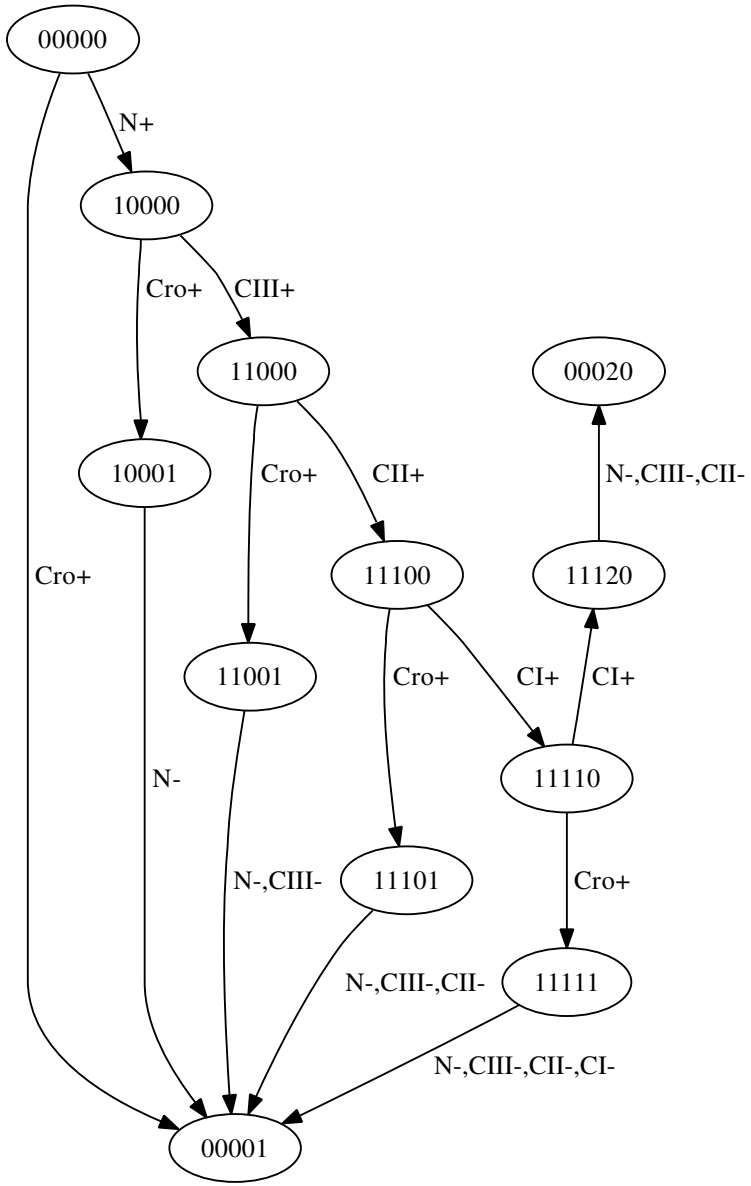
Qualitative Logical Model for the Complete Phage λ

- The full phage λ decision circuit can be represented logically as follows:

$$
\begin{aligned}
b_N &:= (b_{CI} \neq 2) \wedge (b_{Cro} = 0) \\
b_{CIII} &:= (b_N = 1) \wedge (b_{CI} \neq 2) \wedge (b_{Cro} = 0) \\
b_{CII} &:= (b_N = 1) \wedge (b_{CIII} = 1) \wedge (b_{CI} \neq 2) \wedge (b_{Cro} = 0) \\
b_{CI} &:= (((b_{CII} = 1) \wedge (b_{CI} = 0)) + 2(b_{CI} \neq 0)) \wedge (b_{Cro} = 0) \\
b_{Cro} &= (b_{CI} \neq 2)
\end{aligned}
$$

where N, CII, CIII, and Cro are binary and CI is a ternary variable.

Partial State Graph for the Complete Phage λ Model



Sources

- Relation of electrical and biochemical circuits:
  - Arkin (2000).
- Piecewise models:
  - Glass (1977), Plahte et al. (1994), and Gouze (2001).
  - Phage λ - McAdams and Shapiro (1995).
  - Hybrid Petri nets - Matsuno et al. (2000) and Little et al. (2006).
- Stochastic FSMs:
  - Kuwahara et al. (2006) and Kuwahara (2007).
- Markov Chains:
  - Stewart (1994).
- Qualitative logical models:
  - Boolean logical models - Kauffman (1969) and Kauffman et al. (1978).
  - Generalized logical models - Thomas (1991) and Thieffry/Thomas (1995).
  - Qualitative differential equations - DeJong et al. (2001).
- Chapter 6 of Engineering Genetic Circuits - Myers (2009).