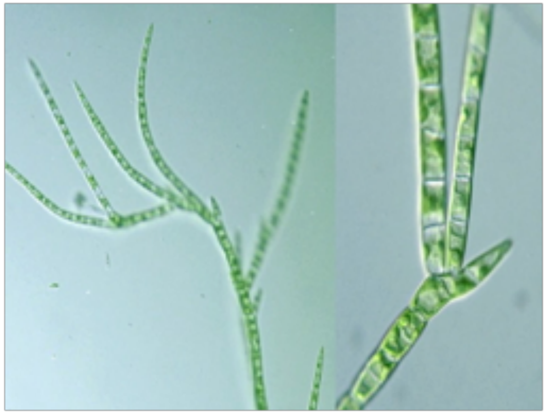
Module 10: Finite State Machines with Gene Regulatory Networks

CSE 590: Molecular programming and neural computation

Guest Lecture: Kevin Oishi

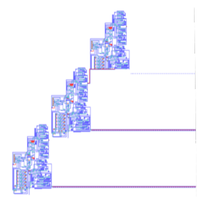
Living Systems Perform Discrete Computation

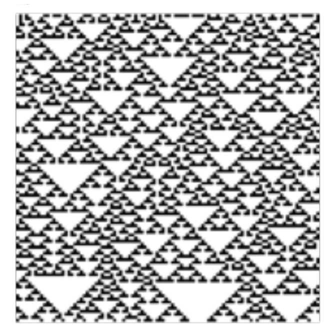


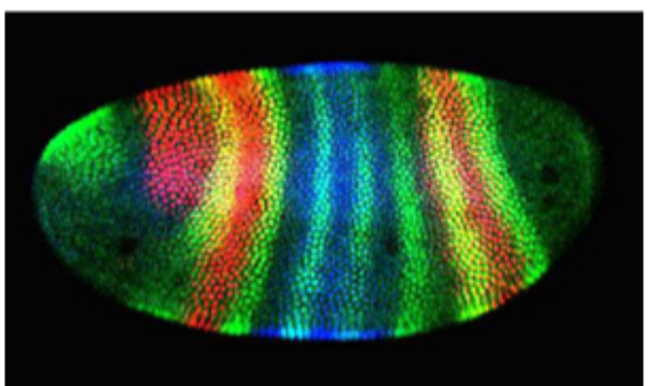
Lindenmeyer Systems
(1968)

Variables: A,B
Start: A
Rules: A→AB, B→A

Cellular Automata (1940s)
Conway's Game of Life

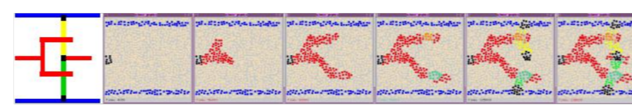


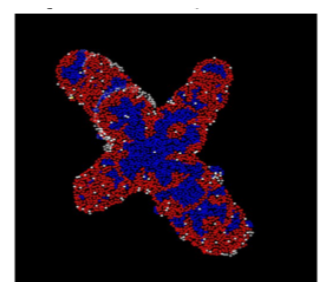


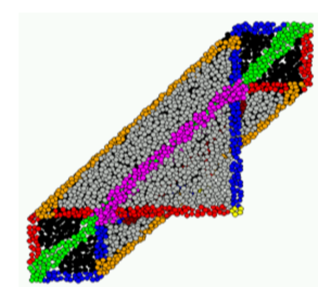


Amorphous Computing (1996)

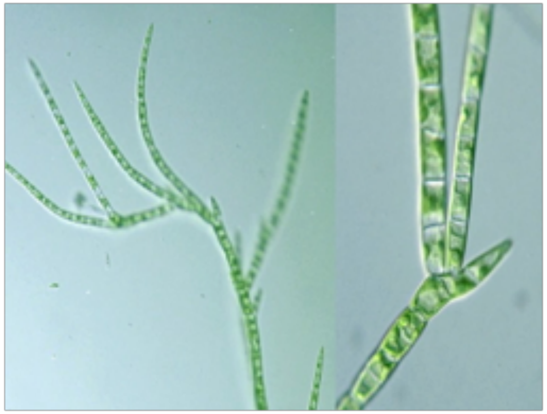
- Growing Point Language
- Origami Shape Language
- Morphogenesis Language





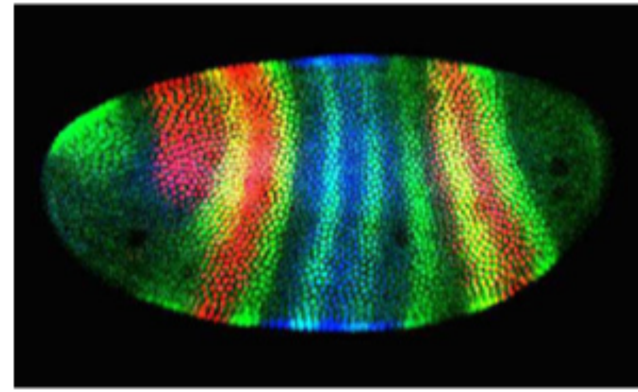


Motivation

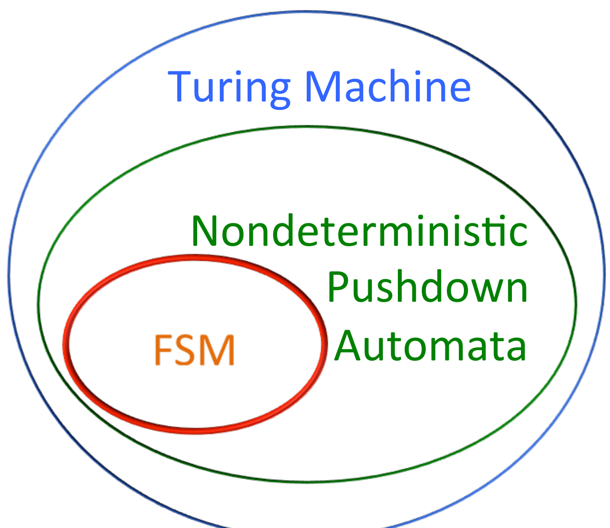


Nondeterministic pushdown automata

Cellular Automata

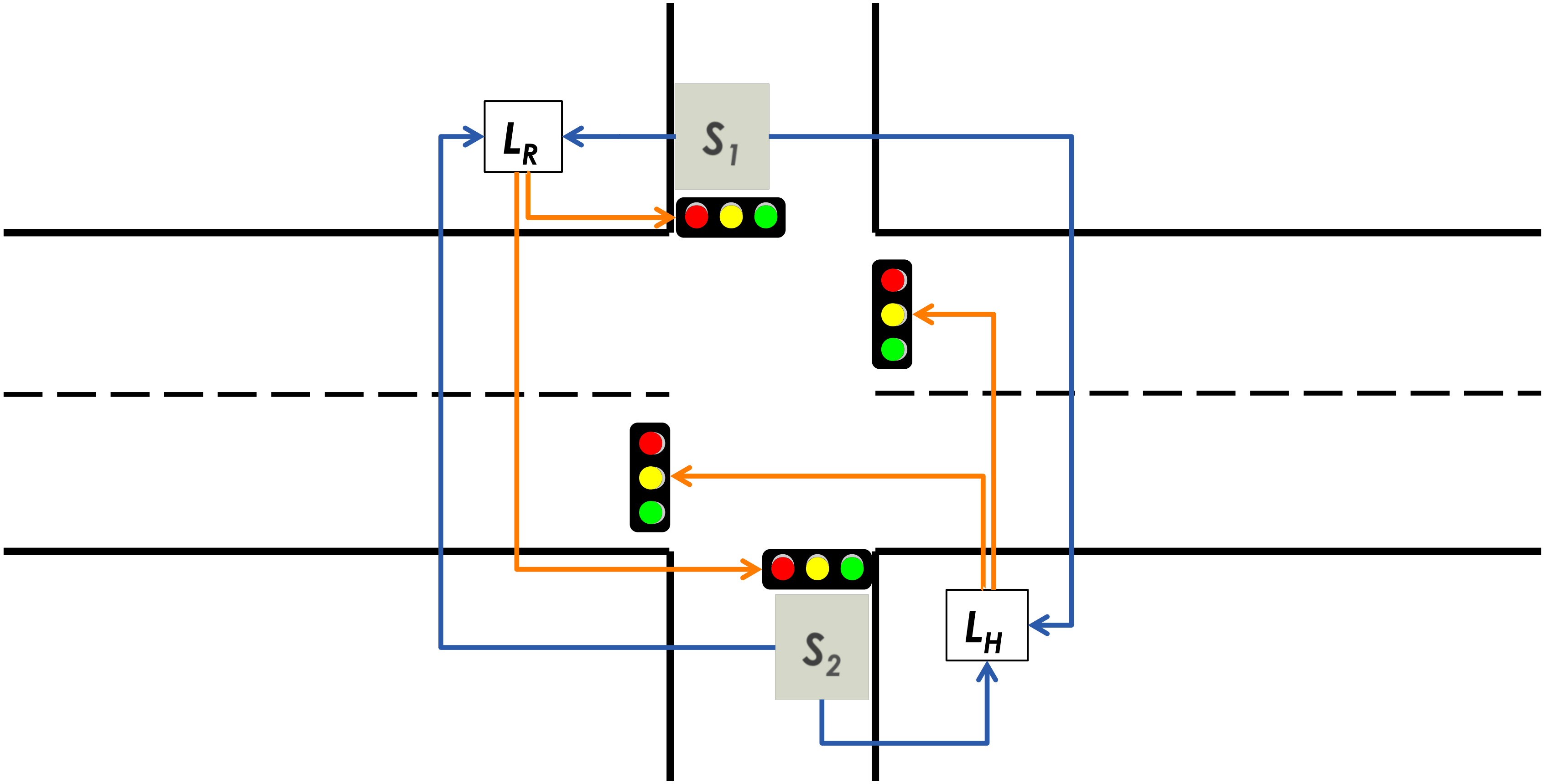


Turing Machine



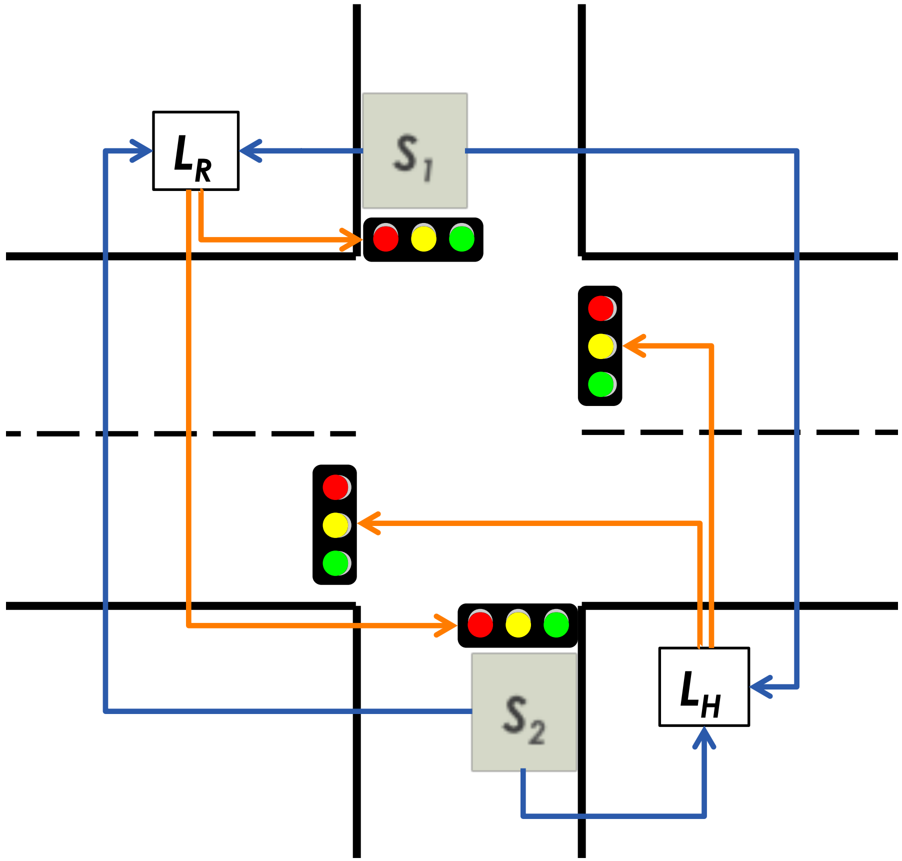
Finite State Machines

Example: Traffic Light Controller



Finite State Machines

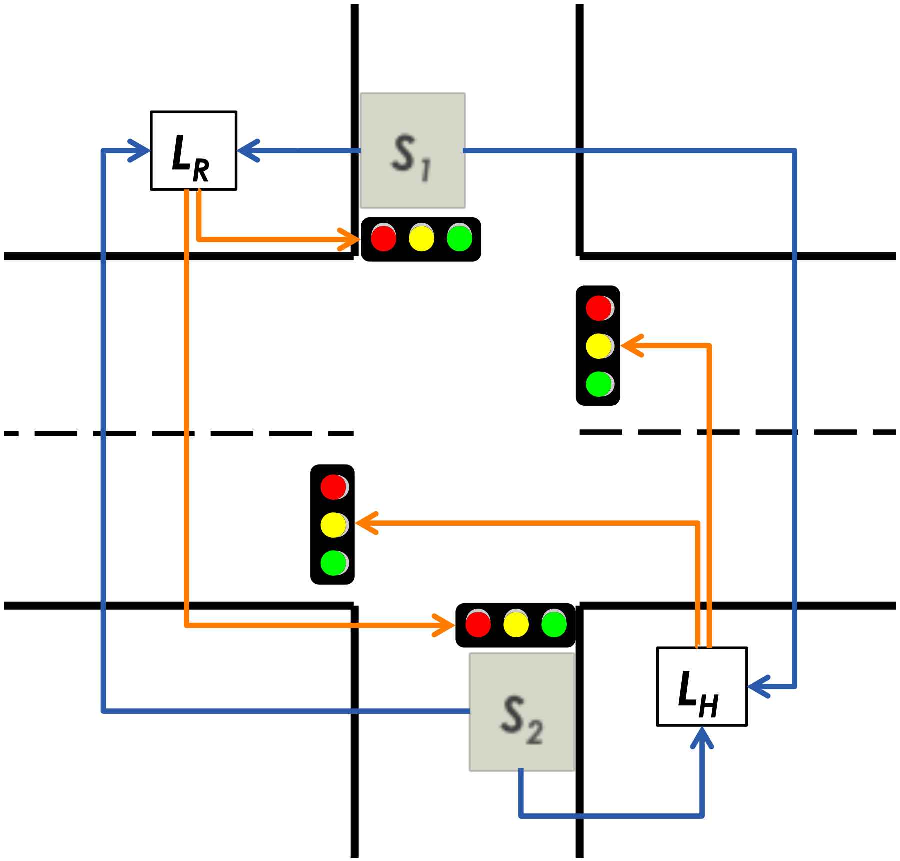
Example: Traffic Light Controller



Objective:
Design traffic light controllers L_R and L_H that use sensors S_1 and S_2 to give a green light to highway traffic unless there are cars waiting to cross from the minor road.

Finite State Machines

Example: Traffic Light Controller



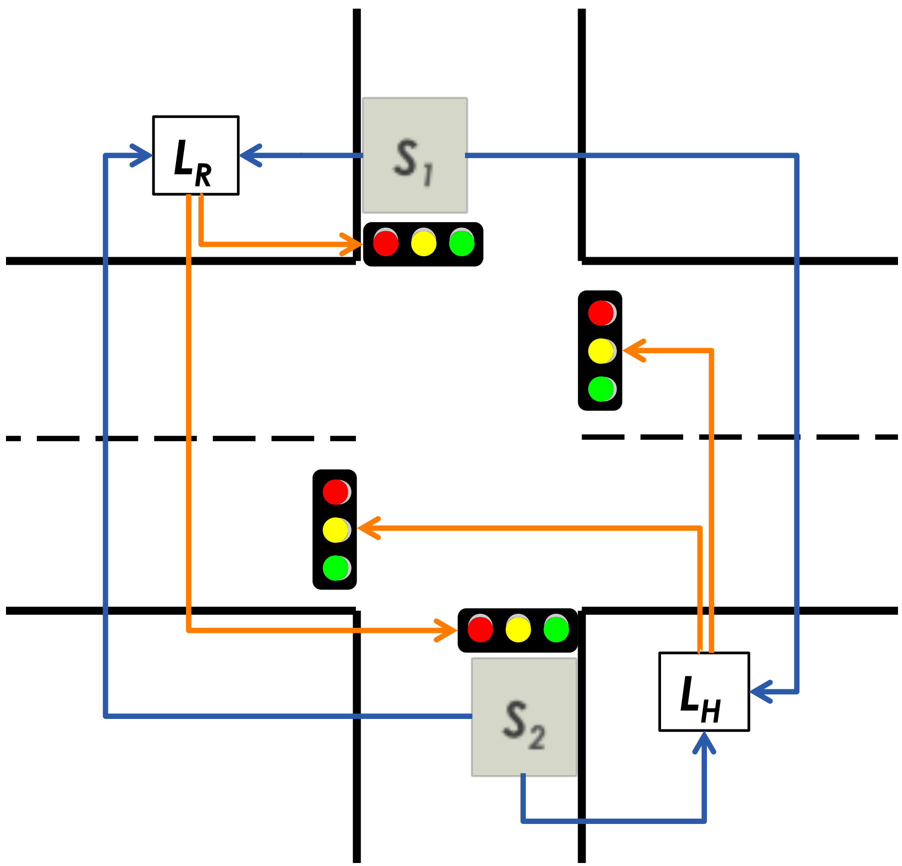
States: $L_R \times L_H$

$L_R, L_H \in \{R, Y, G\}$

Inputs: $\{S_1, S_2, \bar{S}_1, \bar{S}_2\}$

Finite State Machines

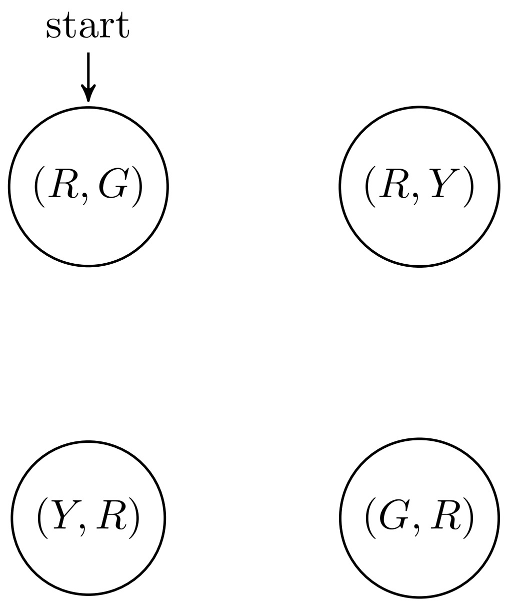
Example: Traffic Light Controller



States: $L_R \times L_H$

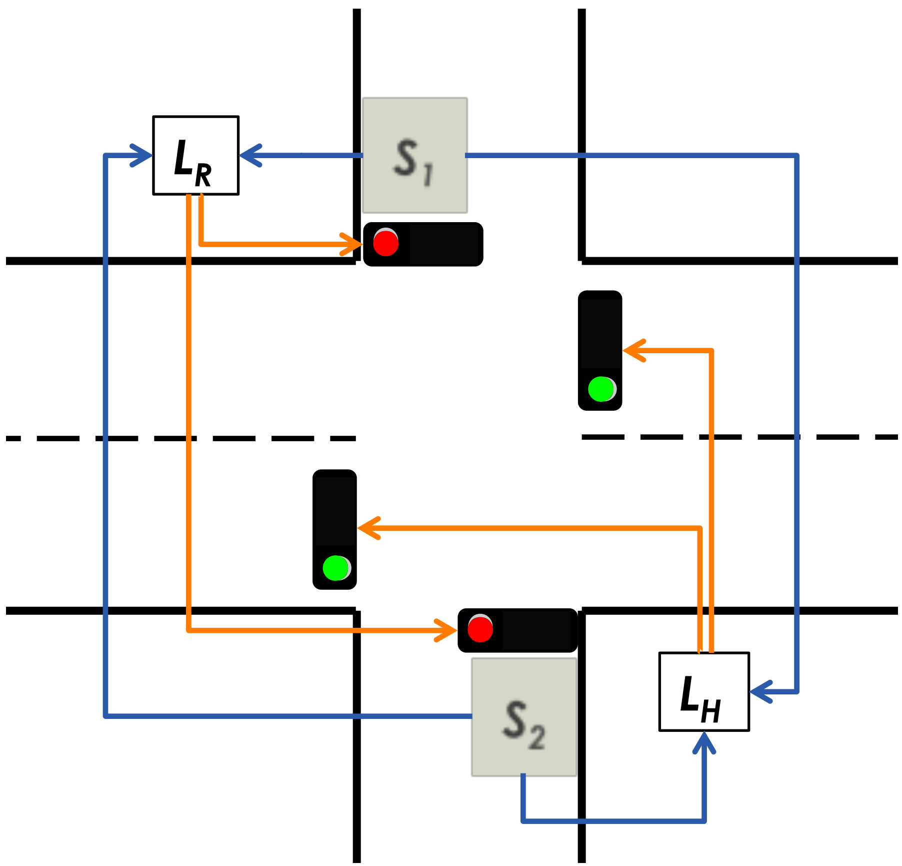
$L_R, L_H \in \{R, Y, G\}$

Inputs: $\{S_1, S_2, \bar{S}_1, \bar{S}_2\}$



Finite State Machines

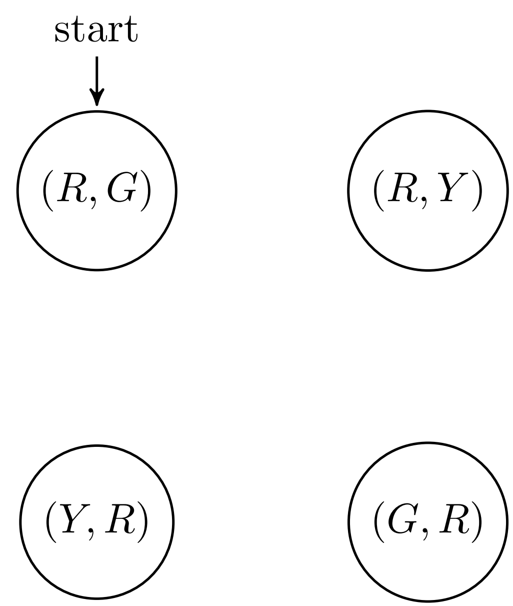
Example: Traffic Light Controller



States: $L_R \times L_H$

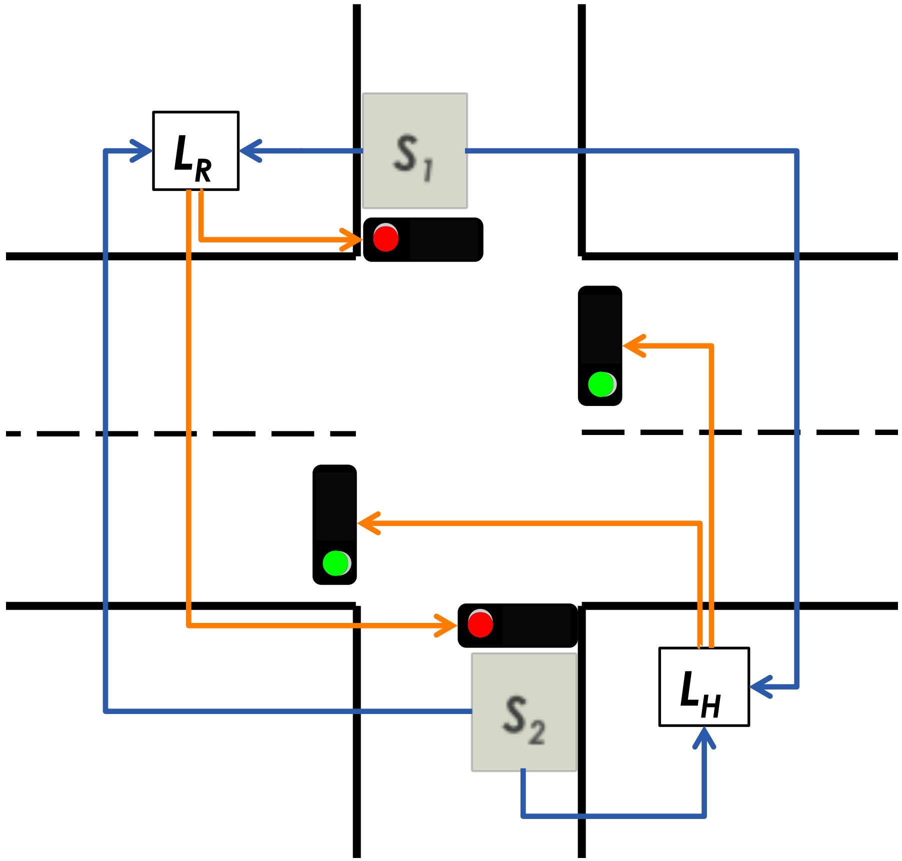
$$L_R, L_H \in \{R, Y, G\}$$

Inputs: $\{S_1, S_2, \bar{S}_1, \bar{S}_2\}$



Finite State Machines

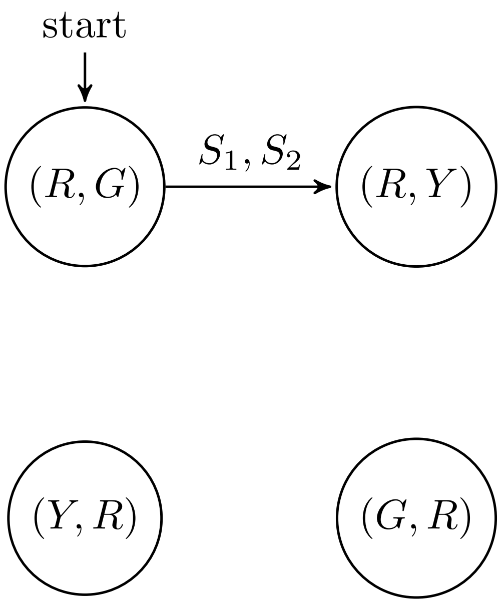
Example: Traffic Light Controller



States: $L_R \times L_H$

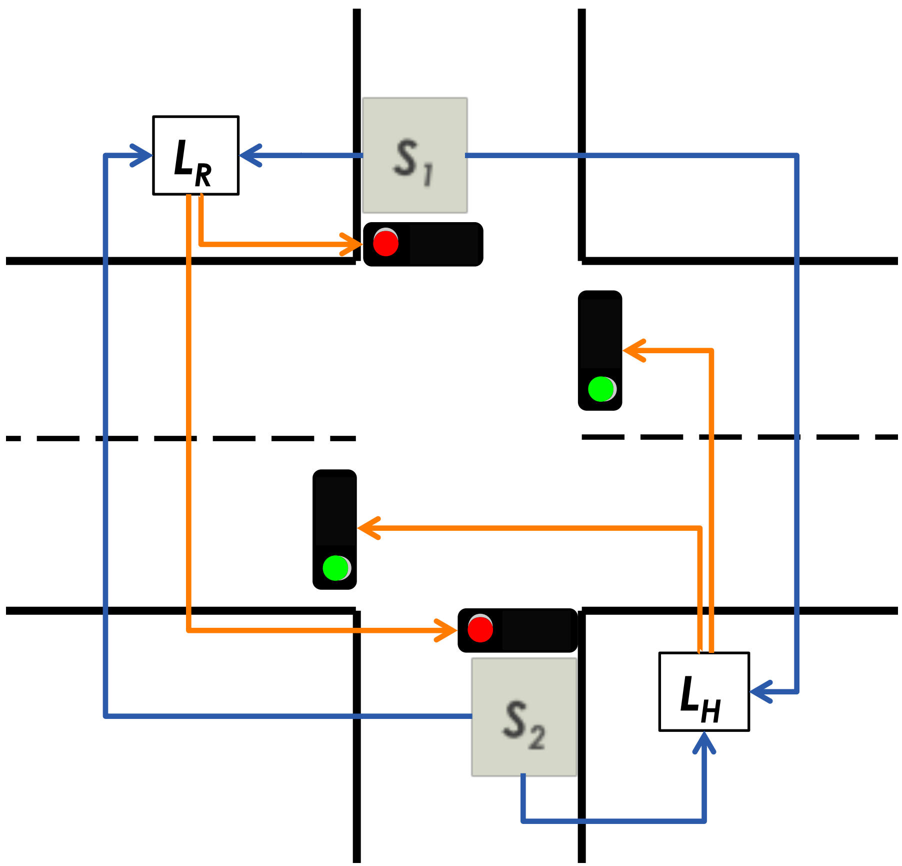
$L_R, L_H \in \{R, Y, G\}$

Inputs: $\{S_1, S_2, \bar{S}_1, \bar{S}_2\}$



Finite State Machines

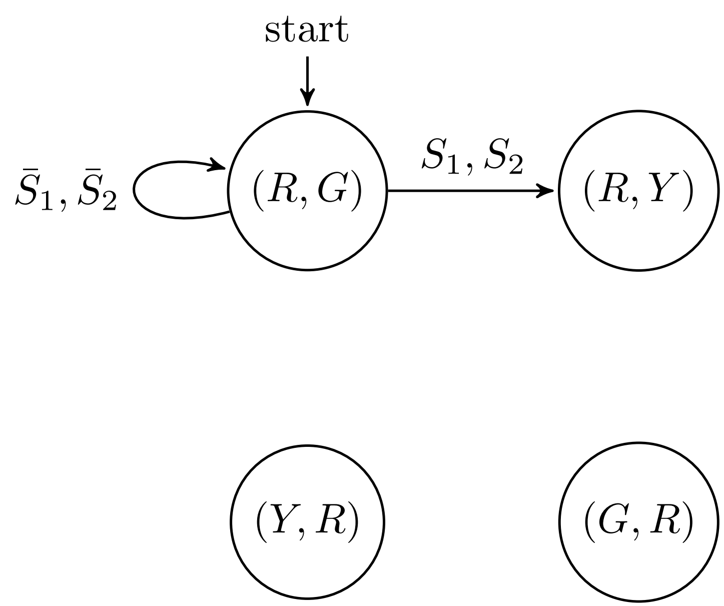
Example: Traffic Light Controller



States: $L_R \times L_H$

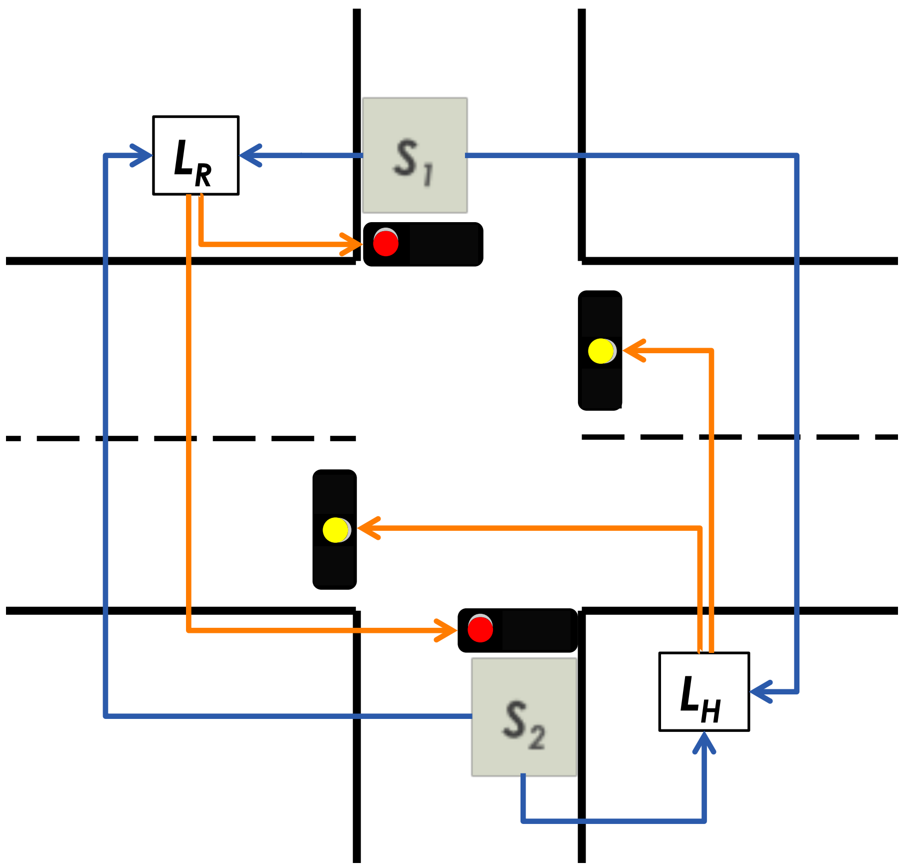
$L_R, L_H \in \{R, Y, G\}$

Inputs: $\{S_1, S_2, \bar{S}_1, \bar{S}_2\}$



Finite State Machines

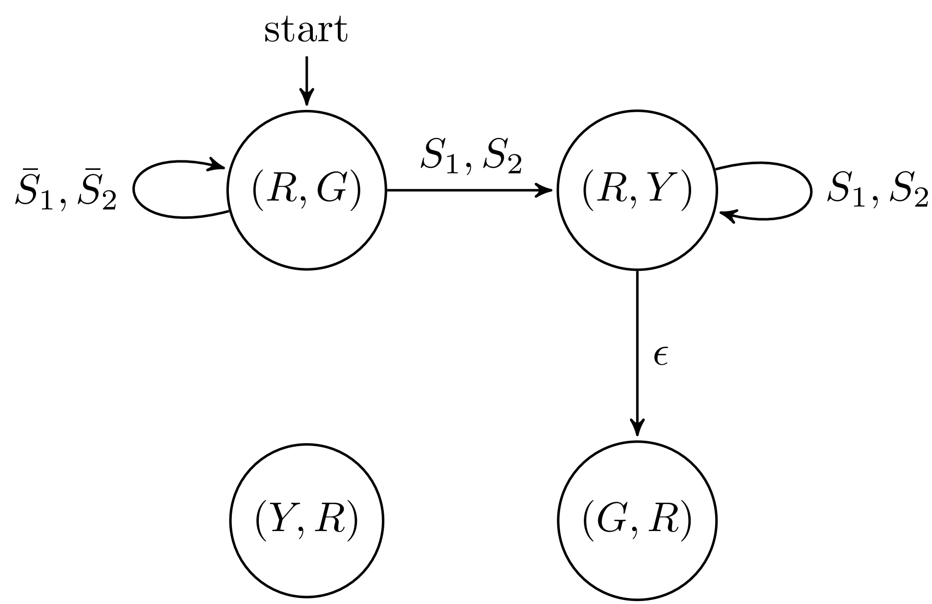
Example: Traffic Light Controller



States: $L_R \times L_H$

$L_R, L_H \in \{R, Y, G\}$

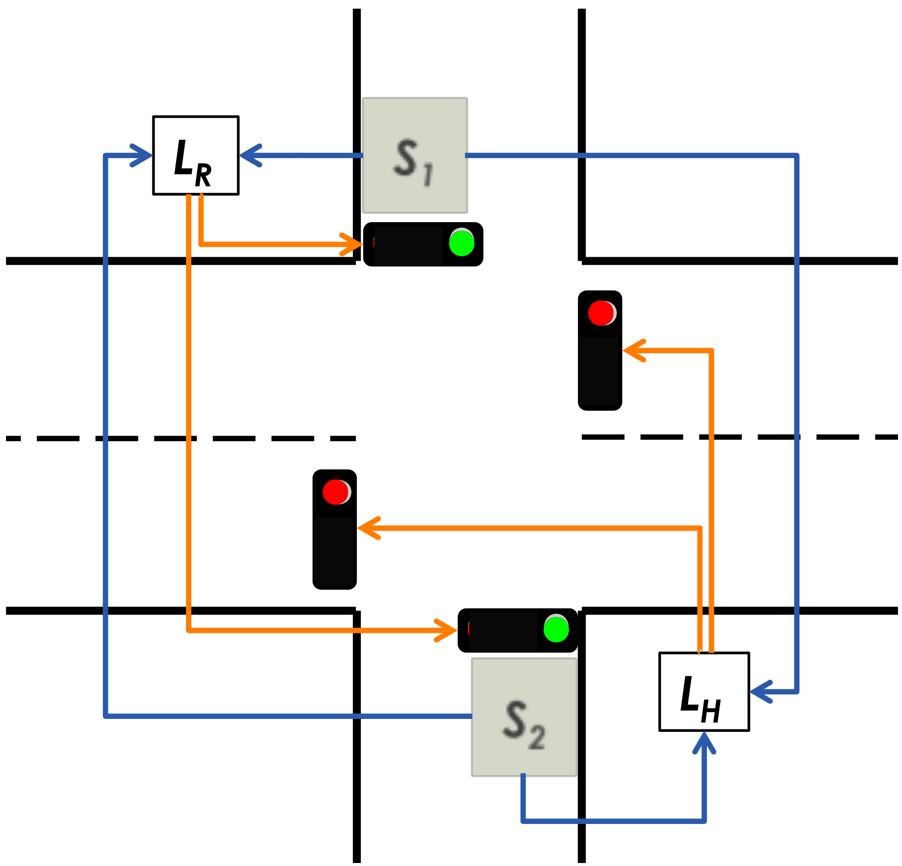
Inputs: $\{S_1, S_2, \bar{S}_1, \bar{S}_2, \epsilon\}$



What are the rest of the state transitions?

Finite State Machines

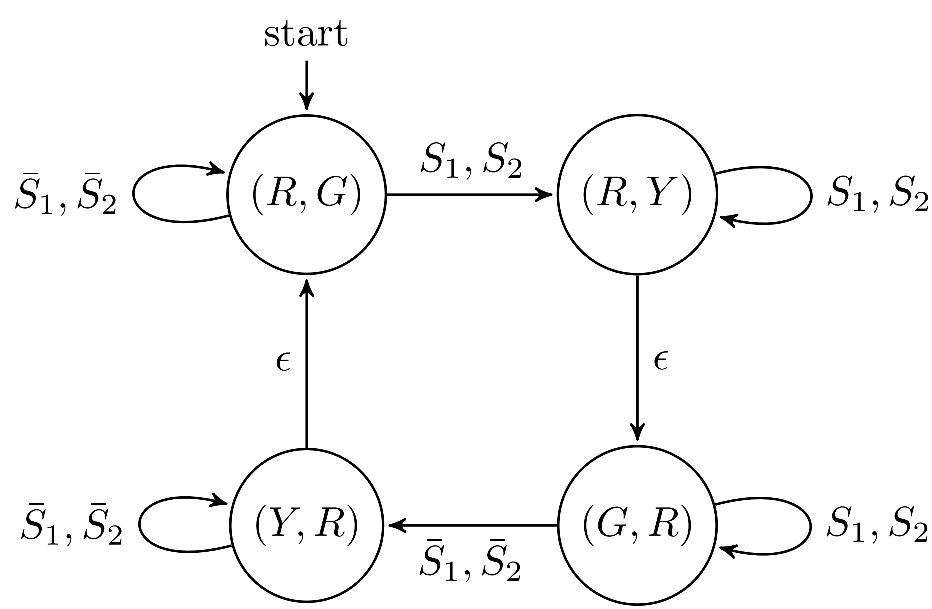
Example: Traffic Light Controller



States: $L_R \times L_H$

$L_R, L_H \in \{R, Y, G\}$

Inputs: $\{S_1, S_2, \bar{S}_1, \bar{S}_2, \epsilon\}$



What are the rest of the state transitions?

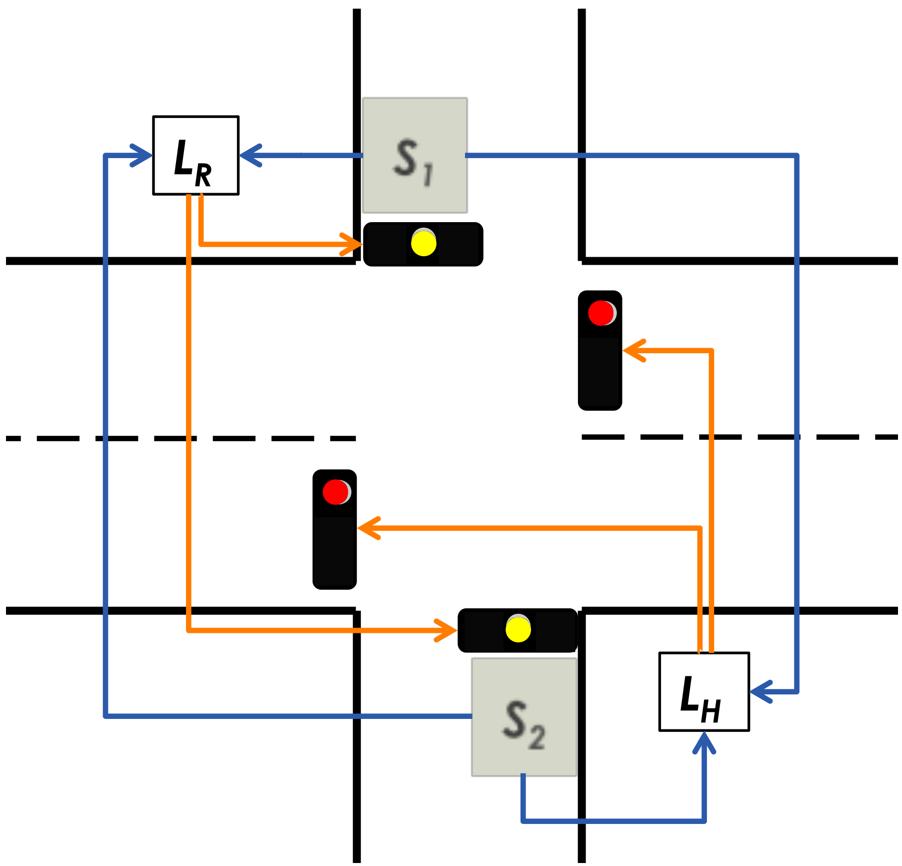
Finite State Machines

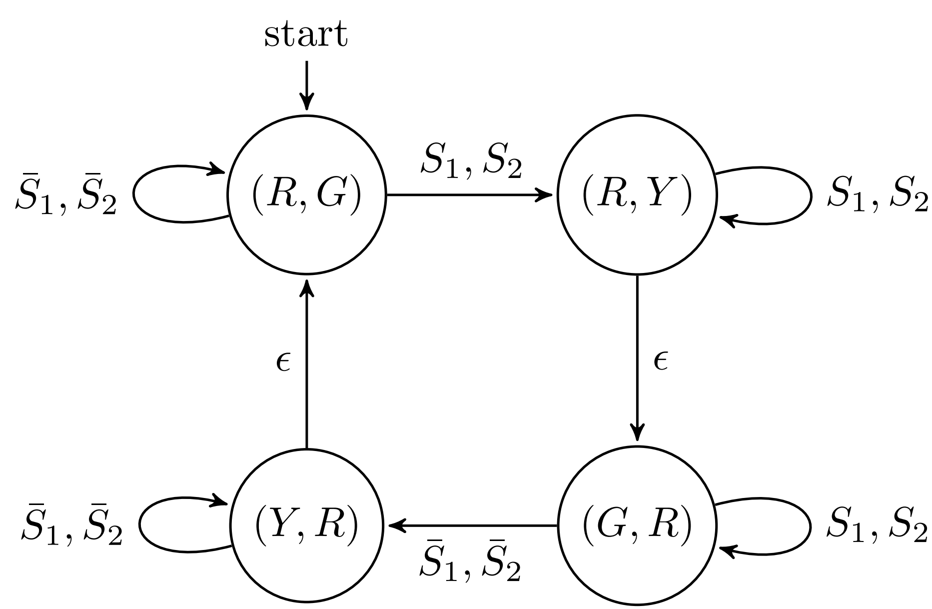
Example: Traffic Light Controller

States: $L_R \times L_H$

$L_R, L_H \in \{R, Y, G\}$

Inputs: $\{S_1, S_2, \bar{S}_1, \bar{S}_2, \epsilon\}$



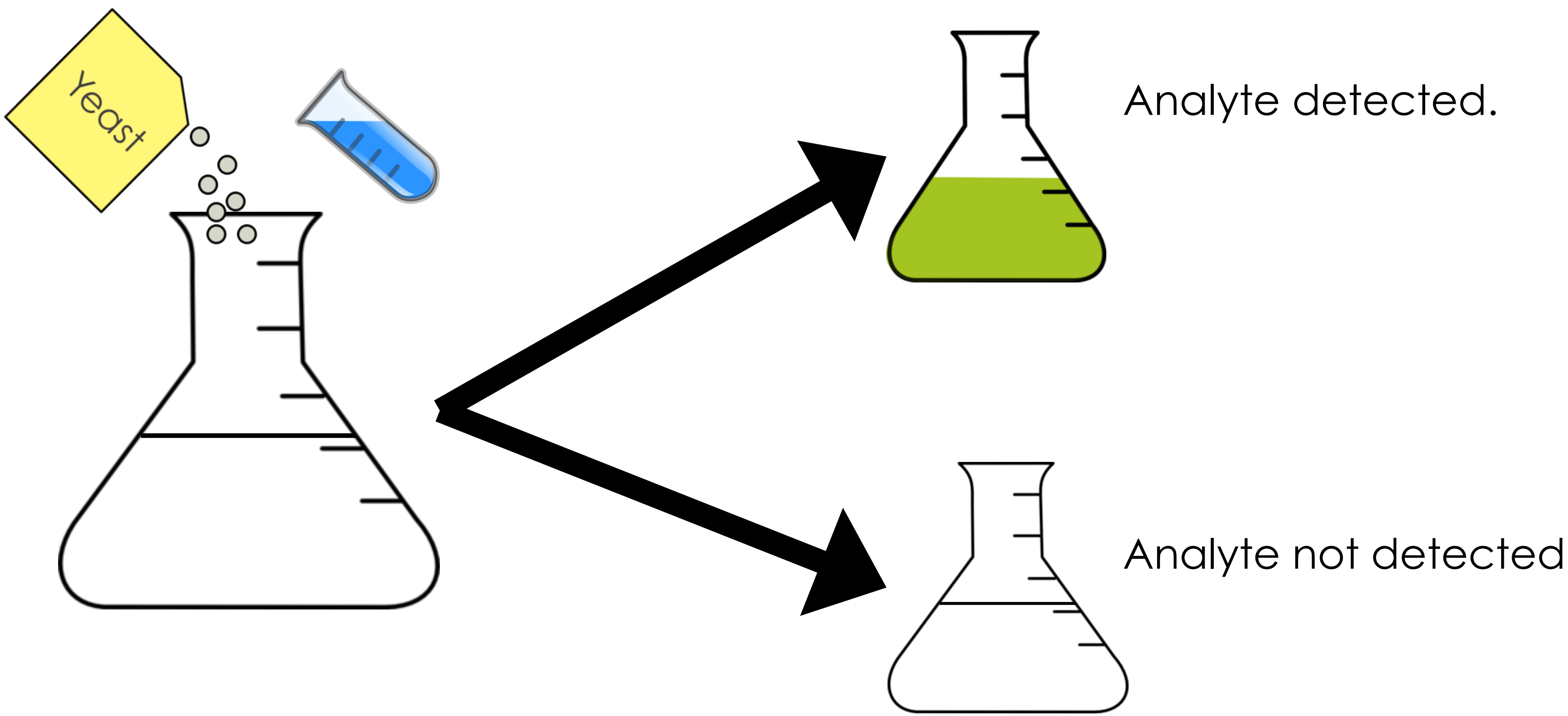


What are the rest of the state transitions?

Finite State Machines

Example: Yeast-based Ultrasensitive Detector

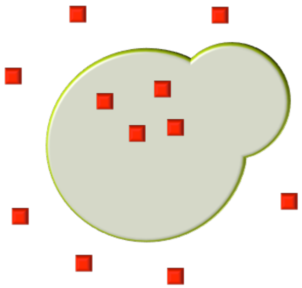
Objective: Use yeast to detect a very small number of a particular type of molecule (eg. protein markers in the early stages of an infection).



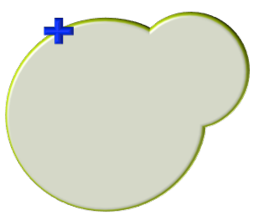
Finite State Machines

Example: Yeast-based Ultrasensitive Detector

This is an engineered yeast strain USD001.



USD001 can produce and sense a small diffusible molecule, AHL.

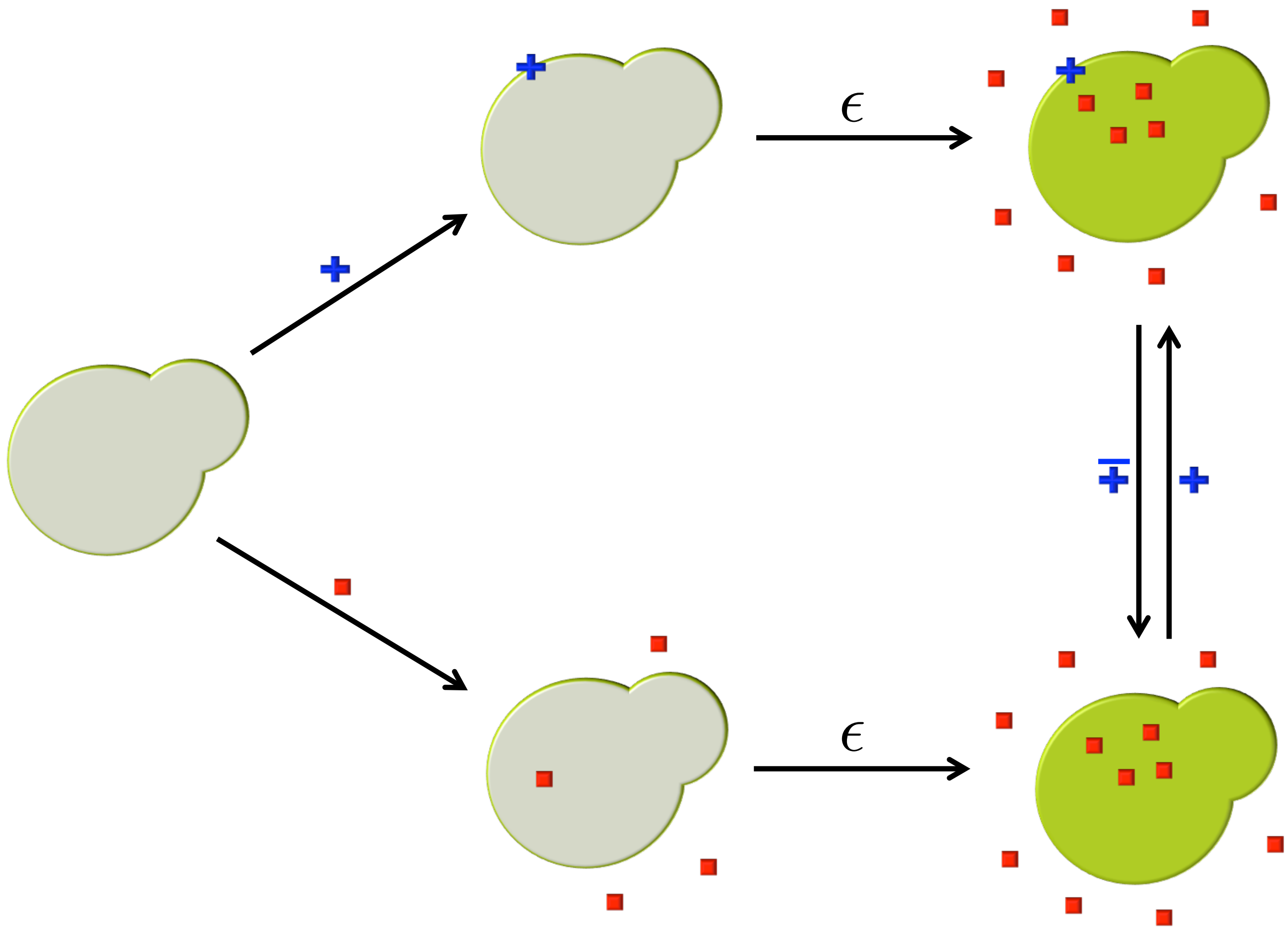


USD001 can also sense a single protein associated with an infectious disease.

USD001 can express a green fluorescent protein.

Finite State Machines

Example: Yeast-based Ultrasensitive Detector

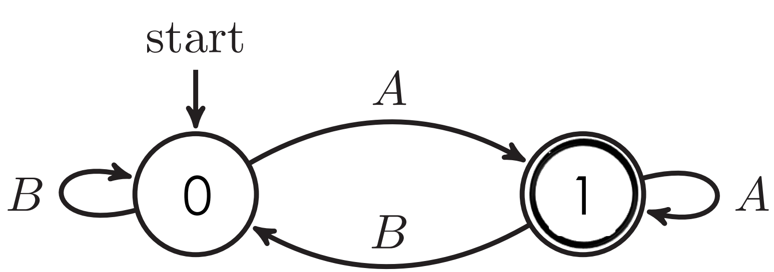


Finite State Machines

Syntax

$$M = (Q, \Sigma, \delta, q_0, F)$$

Symbol	Meaning
Q	set of *states*
Σ	set of *input symbols*
$\delta : Q \times \Sigma \to Q$	*state transition* function
$q_0 \in Q$	*initial state*
$F \subseteq Q$	set of *accepting states*



Semantics

Input: $w = \sigma_1\sigma_2 \ldots \sigma_n \in \Sigma^*$
Output: *Accept* or *Not Accept*

The machine begins in state q_0.

At each step i an input symbol σ_i is taken from the head of the input w.

The next state of the machine is
$q_i = \delta(q_{i-1}, \sigma_i)$

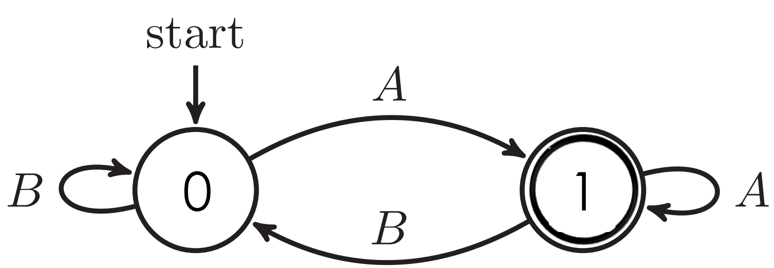
The input w is accepted if and only if $q_n \in F$.

Finite State Machines

Syntax

$$M = (Q, \Sigma, \delta, q_0, F)$$

Symbol	Meaning
Q	set of *states*
Σ	set of *input symbols*
$\delta : Q \times \Sigma \to Q$	*state transition* function
$q_0 \in Q$	*initial state*
$F \subseteq Q$	set of *accepting states*



Semantics

Input: $w = \sigma_1\sigma_2 \ldots \sigma_n \in \Sigma^*$
Output: *Accept* or *Not Accept*

The machine begins in state q_0.

At each step i an input symbol σ_i is taken from the head of the input w.

The next state of the machine is
$q_i = \delta(q_{i-1}, \sigma_i)$

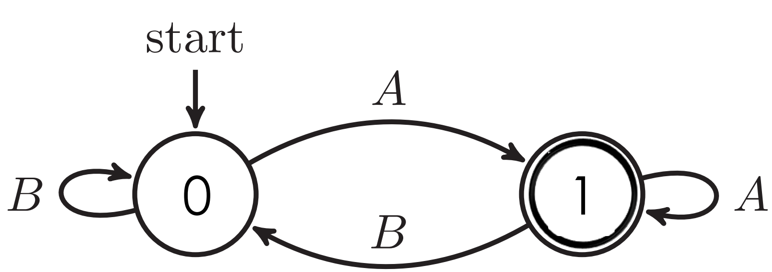
The input w is accepted if and only if $q_n \in F$.

Finite State Machines

Syntax

$$M = (Q, \Sigma, \delta, q_0, F)$$

Symbol	Meaning
Q	set of *states*
Σ	set of *input symbols*
$\delta : Q \times \Sigma \to Q$	*state transition* function
$q_0 \in Q$	*initial state*
$F \subseteq Q$	set of *accepting states*



Semantics

Input: $w = \sigma_1\sigma_2\ldots\sigma_n \in \Sigma^*$
Output: *Accept* or *Not Accept*

What strings does this machine accept?

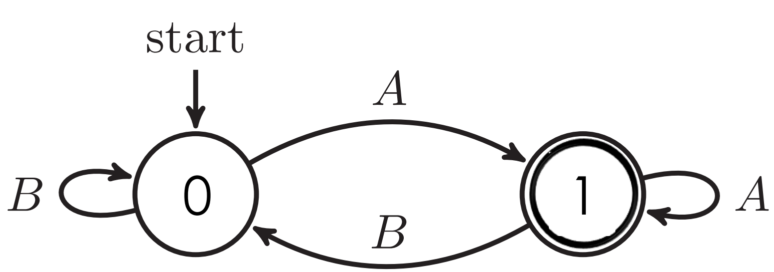
A
BBBB
BABABA
AAAABBBBA

Finite State Machines

Syntax

$$M = (Q, \Sigma, \delta, q_0, F)$$

Symbol	Meaning
Q	set of *states*
Σ	set of *input symbols*
$\delta : Q \times \Sigma \to Q$	*state transition* function
$q_0 \in Q$	*initial state*
$F \subseteq Q$	set of *accepting states*



Semantics

Input: $w = \sigma_1\sigma_2 \ldots \sigma_n \in \Sigma^*$
Output: *Accept* or *Not Accept*

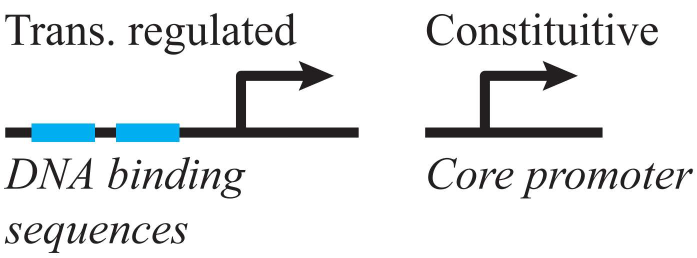
What strings does this machine accept?

A
~~BBBB~~
BABABA
AAAABBBBA

This machine accepts strings that end in "A", i.e. the regular expression (A*B*)*A.

Biomolecular Parts

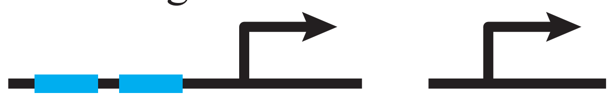
A.



Biomolecular Parts

A.

Trans. regulated　　　　Constituitive

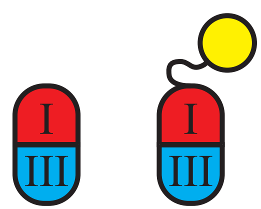


DNA binding sequences　　　　*Core promoter*

B.

I. Transriptional repression domain

III. Programmable DNA binding domain

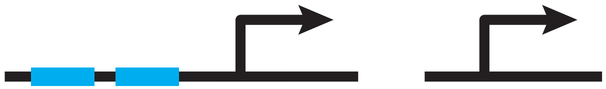


Fluorscent marker

Biomolecular Parts

A.

Trans. regulated | Constituitive



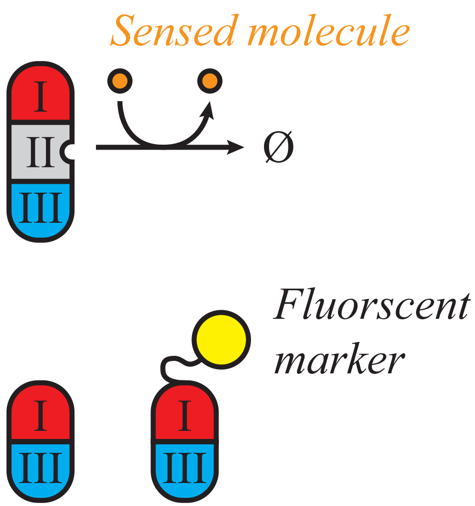
DNA binding sequences | *Core promoter*

B.

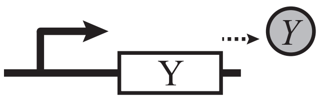
I. Transriptional repression domain

II. Small molecule recognition site/ degron

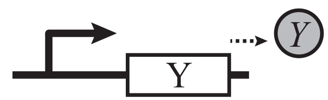
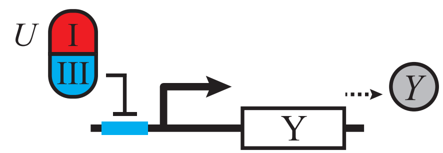
III. Programmable DNA binding domain



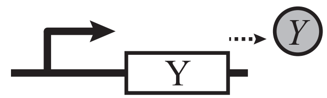
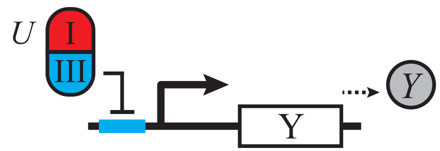
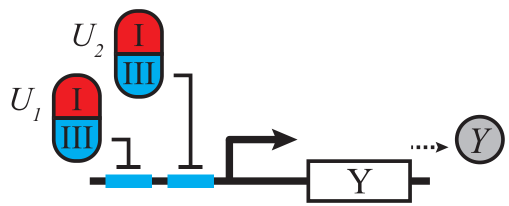
Biomolecular Parts

Component Type	Biomolecular Realization	Gene Regulatory Network
Transcriptionally Unregulated Gene		Y

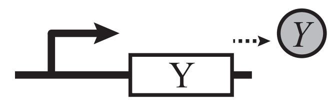
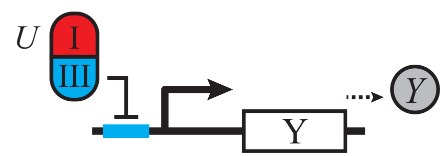
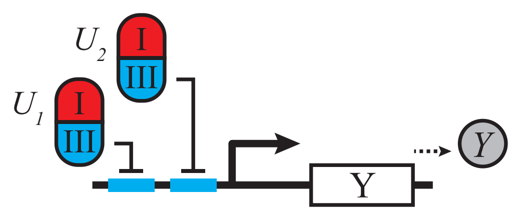
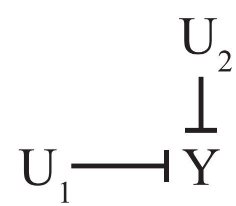
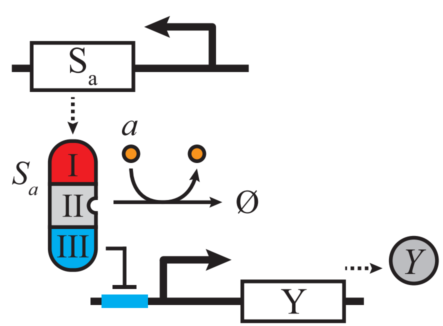
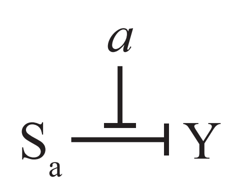
Biomolecular Parts

Component Type	Biomolecular Realization	Gene Regulatory Network
Transcriptionally Unregulated Gene		Y
Singly Regulated Gene		$U \dashv Y$

Biomolecular Parts

Component Type	Biomolecular Realization	Gene Regulatory Network
Transcriptionally Unregulated Gene		Y
Singly Regulated Gene		$U \dashv Y$
Doubly Regulated Gene		$U_1 \dashv Y$ $U_2 \dashv Y$

Biomolecular Parts

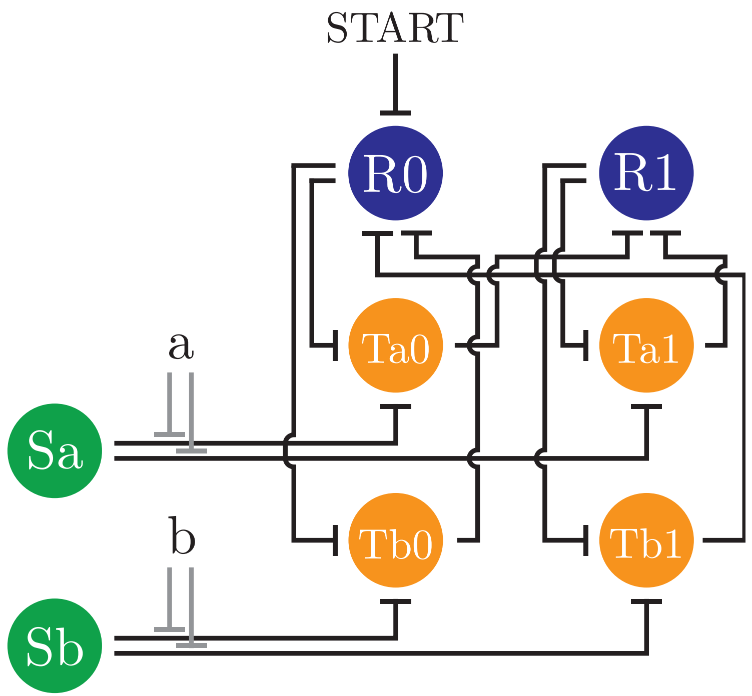
Component Type	Biomolecular Realization	Gene Regulatory Network
Transcriptionally Unregulated Gene		Y
Singly Regulated Gene		$U \dashv Y$
Doubly Regulated Gene		
Small Molecule Sensor		

Gene Regulatory Networks

Syntax

$$G = (V, U, E_r, H_r)$$

Symbol	Meaning
V	Set of *gene products*
U	Set of *inducers*
$E_r \subset V \times V$	*repression relation*
$H_r \subset U \times E_r$	*inducible repression relation*

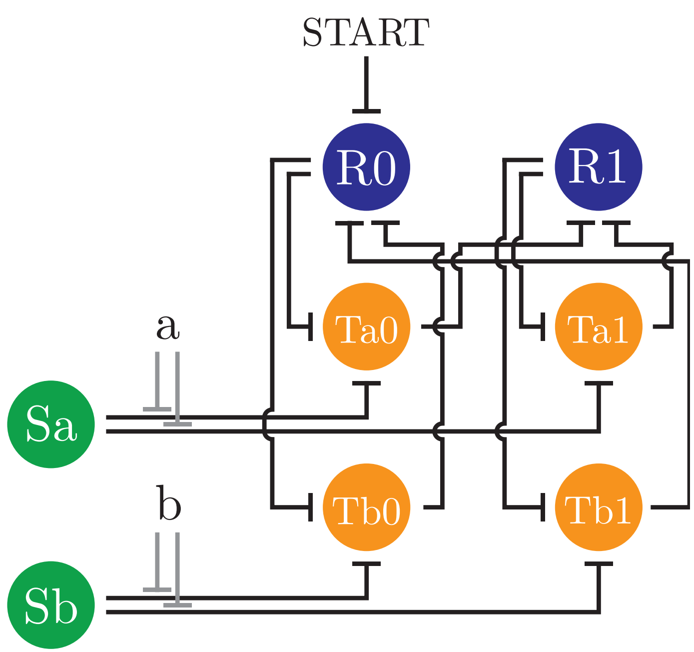


Gene Regulatory Networks

Syntax

$$G = (V, U, E_r, H_r)$$

Symbol	Meaning
V	Set of *gene products*
U	Set of *inducers*
$E_r \subset V \times V$	*repression relation*
$H_r \subset U \times E_r$	*inducible repression relation*



Semantics

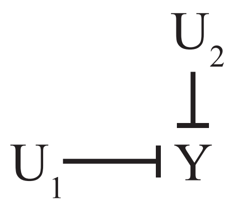
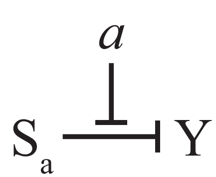
Boolean Network Dynamics (Kauffman, 1969)

$$Y^t = f(Y^{t-1}, U^t)$$

Let Y^t be a time-varying state vector and U^t be a time-varying input vector, i.e.,

$$Y^t = \begin{bmatrix} R_0^t \\ R_1^t \\ S_a^t \\ S_b^t \\ T_{a,0}^t \\ T_{b,0}^t \\ T_{a,1}^t \\ T_{b,1}^t \end{bmatrix} \qquad U^t = \begin{bmatrix} a^t \\ b^t \end{bmatrix}$$

Boolean Network Dynamics

Component Type	Gene Regulatory Network	Boolean Network Equations
Transcriptionally Unregulated Gene	Y	$Y^t = on$
Singly Regulated Gene	$U \dashv Y$	$Y^{t+1} = \neg U^t$
Doubly Regulated Gene		$Y^{t+1} = \neg(U_1^t \vee U_2^t)$
Small Molecule Sensor		$S_a^t = \neg a^t$ $Y^{t+1} = \neg S_a^t$

Boolean Network Dynamics

Component Type	Gene Regulatory Network	Boolean Network Equations
Transcriptionally Unregulated Gene	Y	$Y^t = on$
Singly Regulated Gene	$\mathrm{U} \dashv \mathrm{Y}$	$Y^{t+1} = \neg U^t$
Doubly Regulated Gene	$\begin{array}{ccc} & & \mathrm{U_2} \\ & & \perp \\ \mathrm{U_1} & \dashv & \mathrm{Y} \end{array}$	$Y^{t+1} = \neg(U_1^t \vee U_2^t)$
Small Molecule Sensor	$\begin{array}{ccc} & a & \\ & \perp & \\ \mathrm{S_a} & \dashv & \mathrm{Y} \end{array}$	$S_a^t = \neg a^t$ $Y^{t+1} = \neg S_a^t$

U^t	Y^{t+1}
0	
1	

Boolean Network Dynamics

Component Type	Gene Regulatory Network	Boolean Network Equations
Transcriptionally Unregulated Gene	Y	$Y^t = on$
Singly Regulated Gene	$U \dashv Y$	$Y^{t+1} = \neg U^t$
Doubly Regulated Gene	$\begin{tikzcd} & U_2 \arrow[d, -\vert] \\ U_1 \arrow[r, -\vert] & Y \end{tikzcd}$	$Y^{t+1} = \neg(U_1^t \vee U_2^t)$
Small Molecule Sensor	$\begin{tikzcd} & a \arrow[d, -\vert] & \\ S_a \arrow[rr, -\vert] & & Y \end{tikzcd}$	$S_a^t = \neg a^t$ $Y^{t+1} = \neg S_a^t$

U^t	Y^{t+1}
0	1
1	0

Boolean Network Dynamics

Component Type	Gene Regulatory Network	Boolean Network Equations
Transcriptionally Unregulated Gene	Y	$Y^t = on$
Singly Regulated Gene	$U \dashv Y$	$Y^{t+1} = \neg U^t$
Doubly Regulated Gene	$U_1 \dashv Y$, $U_2 \dashv Y$	$Y^{t+1} = \neg(U_1^t \vee U_2^t)$
Small Molecule Sensor	$a \dashv S_a$, $S_a \dashv Y$	$S_a^t = \neg a^t$ $Y^{t+1} = \neg S_a^t$

What is the truth table for the Double Regulated Gene and Small Molecular Sensor?

U^t	Y^{t+1}
0	1
1	0

U_1^t	U_2^t	Y^{t+1}
0	0	1

a^t	Y^{t+1}

Boolean Network Dynamics

Component Type	Gene Regulatory Network	Boolean Network Equations
Transcriptionally Unregulated Gene	Y	$Y^t = on$
Singly Regulated Gene	$U \dashv Y$	$Y^{t+1} = \neg U^t$
Doubly Regulated Gene	$U_1 \dashv Y$, $U_2 \dashv Y$	$Y^{t+1} = \neg(U_1^t \vee U_2^t)$
Small Molecule Sensor	$a \dashv (S_a \dashv Y)$	$S_a^t = \neg a^t$ $Y^{t+1} = \neg S_a^t$

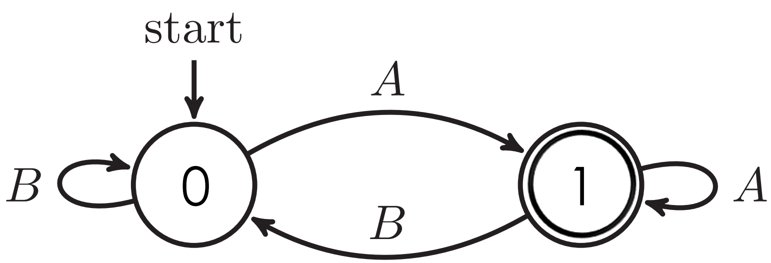
What is the truth table for the Double Regulated Gene and Small Molecular Sensor?

U^t	Y^{t+1}
0	1
1	0

U_1^t	U_2^t	Y^{t+1}
0	0	1
0	1	0
1	0	0
1	1	1

a^t	Y^{t+1}
0	0
1	1

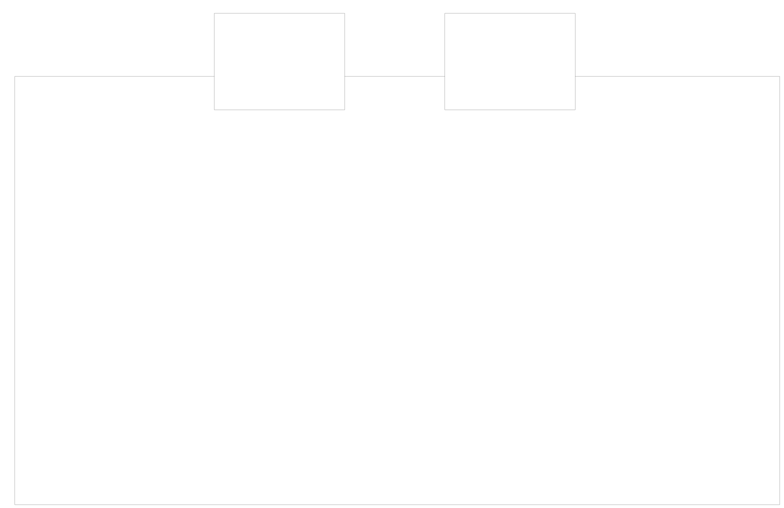
GRN General Construction Method



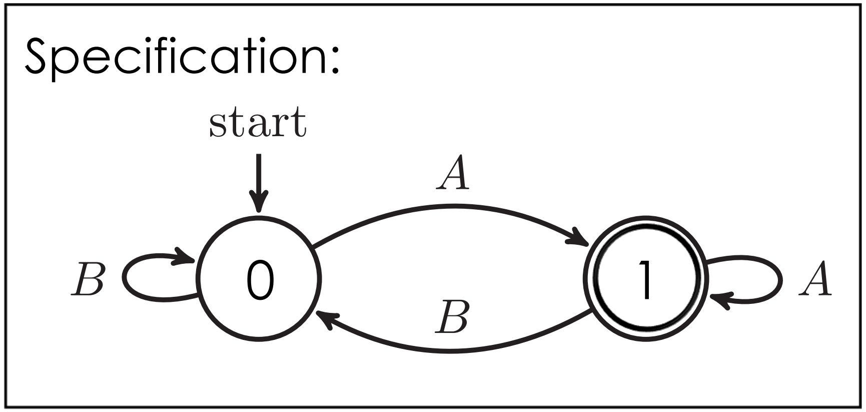
$$M = (Q, \Sigma, \delta, q_0, F)$$

Symbol	Meaning
Q	set of *states*
Σ	set of *input symbols*
$\delta : Q \times \Sigma \rightarrow Q$	*state transition* function
$q_0 \in Q$	*initial state*
$F \subseteq Q$	set of *accepting states*

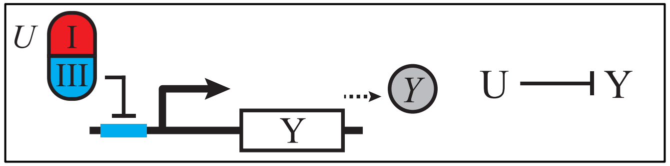
Objective:
Given a FSM M as a specification, construct a GRN $g(M)$ that encodes the behavior the FSM.

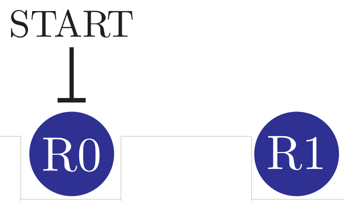


GRN General Construction Method



For each q in Q, let Rq be a singly regulated ***state gene***, and "wire" START to Rq_0

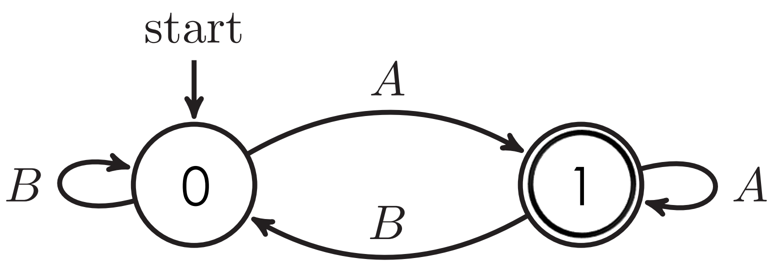




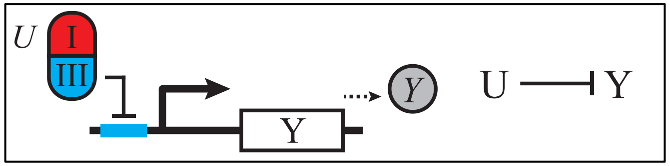
Each state q in Q, is represented by a gene expression profile where Rq is at a low level of expression, and all other state genes are at a high level of expression.

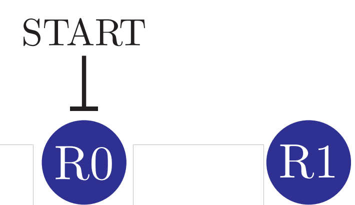
GRN General Construction Method

Specification:



For each q in Q, let Rq be a singly regulated ***state gene***, and "wire" START to Rq_0

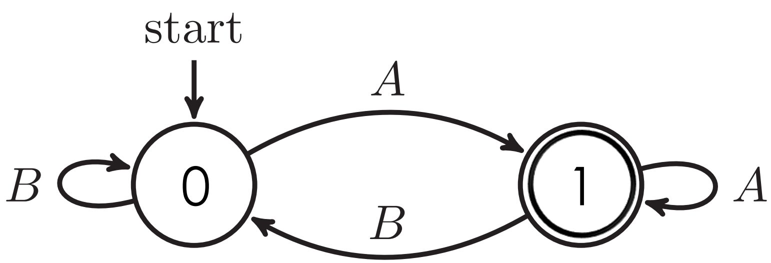




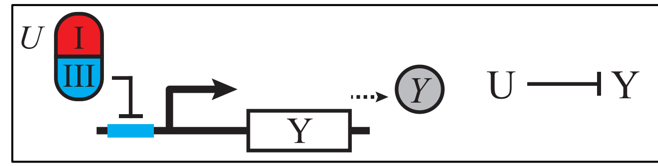
What are the expression levels of R0 and R1 with START on? START off?

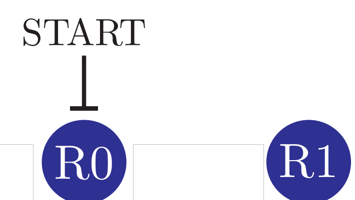
GRN General Construction Method

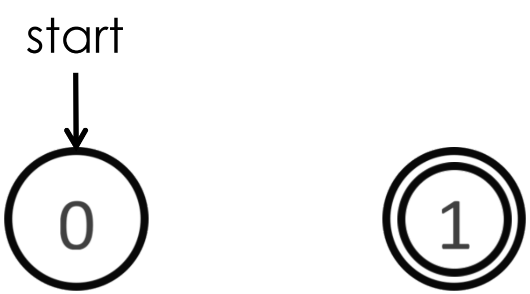
Specification:



For each q in Q, let Rq be a singly regulated ***state gene***, and "wire" START to Rq_0

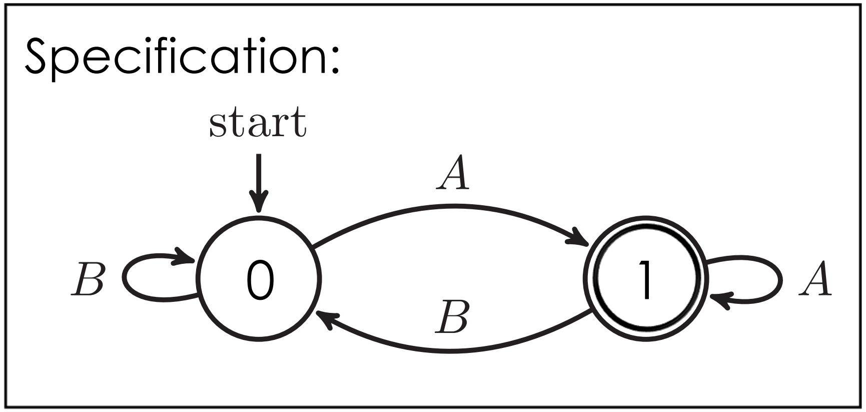




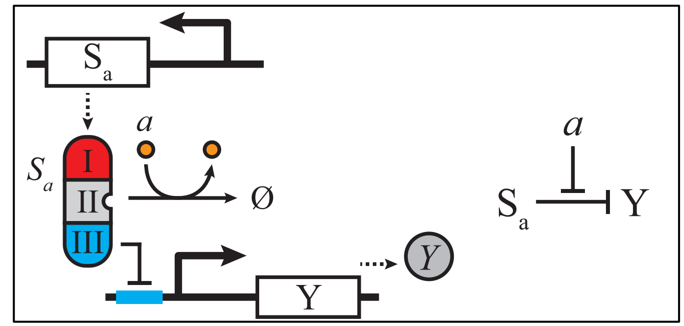


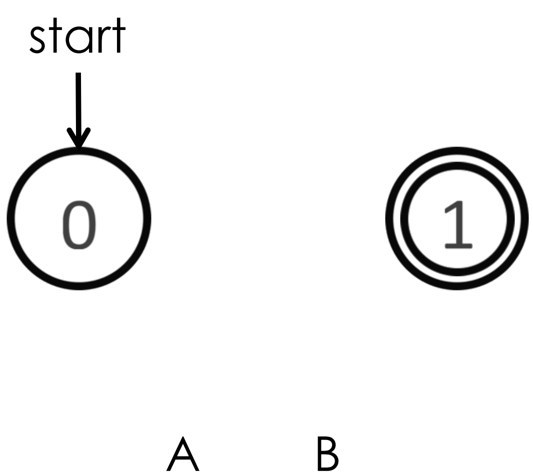
$START^t$	R_0^{t+1}	R_1^{t+1}
0	1	1
1	0	1

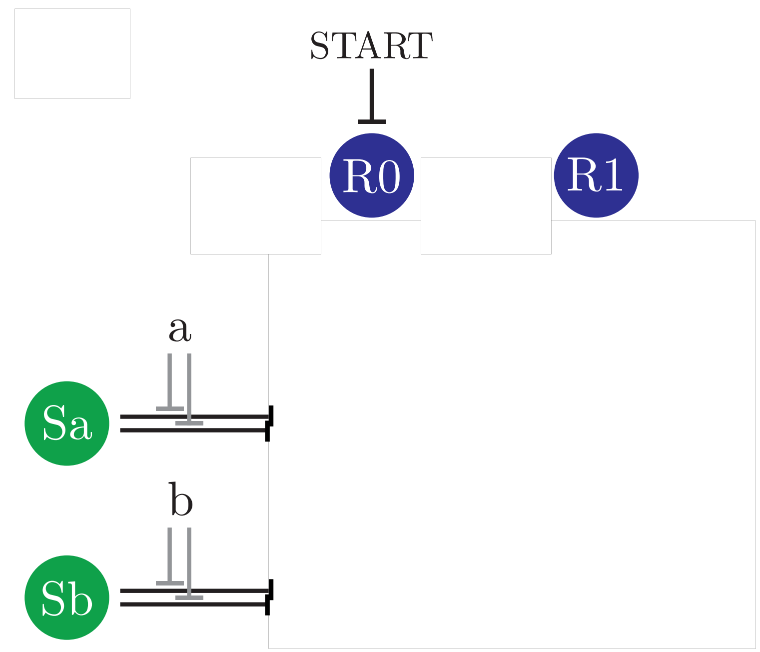
GRN General Construction Method



For each σ in Σ, let $S\sigma$ be a transcriptionally unregulated ***sensor gene*** for ***inducer*** σ.

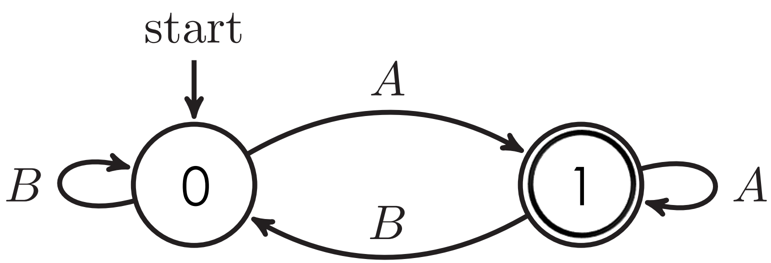




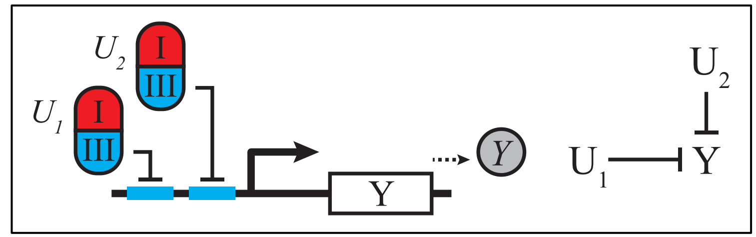


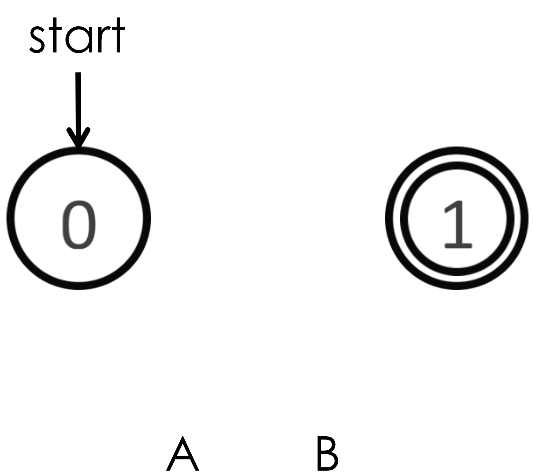
GRN General Construction Method

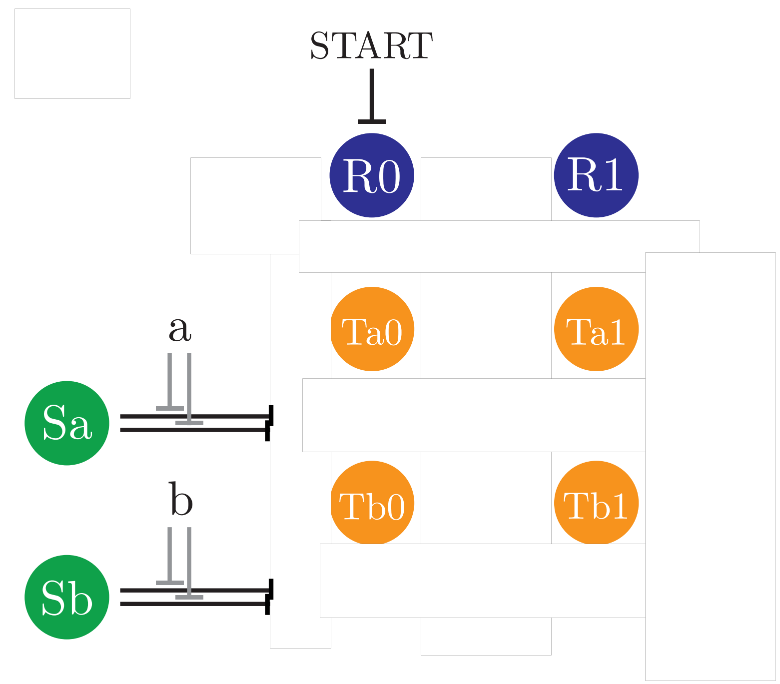
Specification:



For each (q, σ, q') in $Q \times \Sigma \times Q$, such that $\delta(q, \sigma)\ q'$, let $T\sigma q$ be a doubly regulated ***transition gene***.

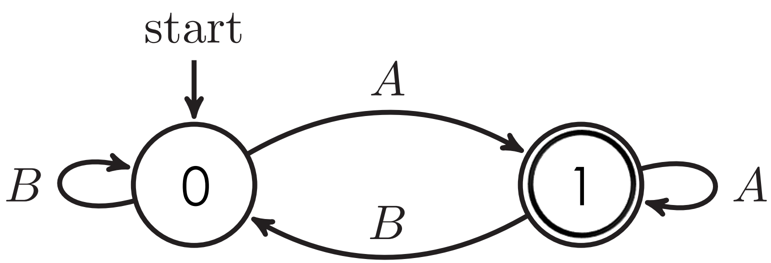




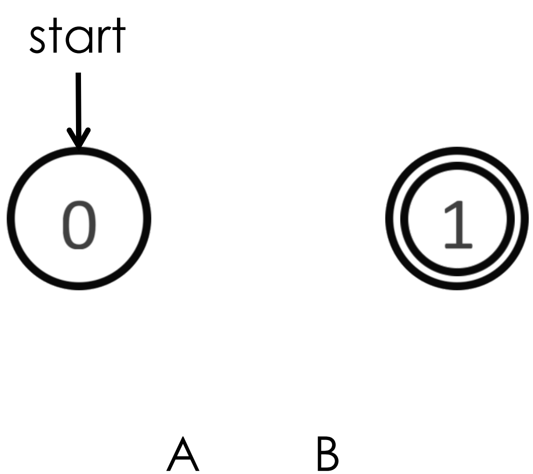


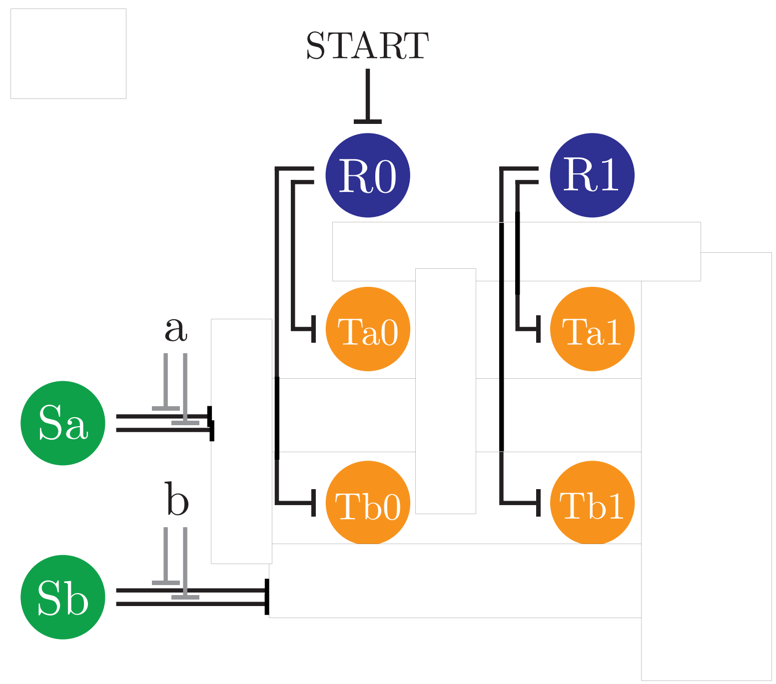
GRN General Construction Method

Specification:

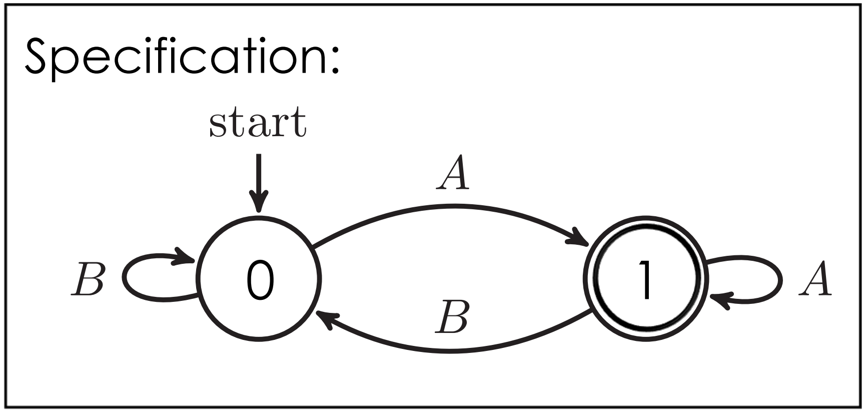


For each (q, σ, q') in $Q \times \Sigma \times Q$, such that $\delta(q, \sigma)\ q'$, wire Rq to $T\sigma q$.



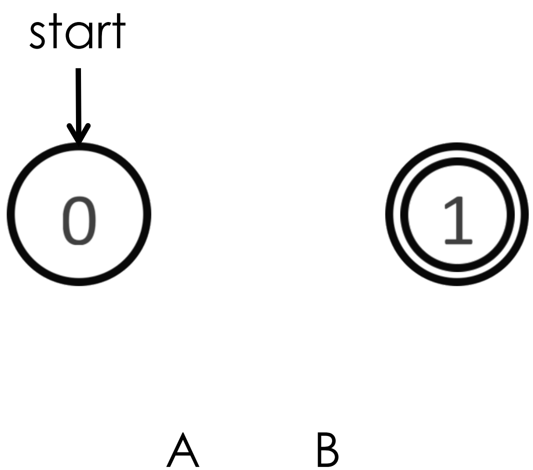


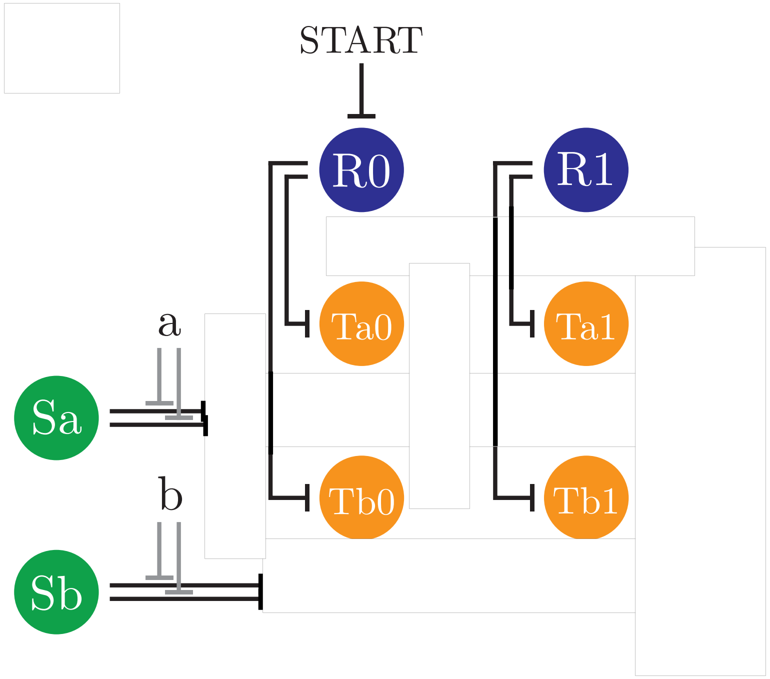
GRN General Construction Method



What is the Boolean function that describes the expression level of Ta0 at time t+1?

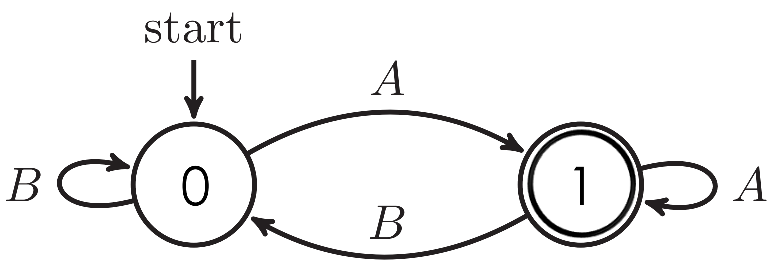
$$T_{a,0}^{t+1} = \;???$$





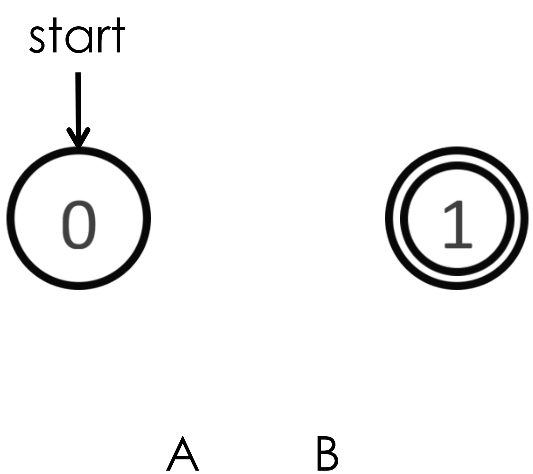
GRN General Construction Method

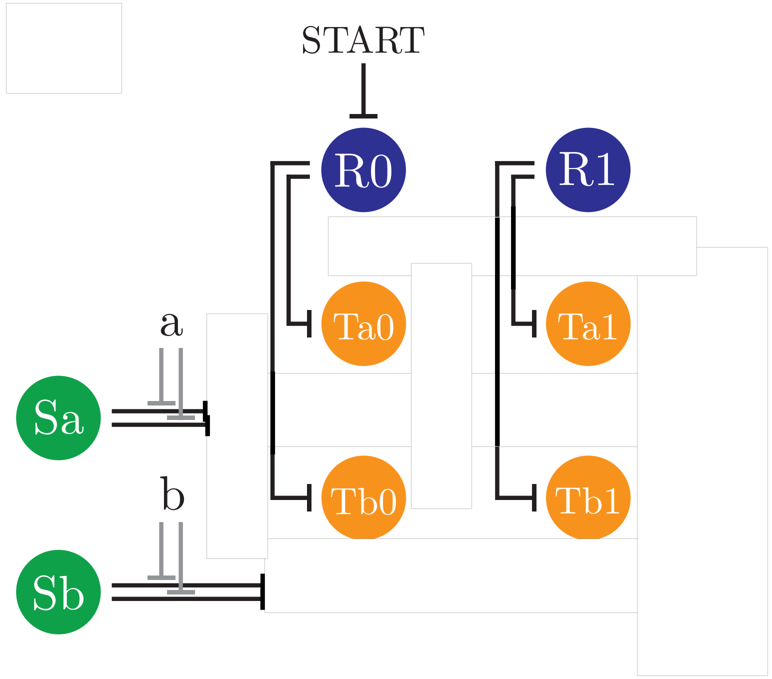
Specification:



What is the Boolean function that describes the expression level of Ta0 at time t+1?

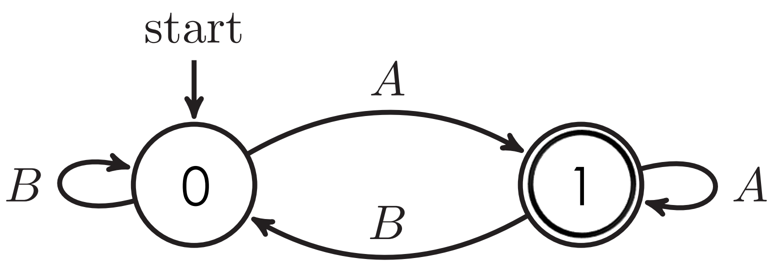
$$T_{a,0}^{t+1} = \neg R_0^t$$
$$= START^{t-1}$$



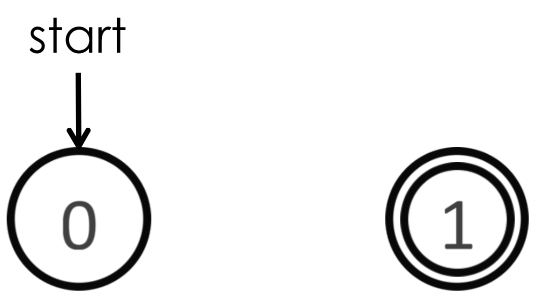


GRN General Construction Method

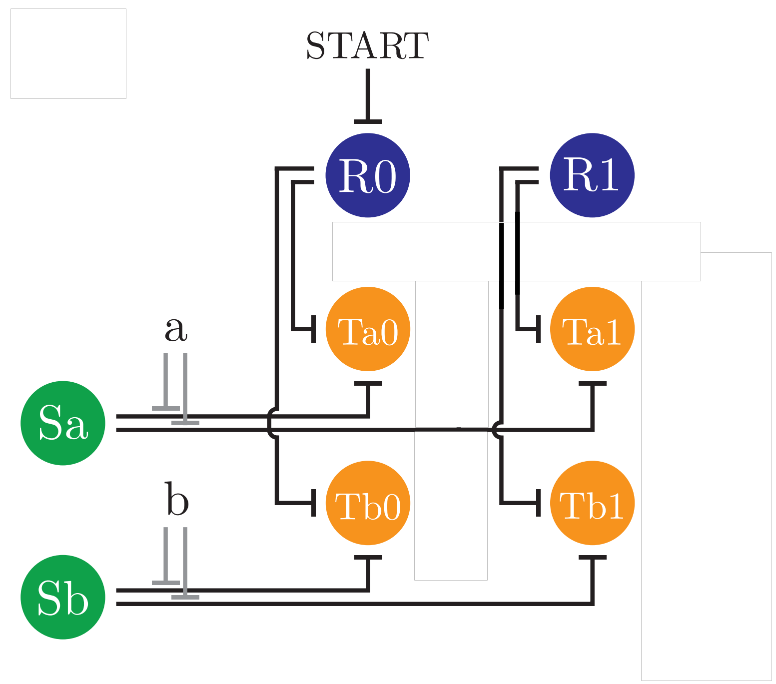
Specification:



For each (q, σ, q') in $Q \times \Sigma \times Q$, such that $\delta(q, \sigma)\ q'$, wire $S\sigma$ to $T\sigma q$.

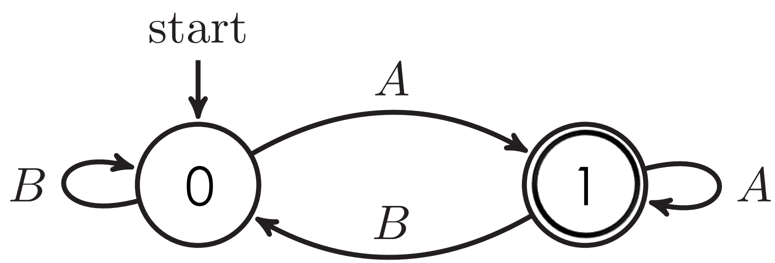


A B



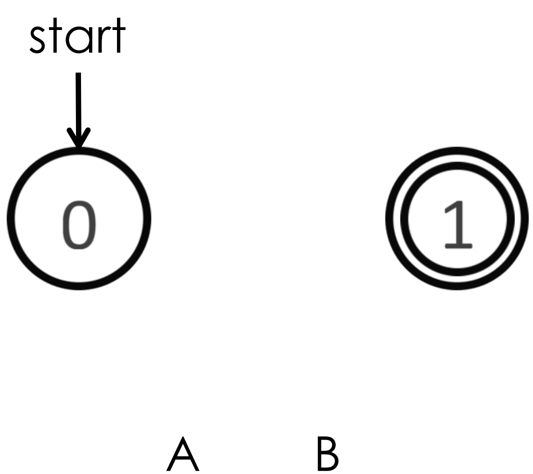
GRN General Construction Method

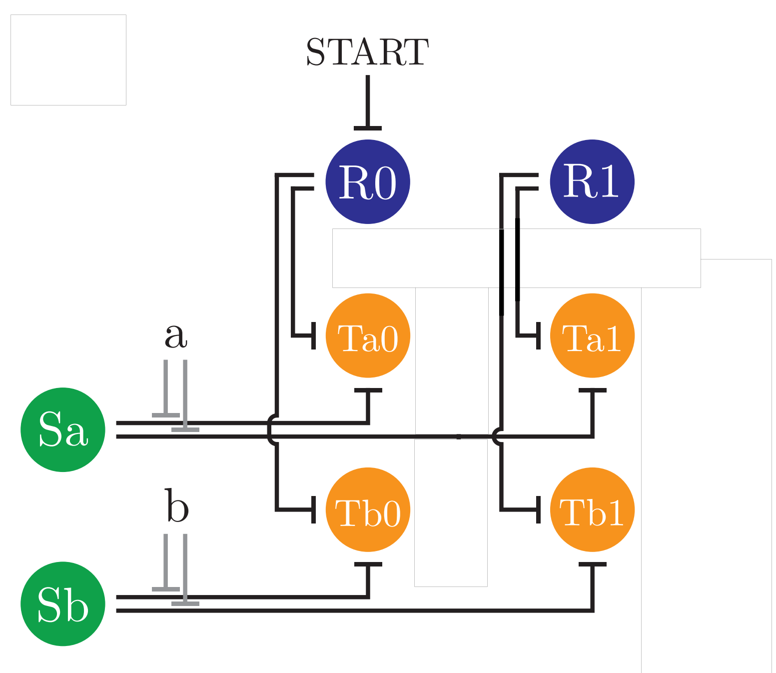
Specification:



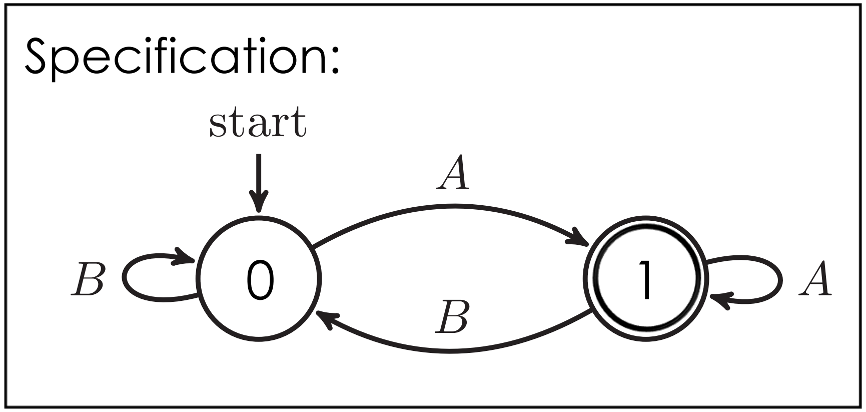
What is the Boolean function that describes the expression level of each transition gene at time t+1?

$$T_{a,0}^{t+1} = f_1(R_0^t, R_1^t, a^t, b^t)$$
$$T_{b,0}^{t+1} = f_2(R_0^t, R_1^t, a^t, b^t)$$
$$T_{a,1}^{t+1} = f_3(R_0^t, R_1^t, a^t, b^t)$$
$$T_{b,1}^{t+1} = f_4(R_0^t, R_1^t, a^t, b^t)$$



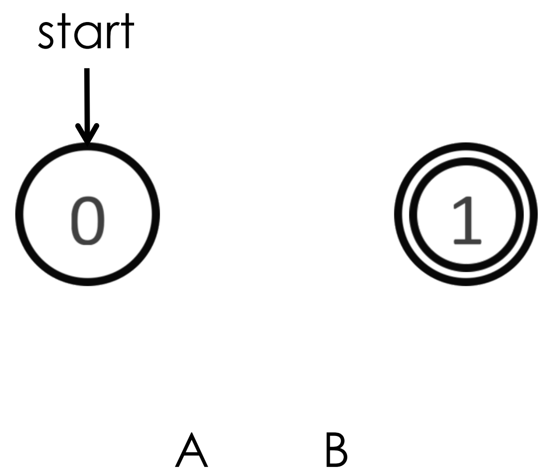


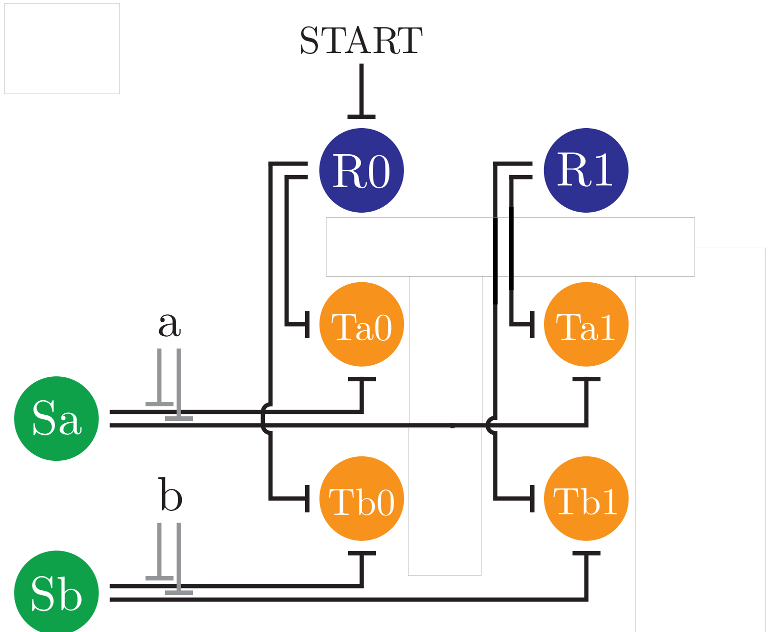
GRN General Construction Method



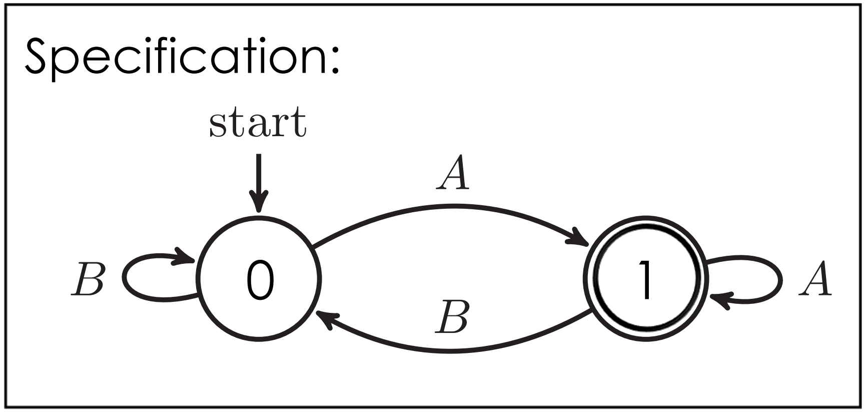
What is the Boolean function that describes the expression level of each transition gene at time t+1?

$$T_{a,0}^{t+1} = \neg R_0^t \wedge a^t$$
$$T_{b,0}^{t+1} = \neg R_0^t \wedge b^t$$
$$T_{a,1}^{t+1} = \neg R_1^t \wedge a^t$$
$$T_{b,1}^{t+1} = \neg R_1^t \wedge b^t$$

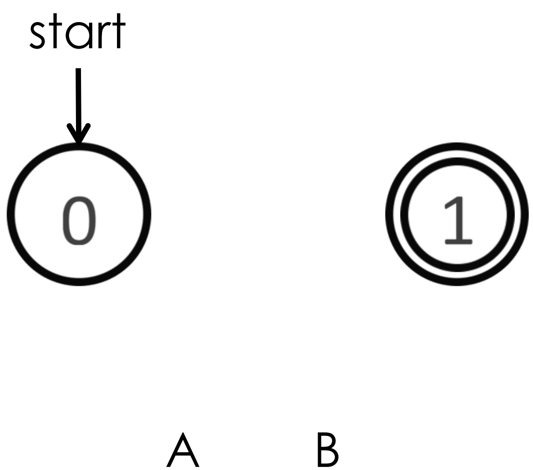


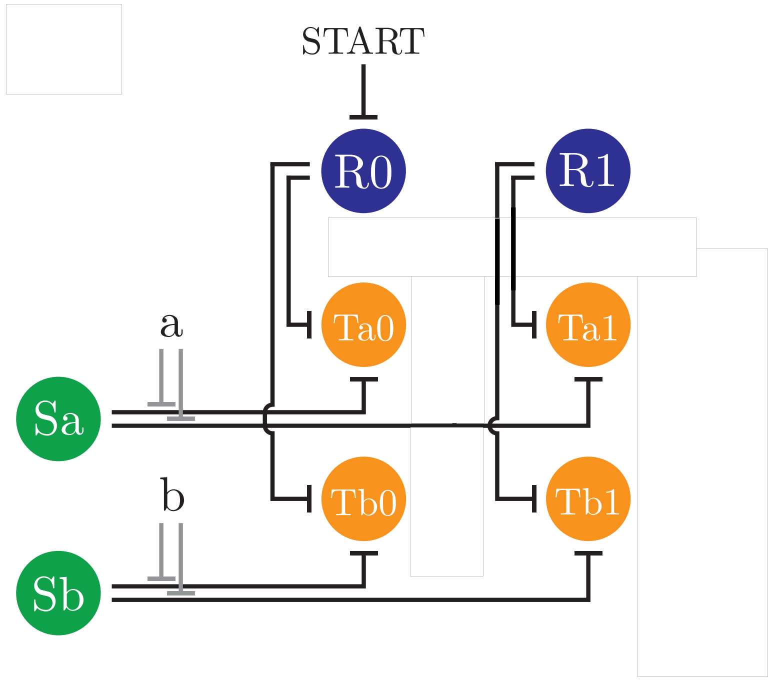


GRN General Construction Method

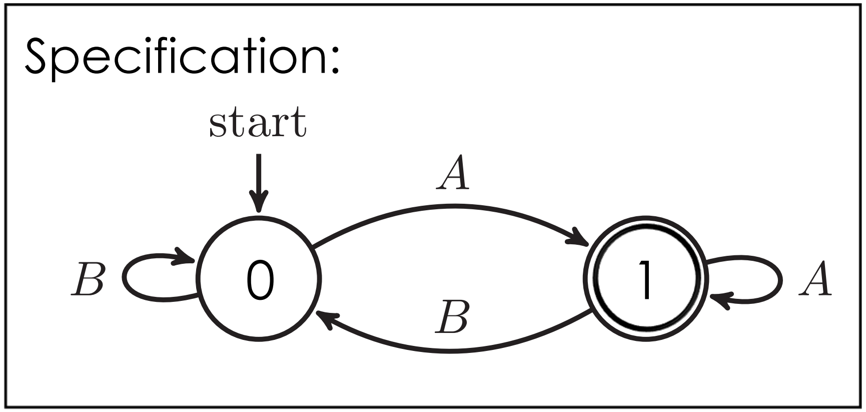


For each (q, σ, q') in Q x Σ x Q, such that δ (q, σ) q', wire T σq to Rq'.

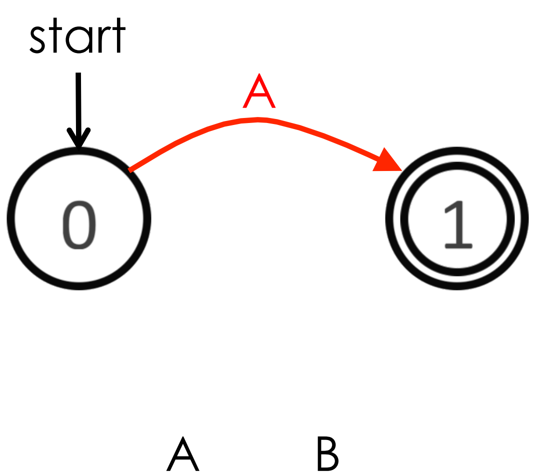


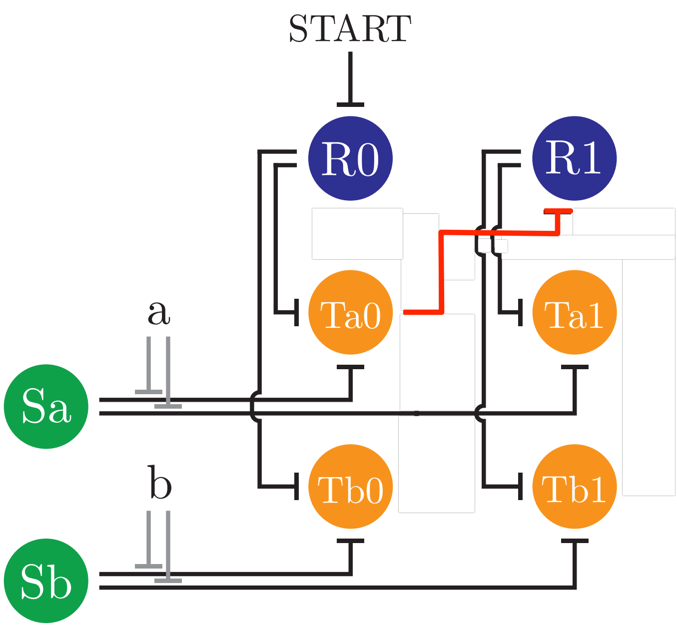


GRN General Construction Method



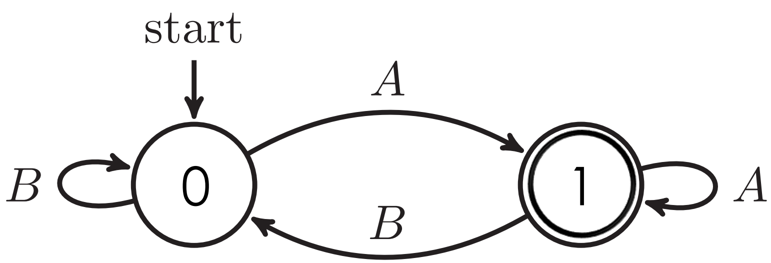
For each (q, σ, q') in Q x Σ x Q, such that δ (q, σ) q', wire Tσq to Rq'.



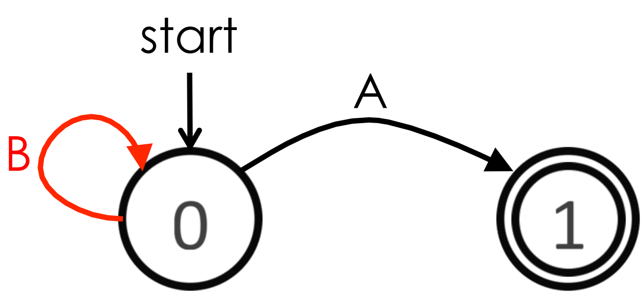


GRN General Construction Method

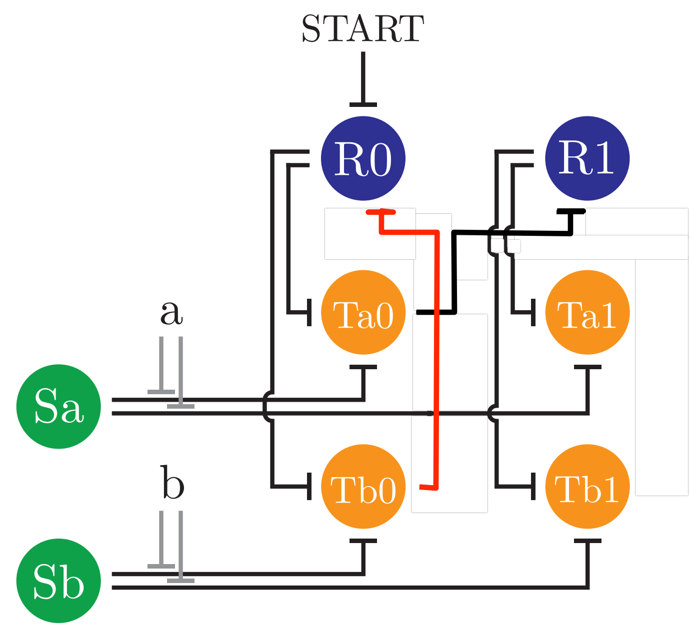
Specification:



For each (q, σ, q') in $Q \times \Sigma \times Q$, such that $\delta(q, \sigma)\ q'$, wire $T\sigma q$ to Rq'.

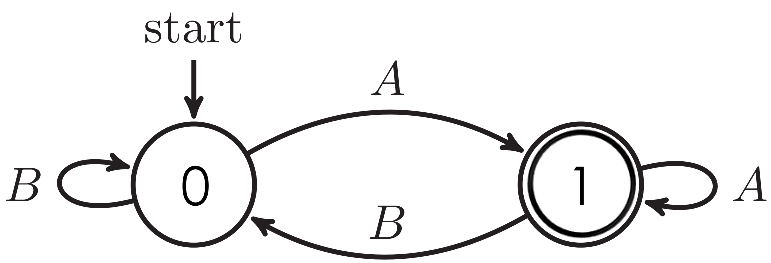


A B

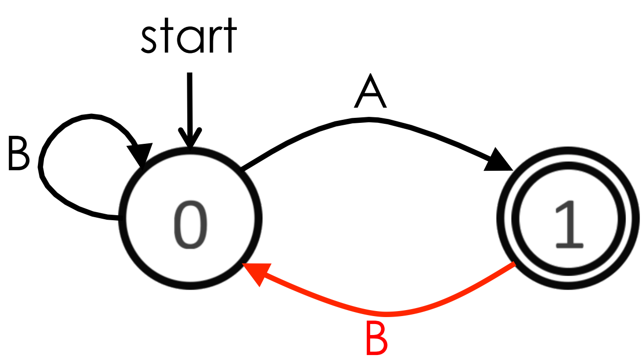


GRN General Construction Method

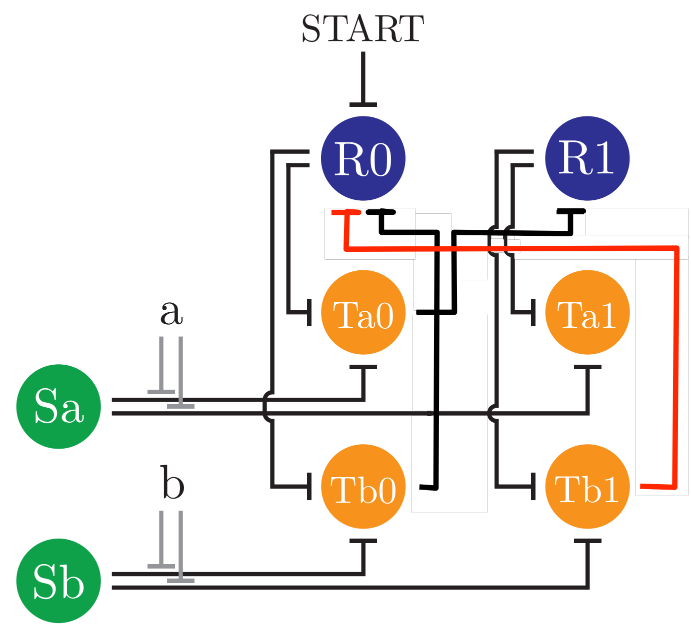
Specification:



For each (q, σ, q') in Q x Σ x Q, such that $\delta(q, \sigma)$ q', wire $T\sigma q$ to Rq'.

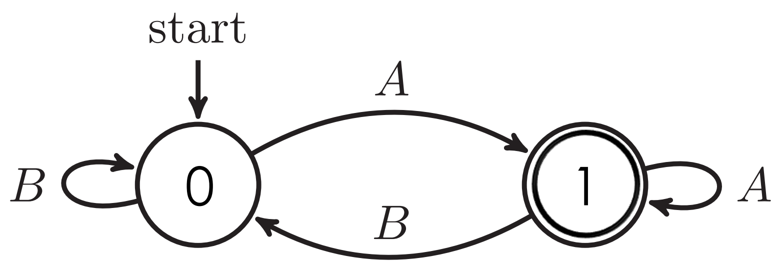


A B

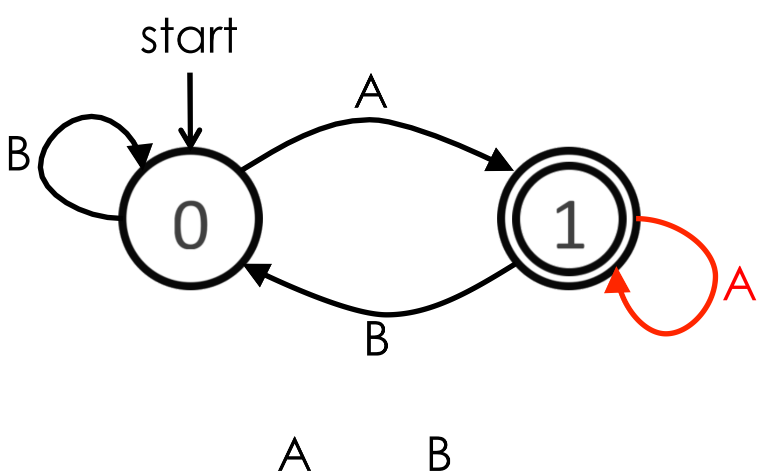


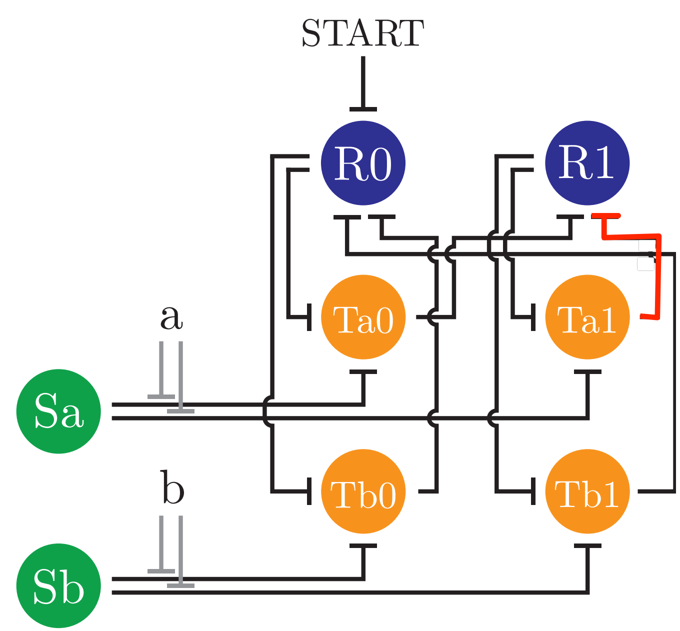
GRN General Construction Method

Specification:



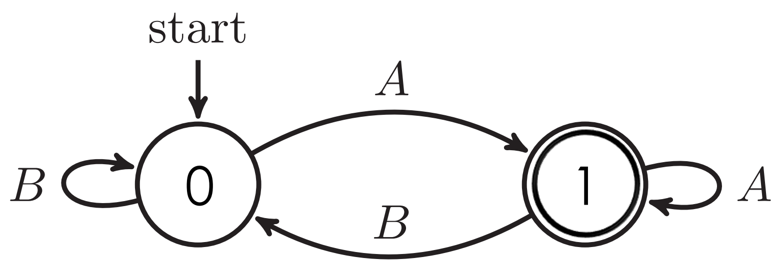
For each (q, σ, q') in Q x Σ x Q, such that δ (q, σ) q', wire Tσq to Rq'.





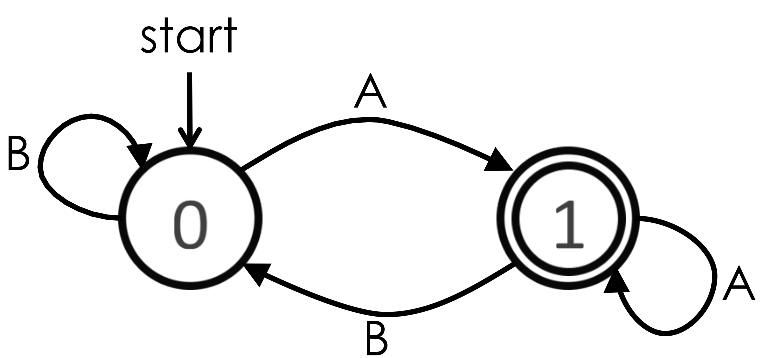
GRN General Construction Method

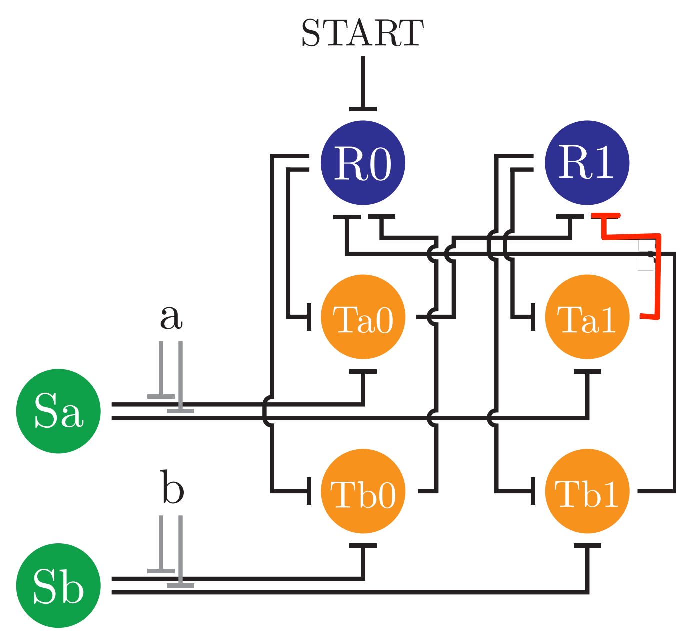
Specification:



For each (q, σ, q') in $Q \times \Sigma \times Q$, such that $\delta(q, \sigma)\ q'$, wire $T\sigma q$ to Rq'.

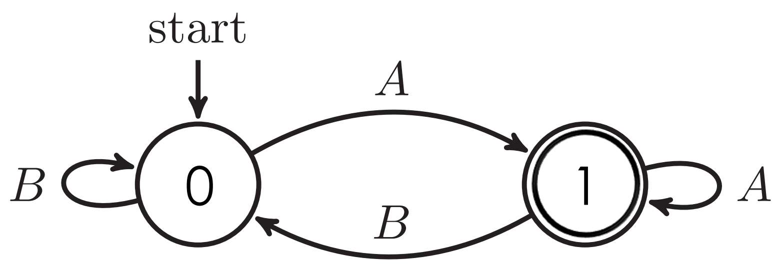
DONE! Almost…



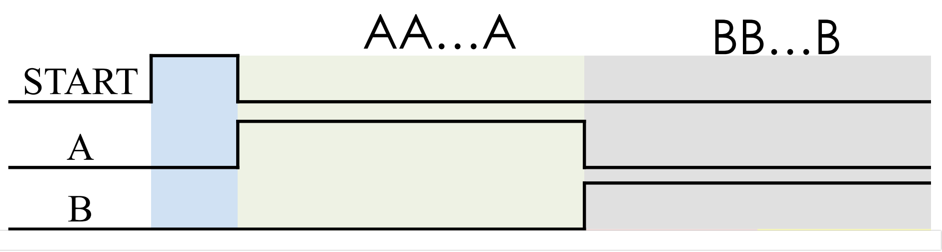


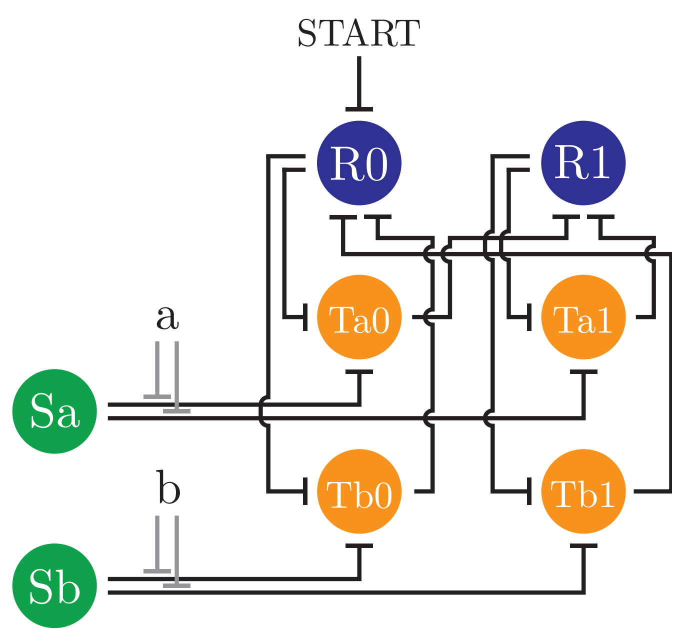
GRN General Construction Method

Specification:

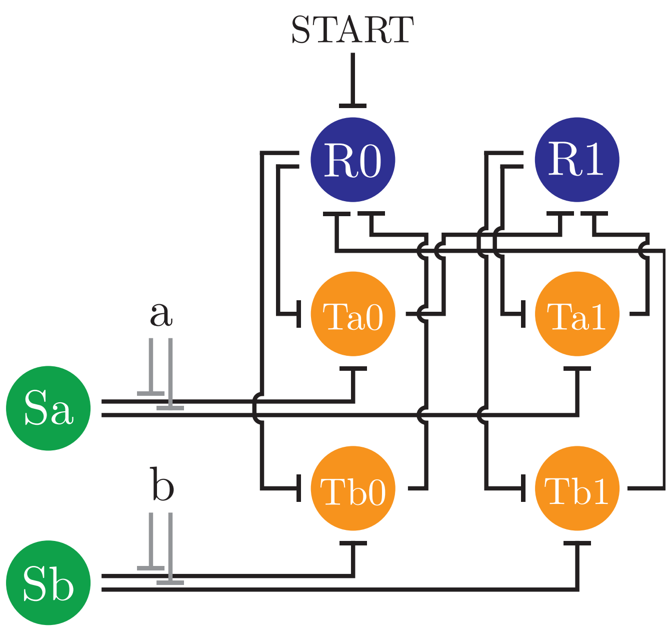


An input sequence w in Σ* is represented by a trajectory over START and the inducers, e.g.





Input trajectories for the BN Model of $g(M)$



Let $h_{BN}(w,t)$ be the input trajectory to $g(M)$ where $w=\sigma_{c1}\sigma_{c2}\ldots\sigma_{cm}$,

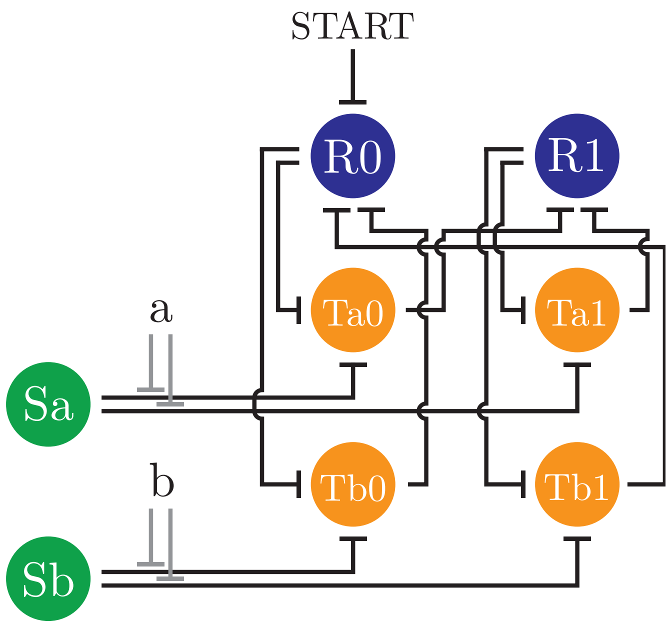
$$h_{BN}(w,t) = \begin{bmatrix} START^t \\ \sigma_1^t \\ \sigma_2^t \\ \vdots \\ \sigma_n^t \end{bmatrix}$$

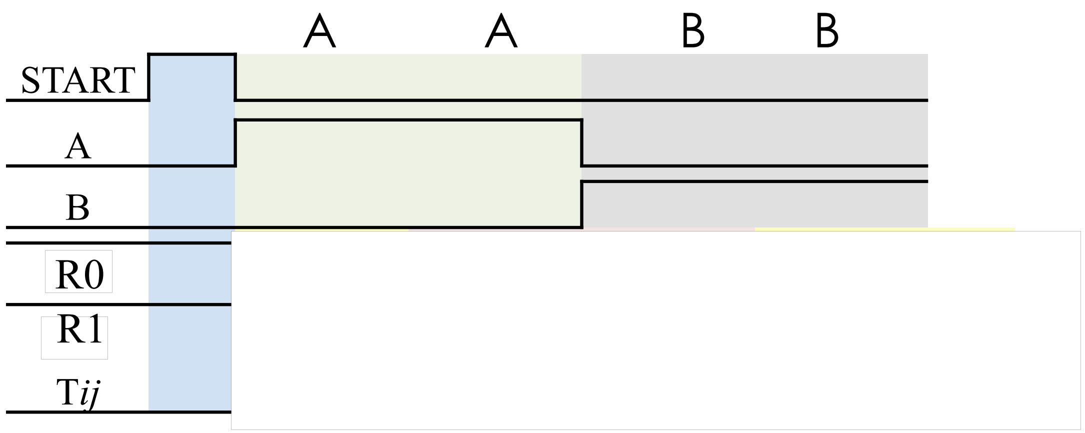
$$START^t = \begin{cases} on, & t \in \{0,1\} \\ off, & \text{otherwise}, \end{cases}$$

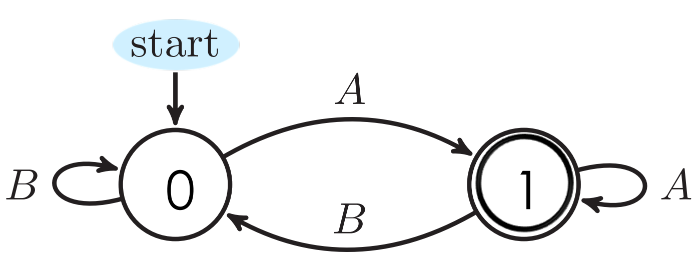
$$\sigma_j^t = \begin{cases} on, & \exists c_i \text{ s.t } j = c_i \text{ and } t \in \{2i, 2i+1\} \\ off, & \text{otherwise}. \end{cases}$$

What does the input trajectory for "AABB" look like?

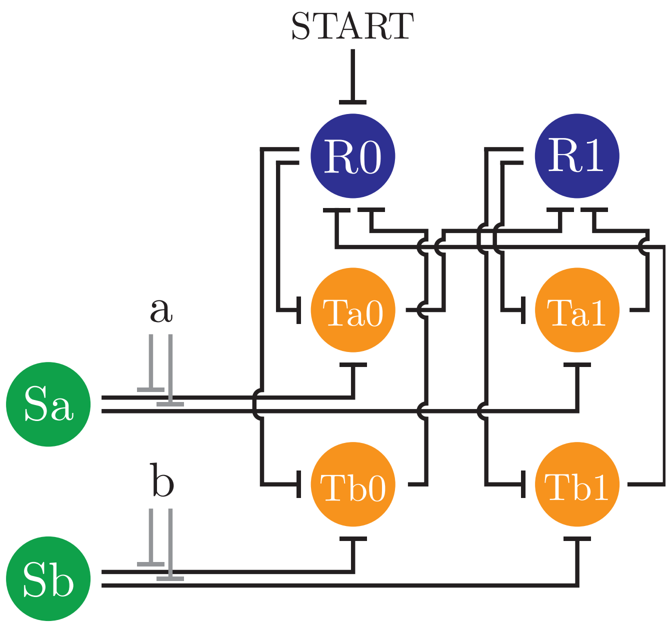
Example: Two-State Machine BN Model of $g(M)$

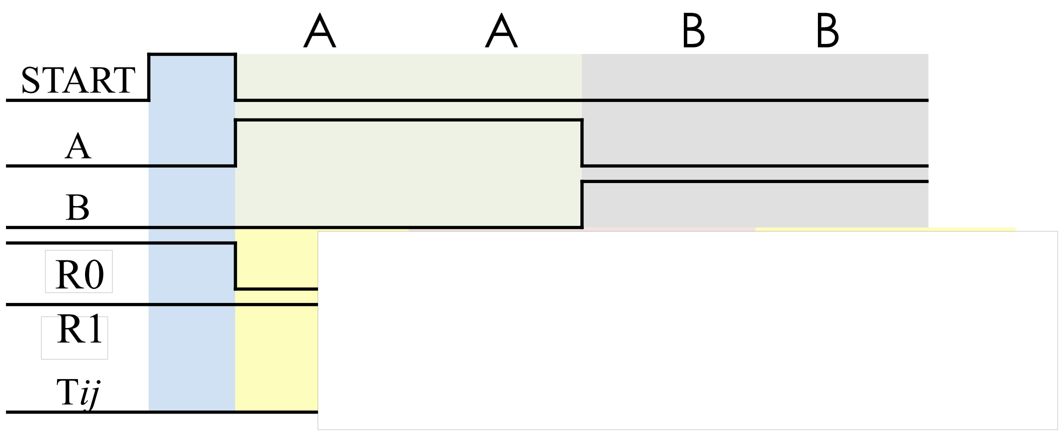


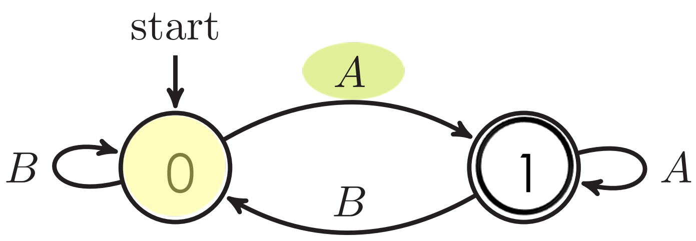




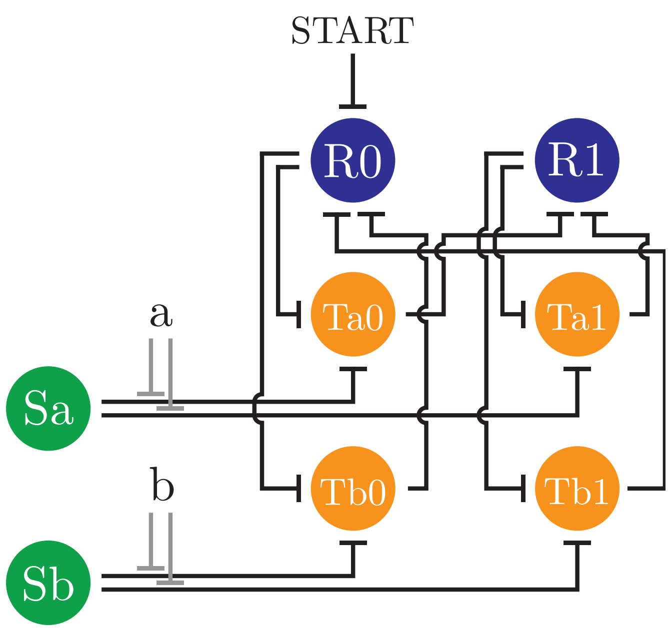
Example: Two-State Machine
BN Model of $g(M)$

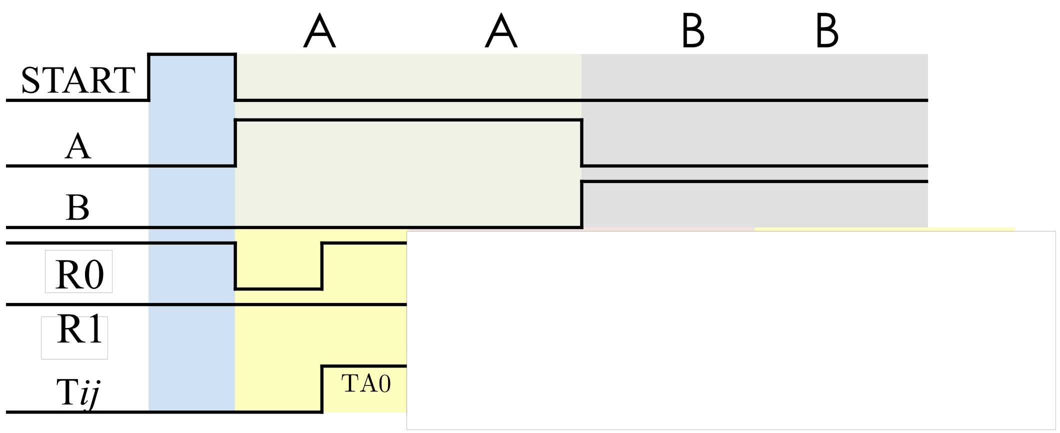


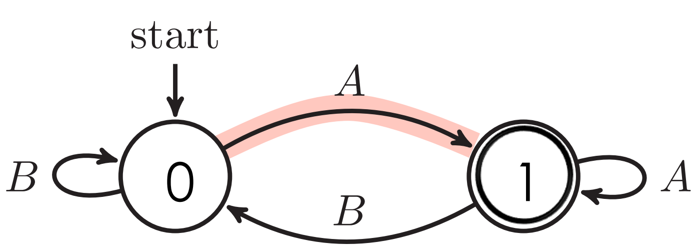




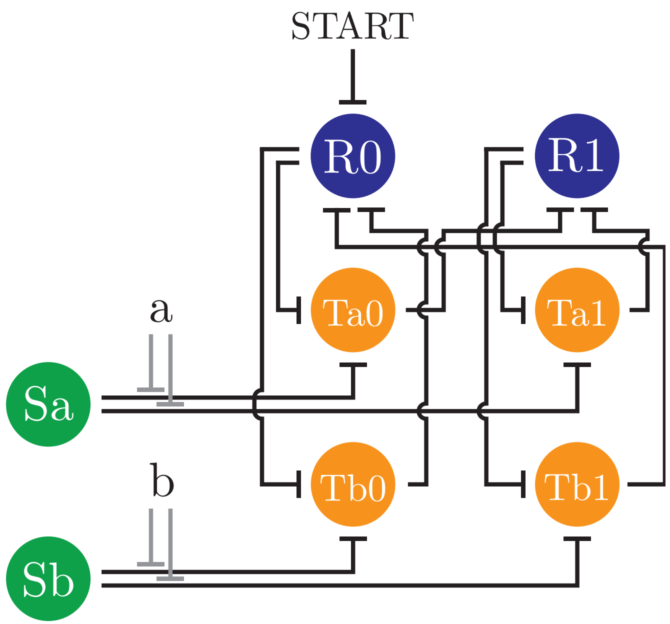
Example: Two-State Machine BN Model of $g(M)$

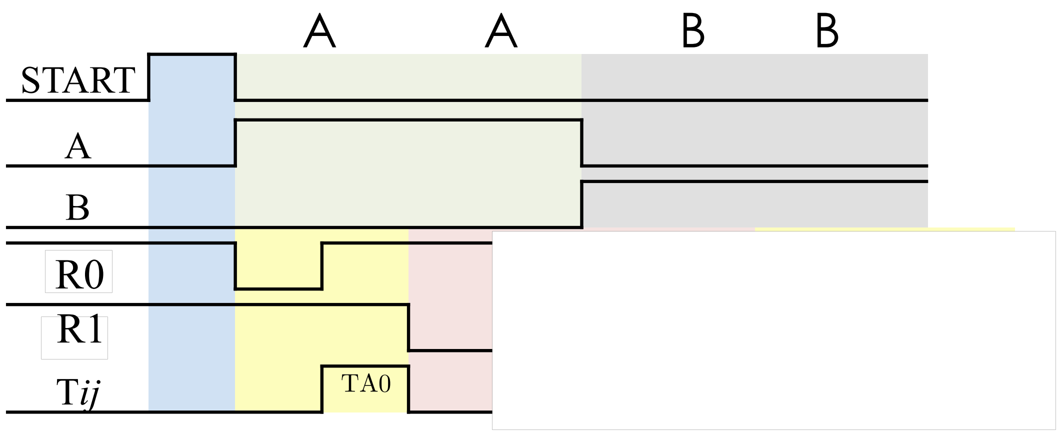


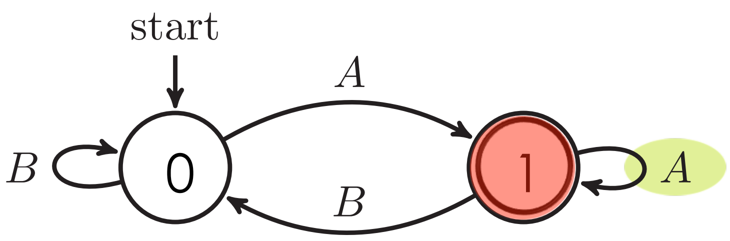




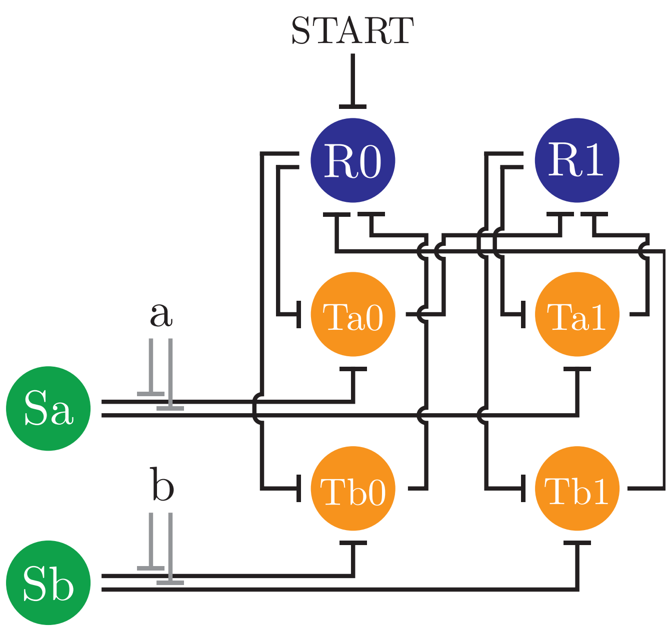
Example: Two-State Machine
BN Model of $g(M)$

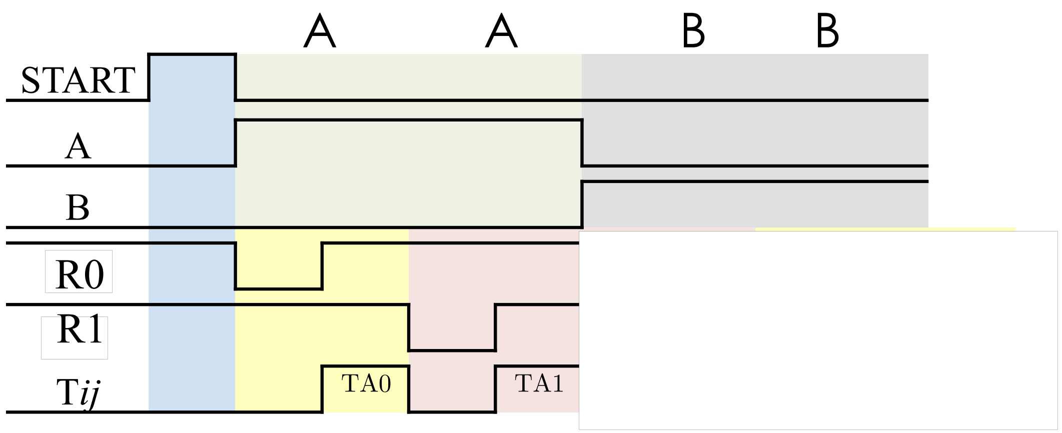


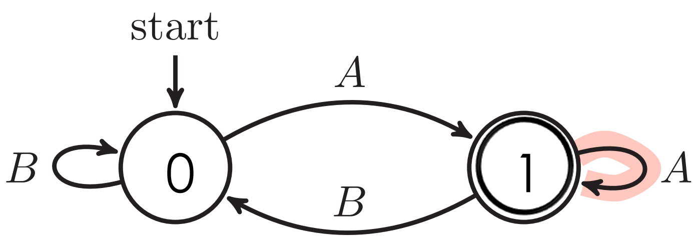




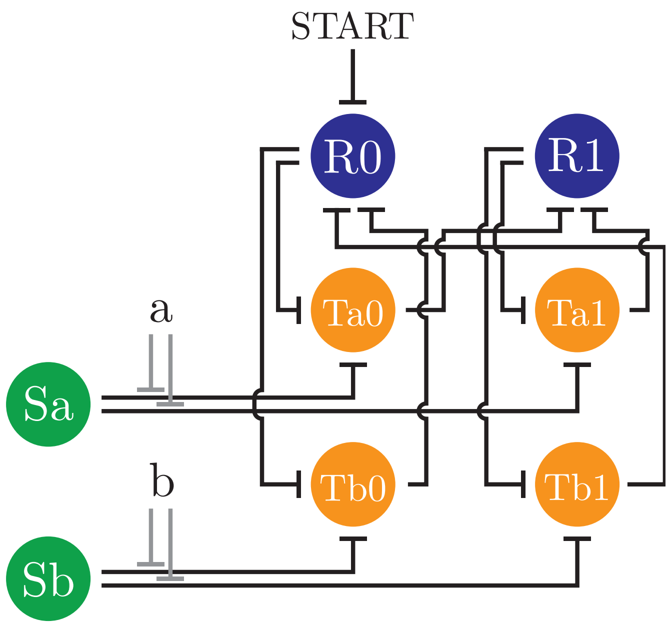
Example: Two-State Machine BN Model of $g(M)$

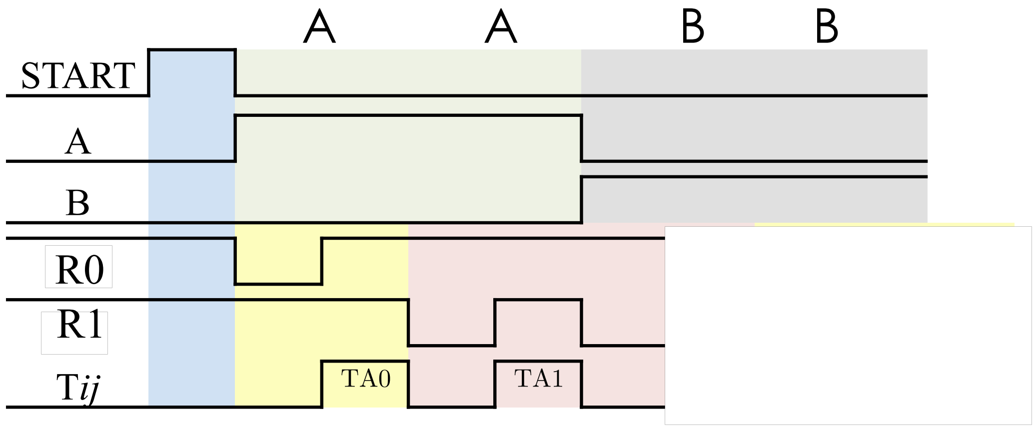


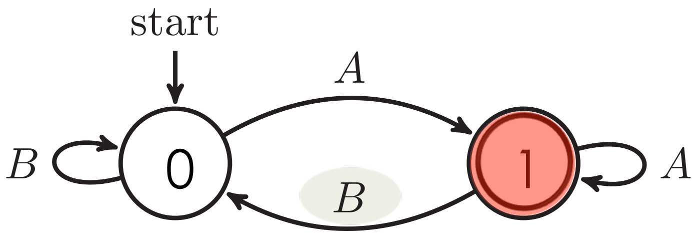




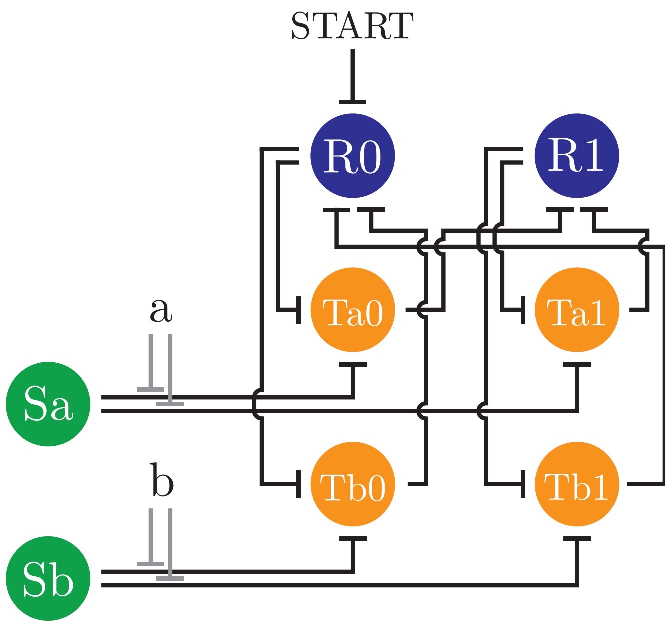
Example: Two-State Machine
BN Model of $g(M)$

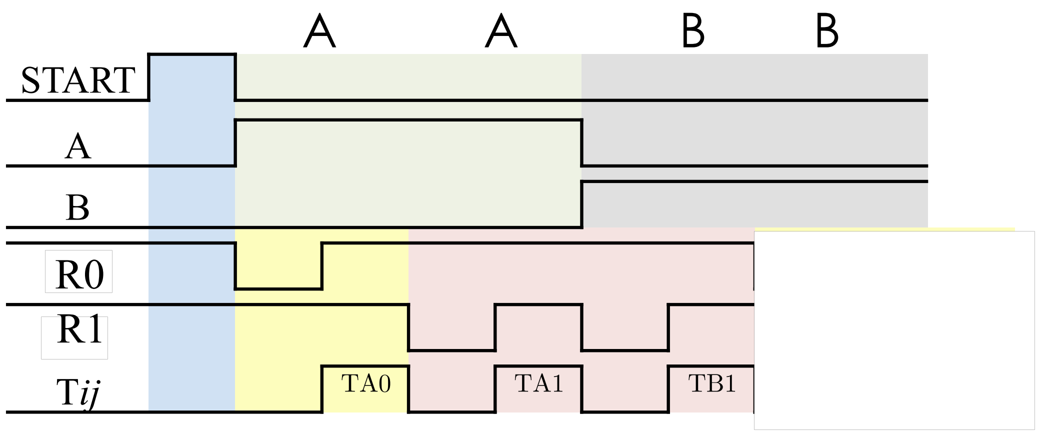


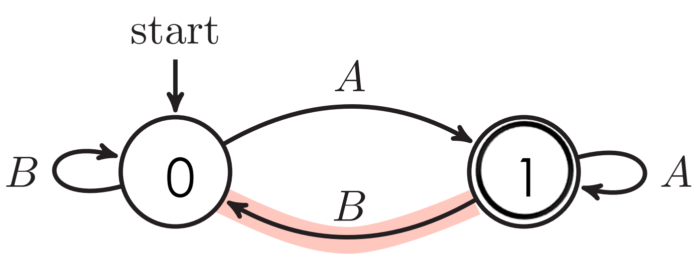




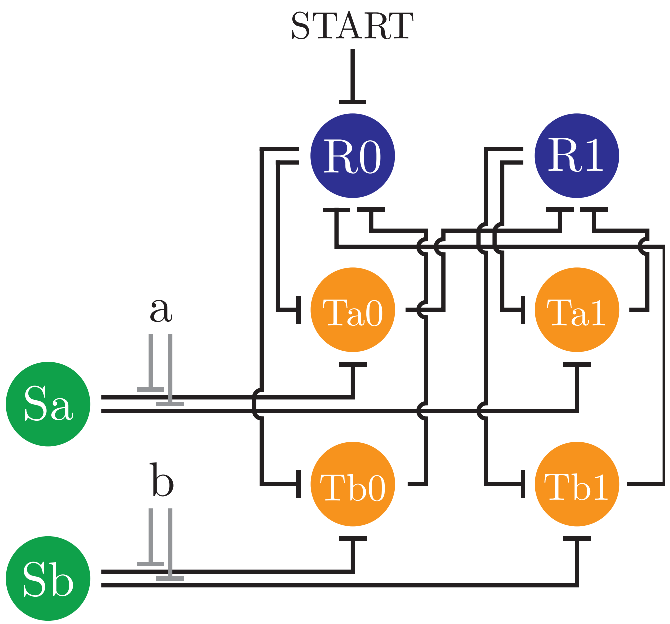
Example: Two-State Machine BN Model of $g(M)$

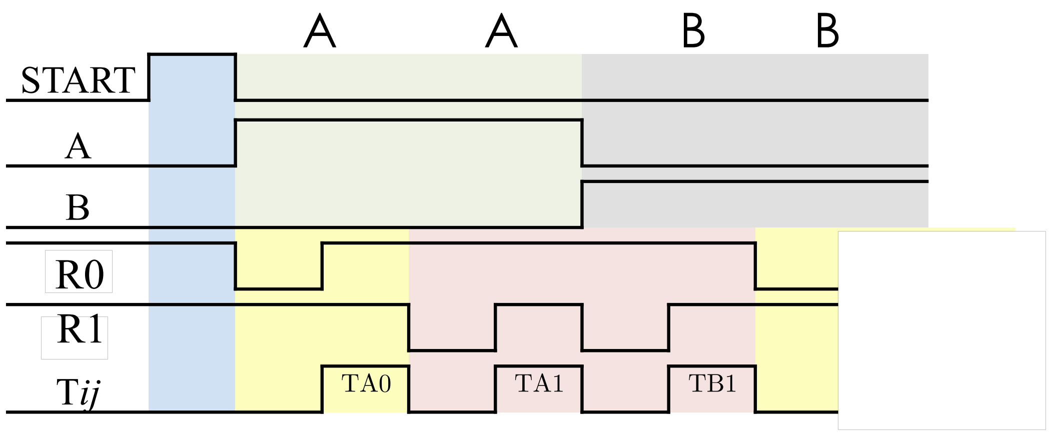


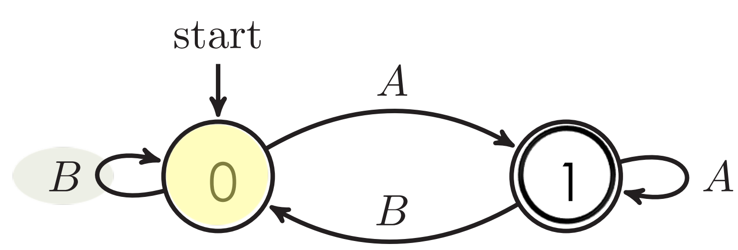




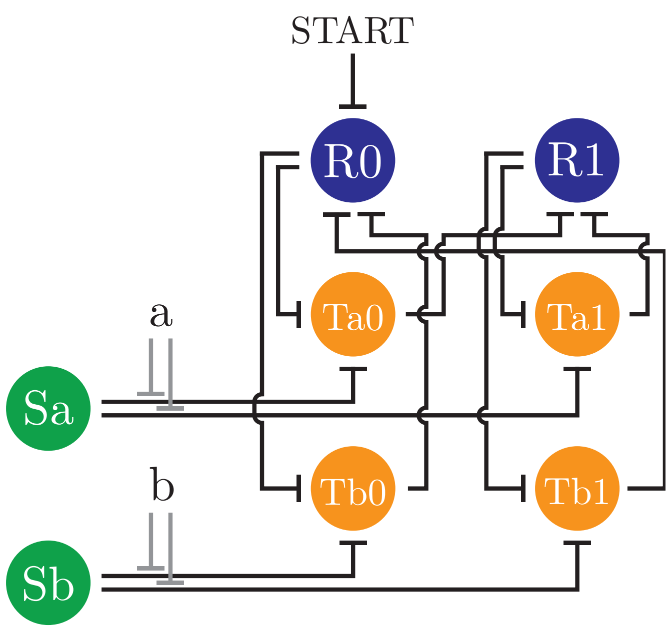
Example: Two-State Machine BN Model of $g(M)$

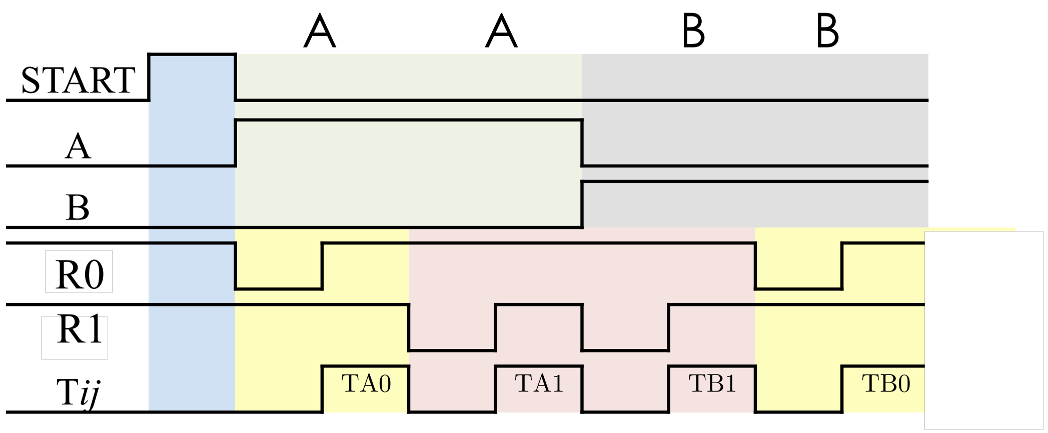


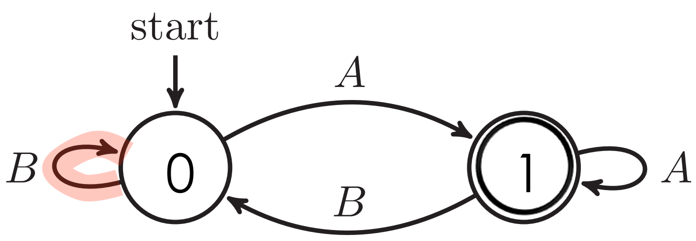




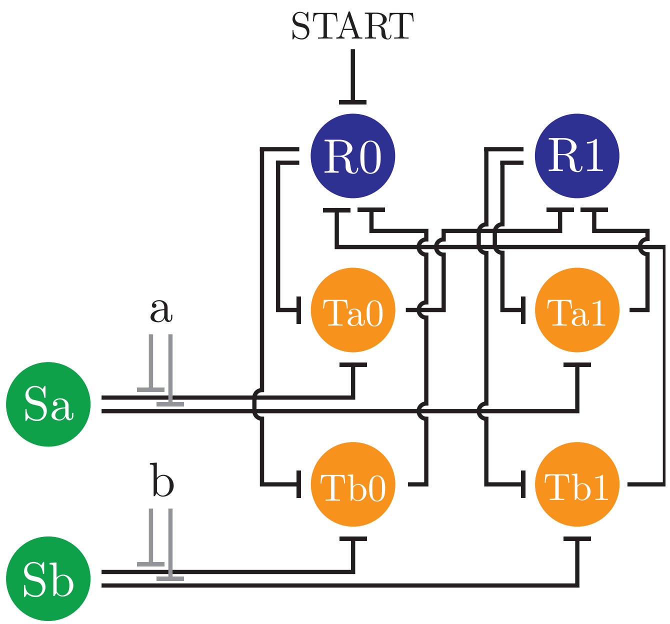
Example: Two-State Machine BN Model of $g(M)$

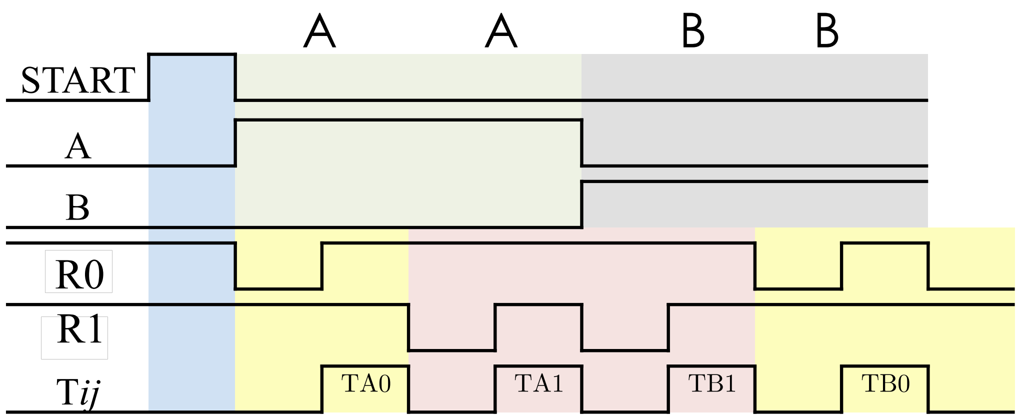


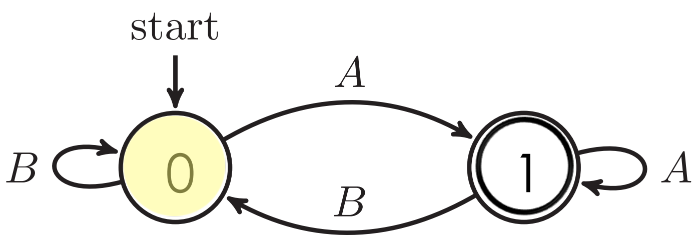




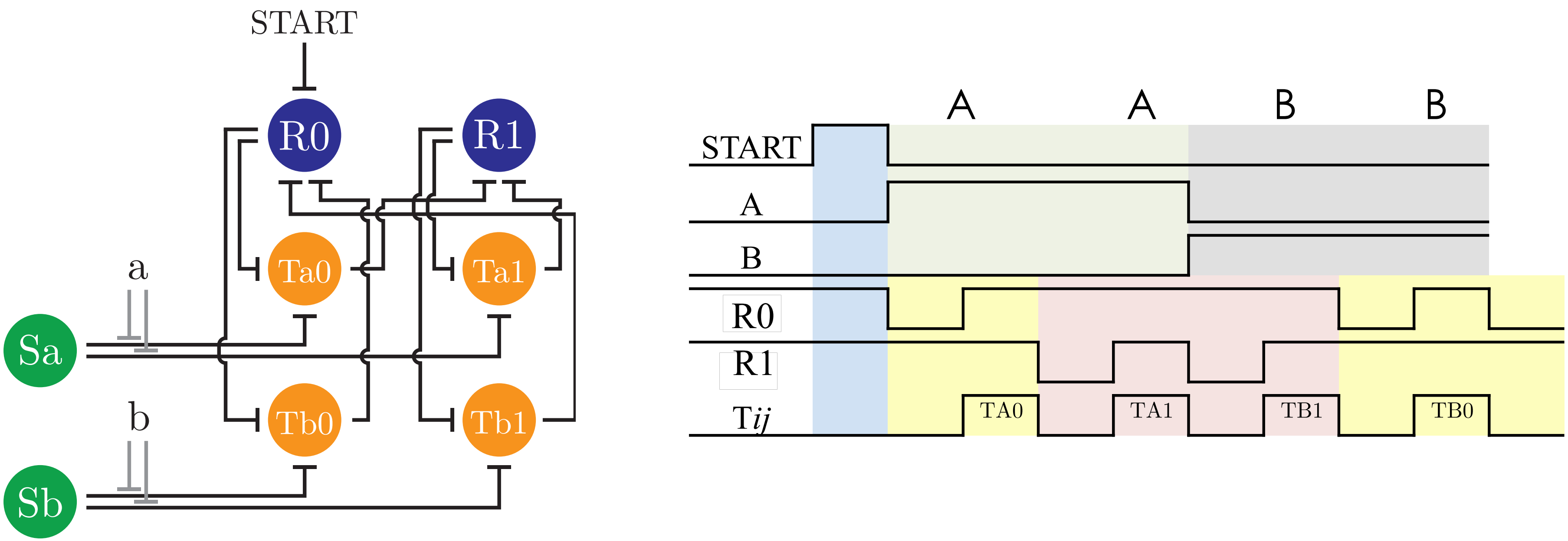
Example: Two-State Machine
BN Model of $g(M)$





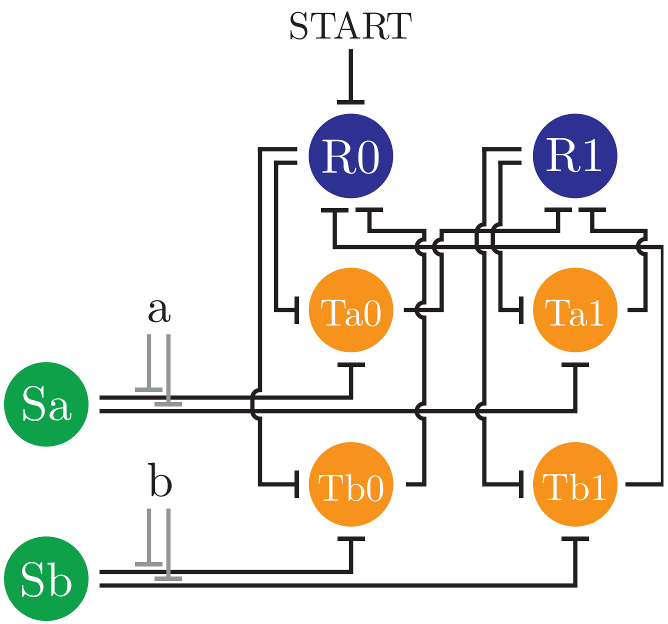


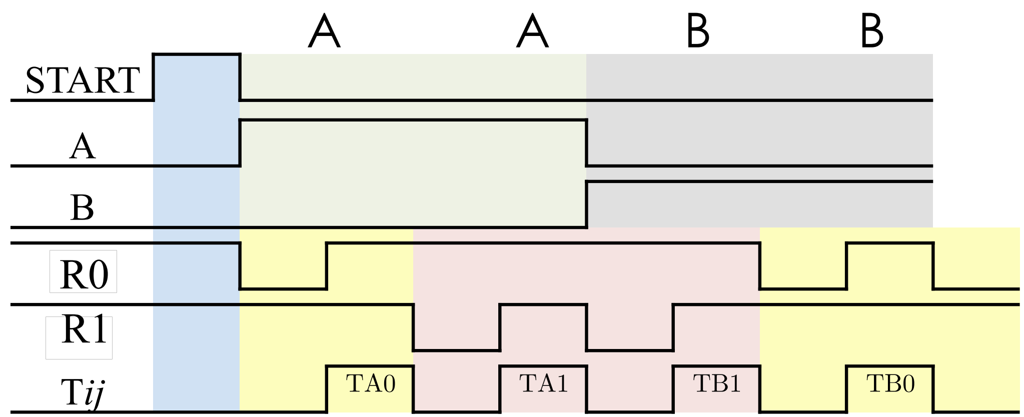
Example: Two-State Machine
BN Model of *g*(*M*)



Theorem. *Given a finite state machine M, the GRN g(M) simulates M when modeled as a Boolean network.*

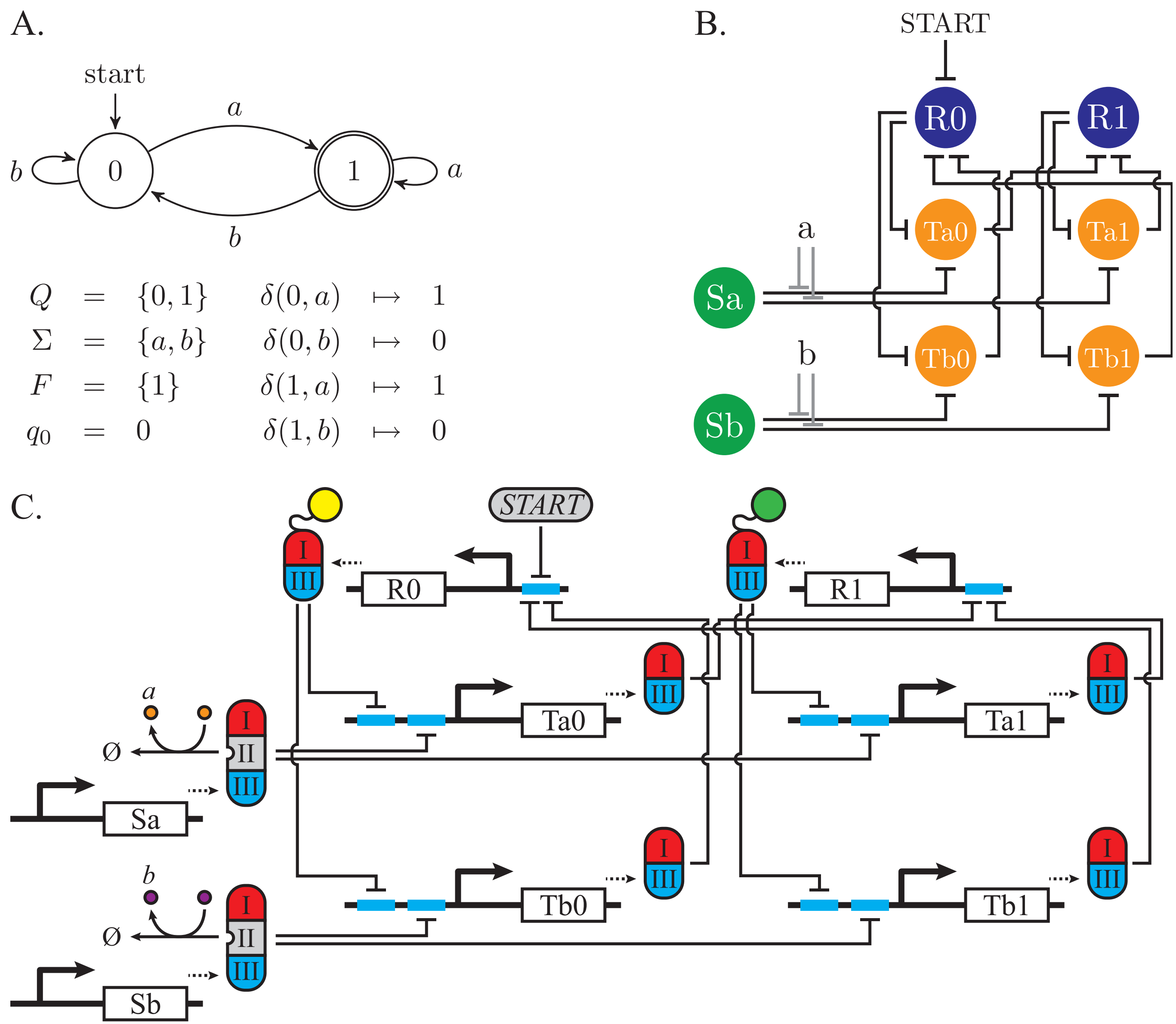
Example: Two-State Machine BN Model of $g(M)$



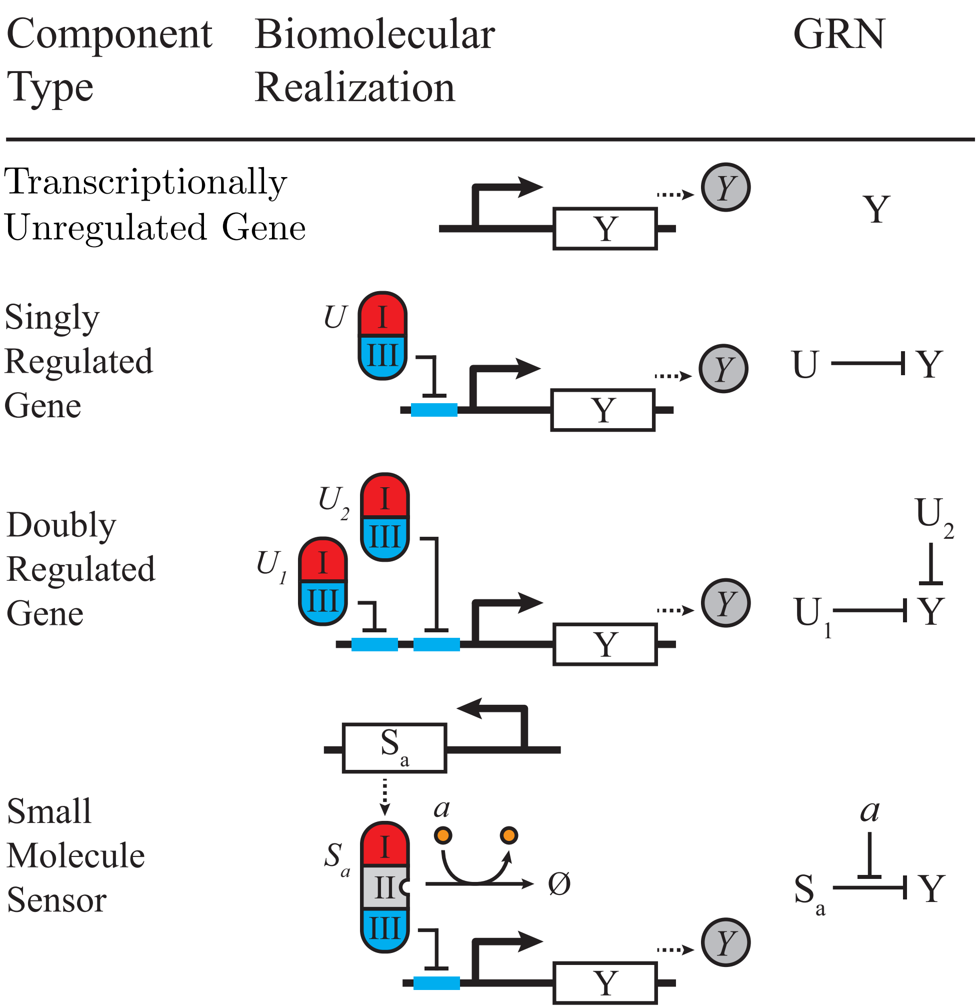


Theorem. *Assuming a Boolean network model, GRNs are computationally equivalent to FSMs.*

Representations of the Two-State Machine



Delay Differential Equations Model



Gene expression levels are NOT generally binary.

Continuous time model.

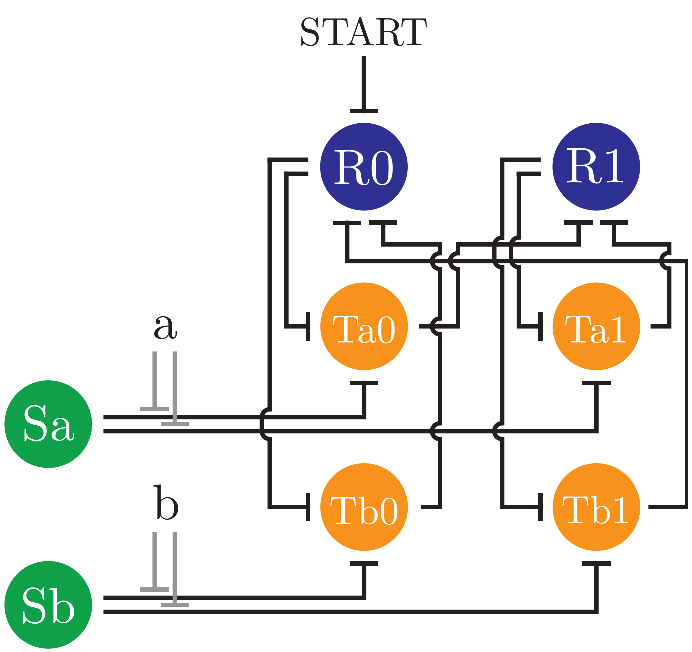
Study the effects of:
- production rate
- degradation rate
- dilution rate
- binding affinity

Delay Differential Equations Model

Syntax

$$G = (V, E_r, E_a)$$

Symbol	Meaning
V	set of *gene products* or *inducers*
$E_r \subset V \times (V \cup E_r \cup E_a)$	*repression relation*
$E_a \subset V \times (V \cup E_r \cup E_a)$	*activation relation*



Semantics

Delay Differential Equations

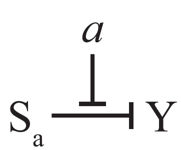
$$\frac{d}{dt} Y(t) = f(Y(t-\tau), U(t))$$

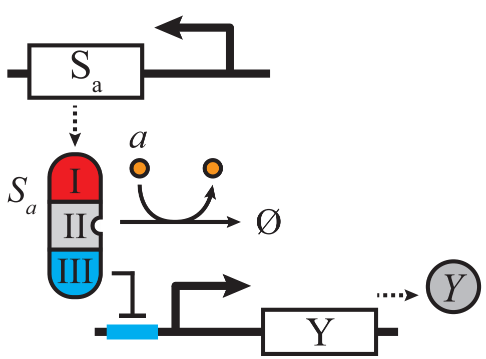
Let $Y(t)$ be a time-varying state vector and $U(t)$ be a time-varying input vector, i.e.,

$$Y^t = \begin{bmatrix} R_0(t) \\ R_1(t) \\ S_a(t) \\ S_b(t) \\ T_{a,0}(t) \\ T_{b,0}(t) \\ T_{a,1}(t) \\ T_{b,1}(t) \end{bmatrix} \quad U(t) = \begin{bmatrix} a(t) \\ b(t) \end{bmatrix}$$

Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$



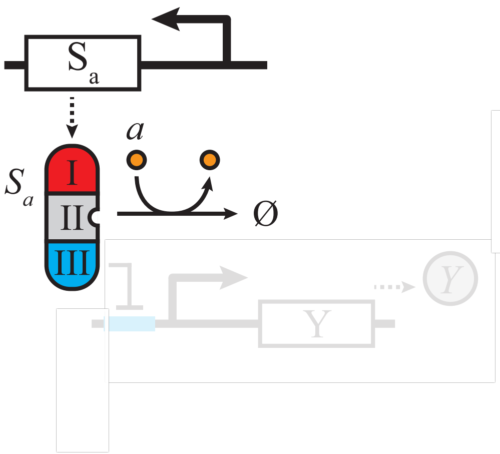


$$
\begin{aligned}
\frac{d}{dt}S_a(t) &= V_{max} - (\beta + k_p a(t))S_a(t) \\
\frac{d}{dt}Y(t) &= \frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)
\end{aligned}
$$

Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt}S_a(t) = \underbrace{V_{max}}_{\text{Production}} - \underbrace{(\beta + k_p a(t))}_{\text{Degradation}} S_a(t)$$

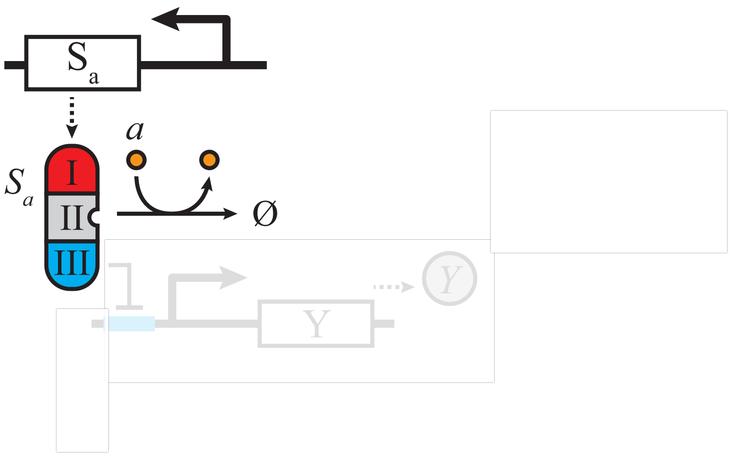


Can you interpret this as a chemical reaction network?

Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt} S_a(t) = \underbrace{V_{max}}_{\text{Production}} - \underbrace{(\beta + k_p a(t))}_{\text{Degradation}} S_a(t)$$



Can you interpret this as a chemical reaction network?

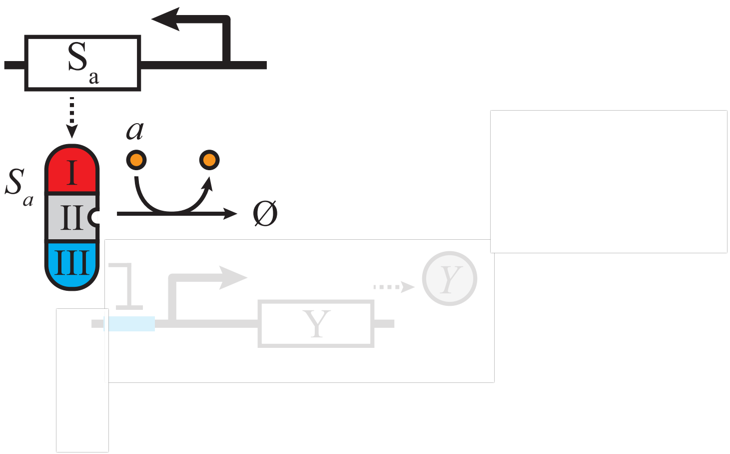
$$\emptyset \xrightarrow{V_{max}} S_a \xrightarrow{\beta + k_p a(t)} \emptyset$$

Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt}S_a(t) = V_{max} - (\beta + k_p a(t))S_a(t)$$

$$0 = V_{max} - (\beta + k_p a^*)S_a^*$$



$$\emptyset \xrightarrow{V_{max}} S_a \xrightarrow{\beta + k_p a(t)} \emptyset$$

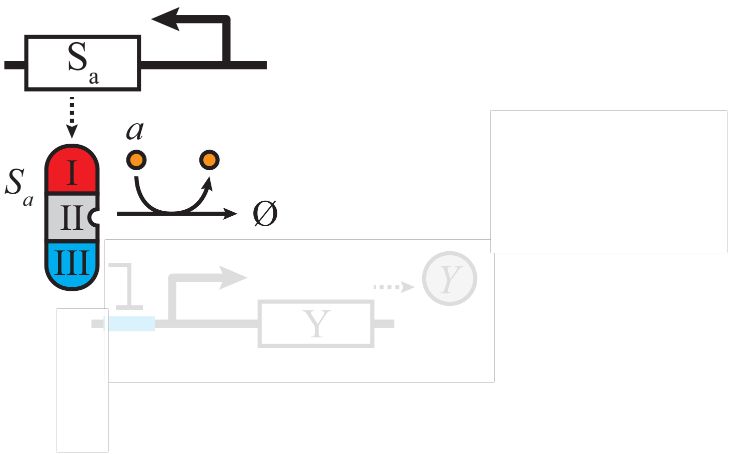
Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt} S_a(t) = V_{max} - (\beta + k_p a(t)) S_a(t)$$

$$0 = V_{max} - (\beta + k_p a^*) S_a^*$$

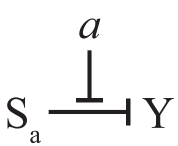
$$\boxed{S_a^* = \frac{V_{max}}{\beta + k_p a^*} = \frac{\text{production}}{\text{degradation}}}$$



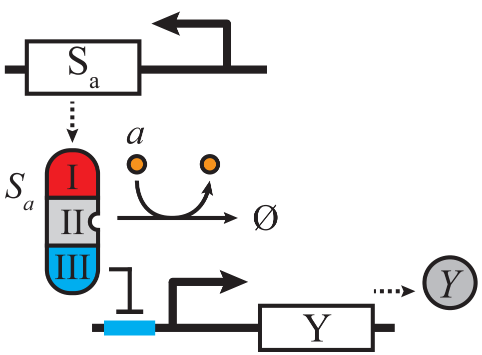
$$\emptyset \xrightarrow{V_{max}} S_a \xrightarrow{\beta + k_p a(t)} \emptyset$$

Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$



Original set of equations:



$$
\begin{aligned}
\frac{d}{dt} S_a(t) &= V_{max} - (\beta + k_p a(t)) S_a(t) \\
\frac{d}{dt} Y(t) &= \frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}} \right)^n} - \beta Y(t)
\end{aligned}
$$

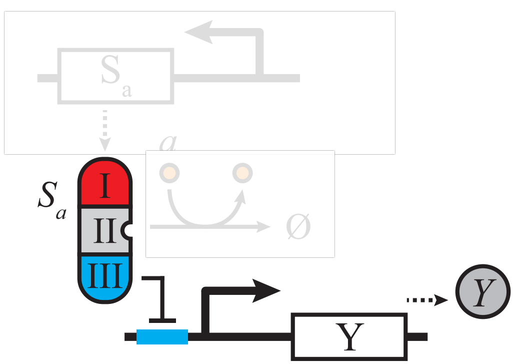
Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt}Y(t) = \underbrace{\frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n}}_{\text{Production}} - \underbrace{\beta}_{\text{Degradation}} Y(t)$$

Production

Degradation



What happens when $S_a^* = 0$?

$S_a^* \to \infty$?

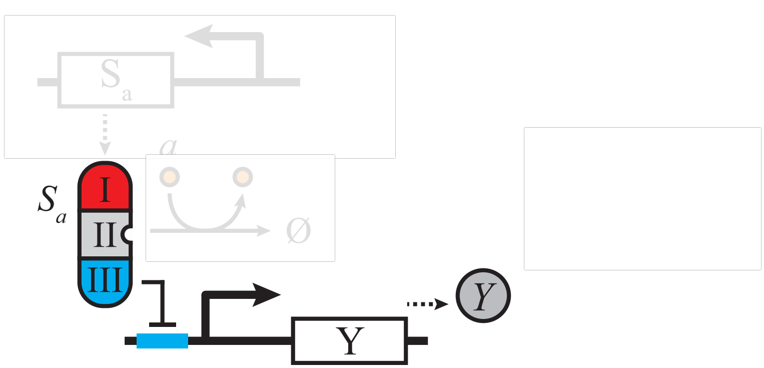
$S_a^* = k_{1/2}$?

Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt}Y(t) = \frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)$$

$$0 = \frac{V_{max}}{1 + \left(\frac{S_a^*}{k_{1/2}}\right)^n} - \beta Y^*$$

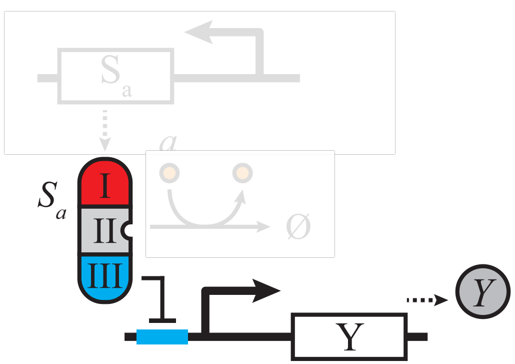


Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt}Y(t) = \frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)$$

$$S_a^* = 0 \qquad\qquad 0 = \frac{V_{max}}{1 + \left(\frac{S_a^*}{k_{1/2}}\right)^n} - \beta Y^*$$



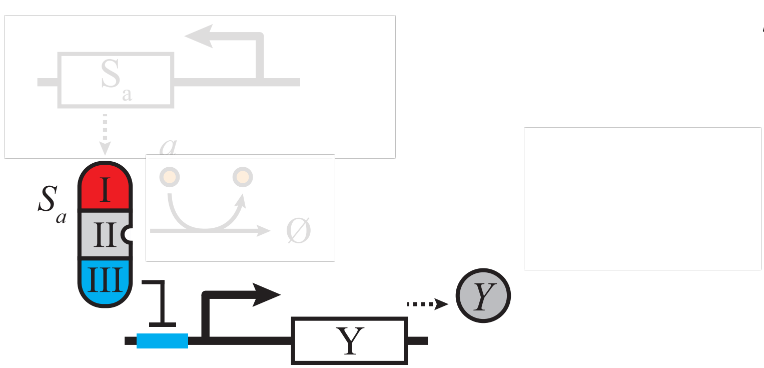
Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt}Y(t) = \frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)$$

$$S_a^* = 0 \qquad\qquad 0 = \frac{V_{max}}{1 + \cancel{\left(\frac{S_a^*}{k_{1/2}}\right)^n}} - \beta Y^*$$

$$= V_{max} - \beta Y^*$$



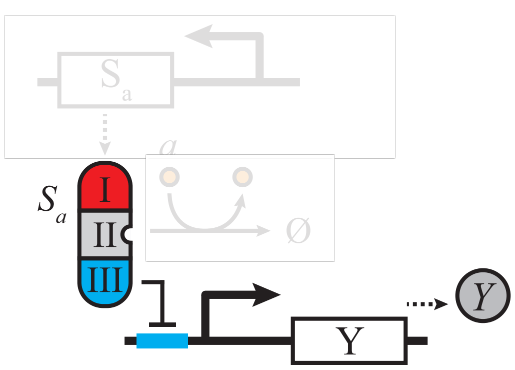
$$\boxed{Y^* = \frac{V_{max}}{\beta} = Y^*_{max}}$$

Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt}Y(t) = \frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)$$

$$S_a^* \to \infty \qquad 0 = \frac{V_{max}}{1 + \left(\frac{S_a^*}{k_{1/2}}\right)^n} - \beta Y^*$$



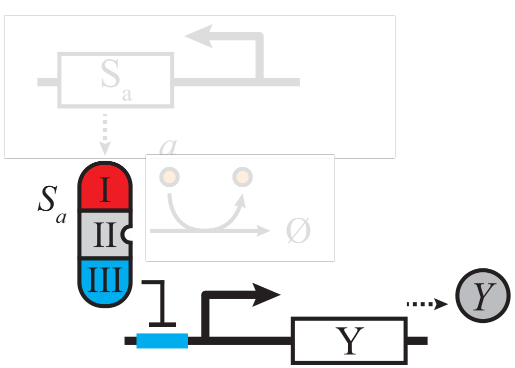
Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt}Y(t) = \frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)$$

$$S_a^* \to \infty \qquad 0 = \cancel{\frac{V_{max}}{1 + \left(\frac{S_a^*}{k_{1/2}}\right)^n}} - \beta Y^*$$

$$= 0 - \beta Y^*$$



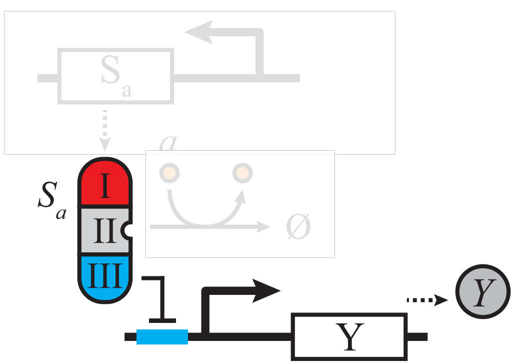
$$\boxed{Y^* = 0}$$

Delay Differential Equations Model

V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt}Y(t) = \frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)$$

$$S_a^* = k_{1/2} \qquad 0 = \frac{V_{max}}{1 + \left(\frac{S_a^*}{k_{1/2}}\right)^n} - \beta Y^*$$



Delay Differential Equations Model

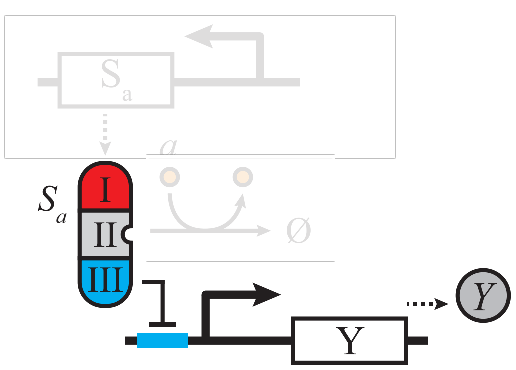
V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$

$$\frac{d}{dt}Y(t) = \frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)$$

$$S_a^* = k_{1/2} \qquad 0 = \frac{V_{max}}{1 + \left(\frac{S_a^*}{k_{1/2}}\right)^n} - \beta Y^*$$

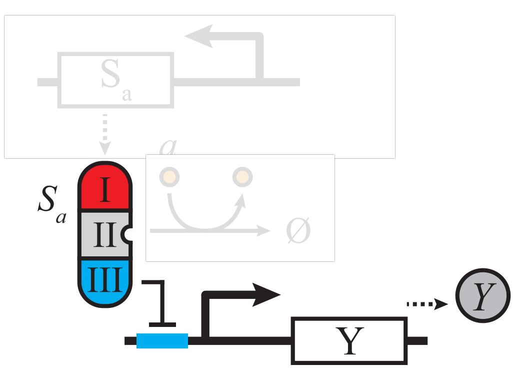
$$= \frac{V_{max}}{2} - \beta Y^*$$

$$\boxed{Y^* = \frac{1}{2} \cdot \frac{V_{max}}{\beta} = \frac{1}{2} Y^*_{max}}$$

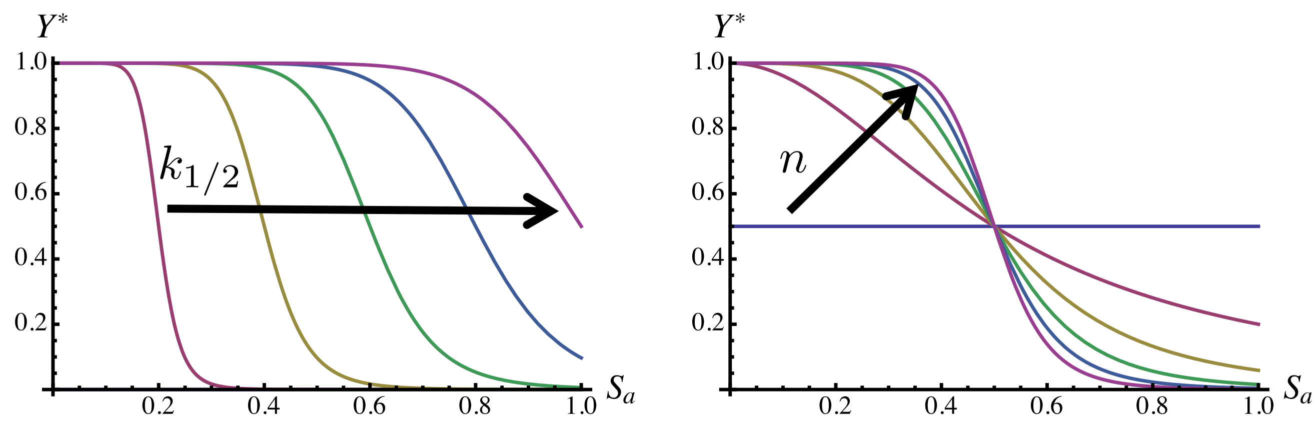


Delay Differential Equations Model

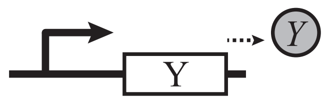
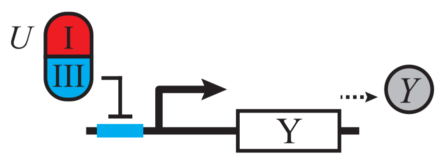
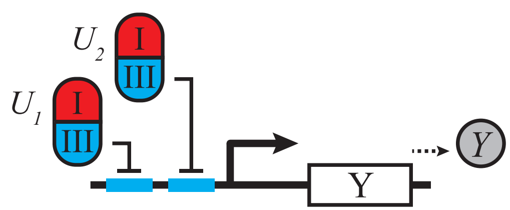
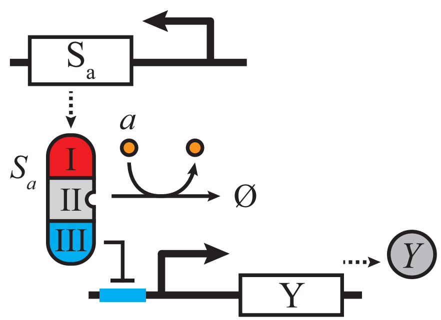
V_{max}, β	protein production and degradation rates	$V_{max} = \beta$
k_p	small molecule binding affinity	$k_p \gg \beta$
$k_{1/2}$	input for half-maximum gene production	
n	Hill coefficient	
τ	time delay, approximates transcription/translation dynamics	$\tau = 1$



Dose Response for $V_{max} = \beta = 1$



Delay Differential Equations Model

Component Type	Biomolecular Realization	GRN	Delay Differential Equations
Transcriptionally Unregulated Gene		Y	$\frac{d}{dt}Y(t) = V_{max} - \beta Y(t)$
Singly Regulated Gene		$U \dashv Y$	$\frac{d}{dt}Y(t) = \frac{V_{max}}{1+\left(\frac{U(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)$
Doubly Regulated Gene		$U_1 \dashv Y$, $U_2 \dashv Y$	$\frac{d}{dt}Y(t) = \frac{V_{max}}{1+\left(\frac{U_1(t-\tau)+U_2(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)$
Small Molecule Sensor		$a \dashv (S_a \dashv Y)$	$\frac{d}{dt}S_a(t) = V_{max} - (\beta + k_p a(t))S_a(t)$ $\frac{d}{dt}Y(t) = \frac{V_{max}}{1+\left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)$

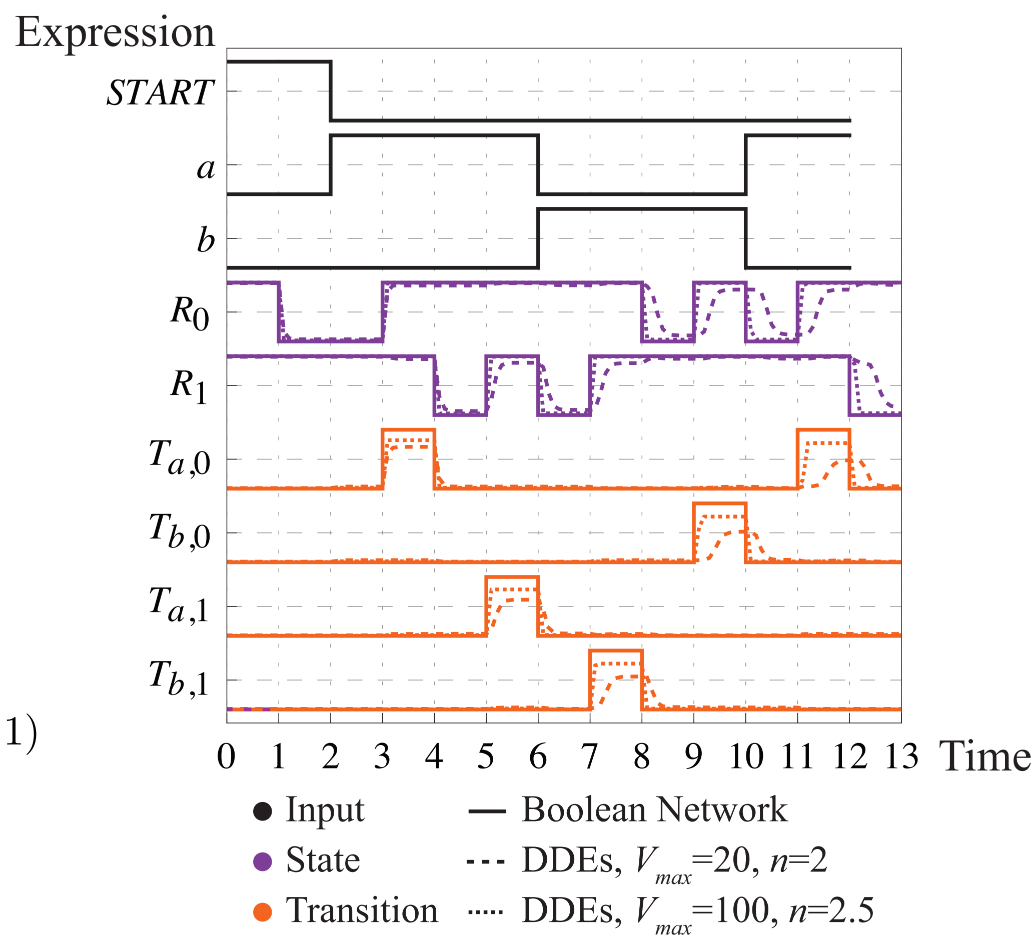
Input trajectories for the DDE Model of $g(M)$

Let $h_{DDE}(w, \Delta t, t)$ be the input trajectory to DDE model of $g(M)$ with pulse width Δt, where $w = \sigma_{c1}\sigma_{c2}\ldots\sigma_{cm}$,

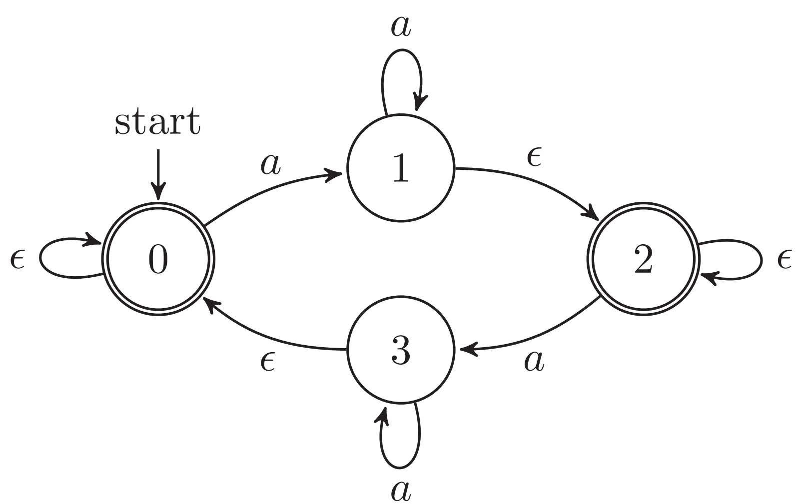
$$h_{DDE}(w,t) = \begin{bmatrix} START(t/\Delta t) \\ \sigma_1(t/\Delta t) \\ \sigma_2(t/\Delta t) \\ \vdots \\ \sigma_n(t/\Delta t) \end{bmatrix}$$

$$START(t) = \begin{cases} 1, & t \in [0,1) \\ 0, & \text{otherwise} \end{cases}$$

$$\sigma_j(t) = \begin{cases} 1, & \exists c_i \text{ s.t. } j = c_i \text{ and } t \in [2i, 2i+1) \\ 0, & \text{otherwise.} \end{cases}$$



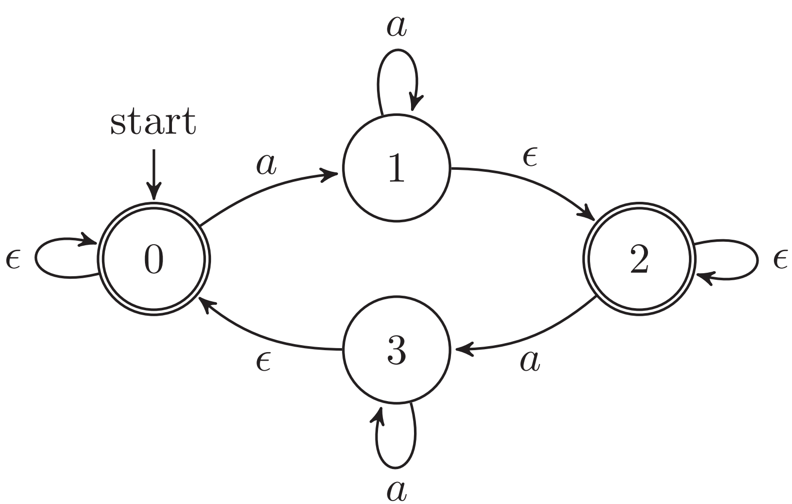
Example II: Modulo-Two Pulse Counter

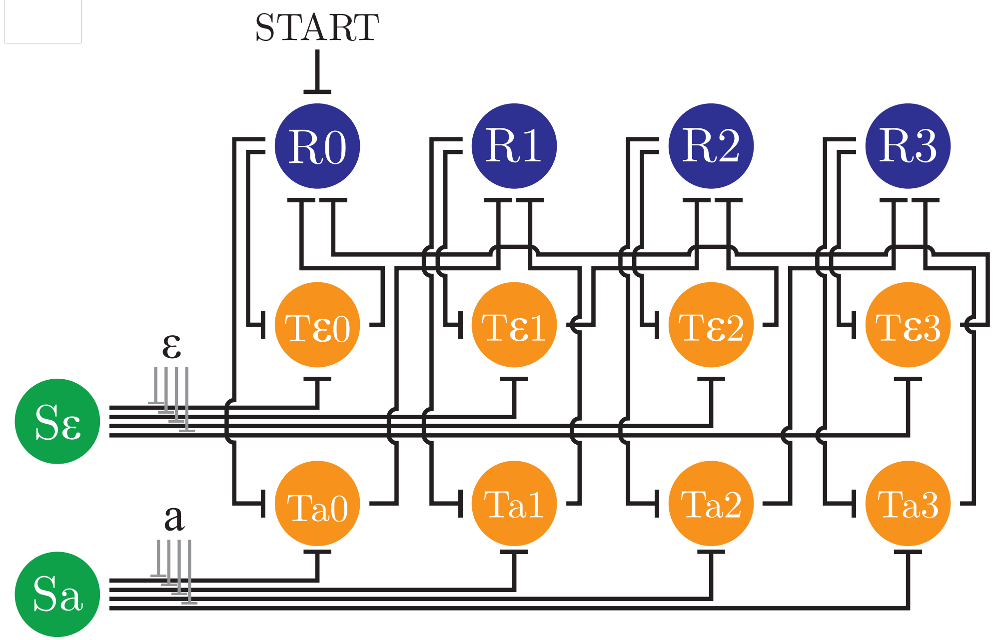


***ε*-symbol**. Applied whenever another input symbol is not supplied.

Input	Final State
$aaaa\ldots aaaaaaaaaaaa\ldots a\ldots$	$q = 2$
$a\ldots$	$q = 2$
$a\ldots aaaaaaaa\ldots$	$q = 0$

Example II: Modulo-Two Pulse Counter



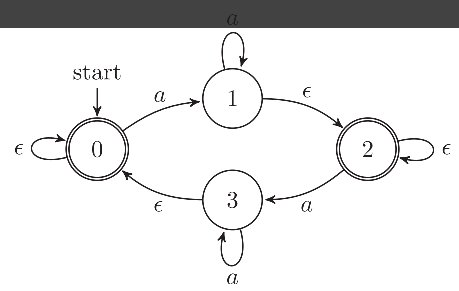


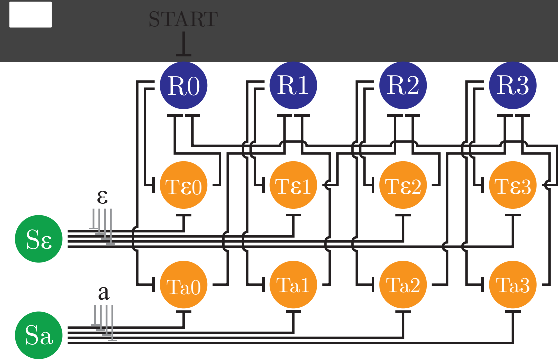
ε-symbol. Applied at any time step where another input symbol is not supplied

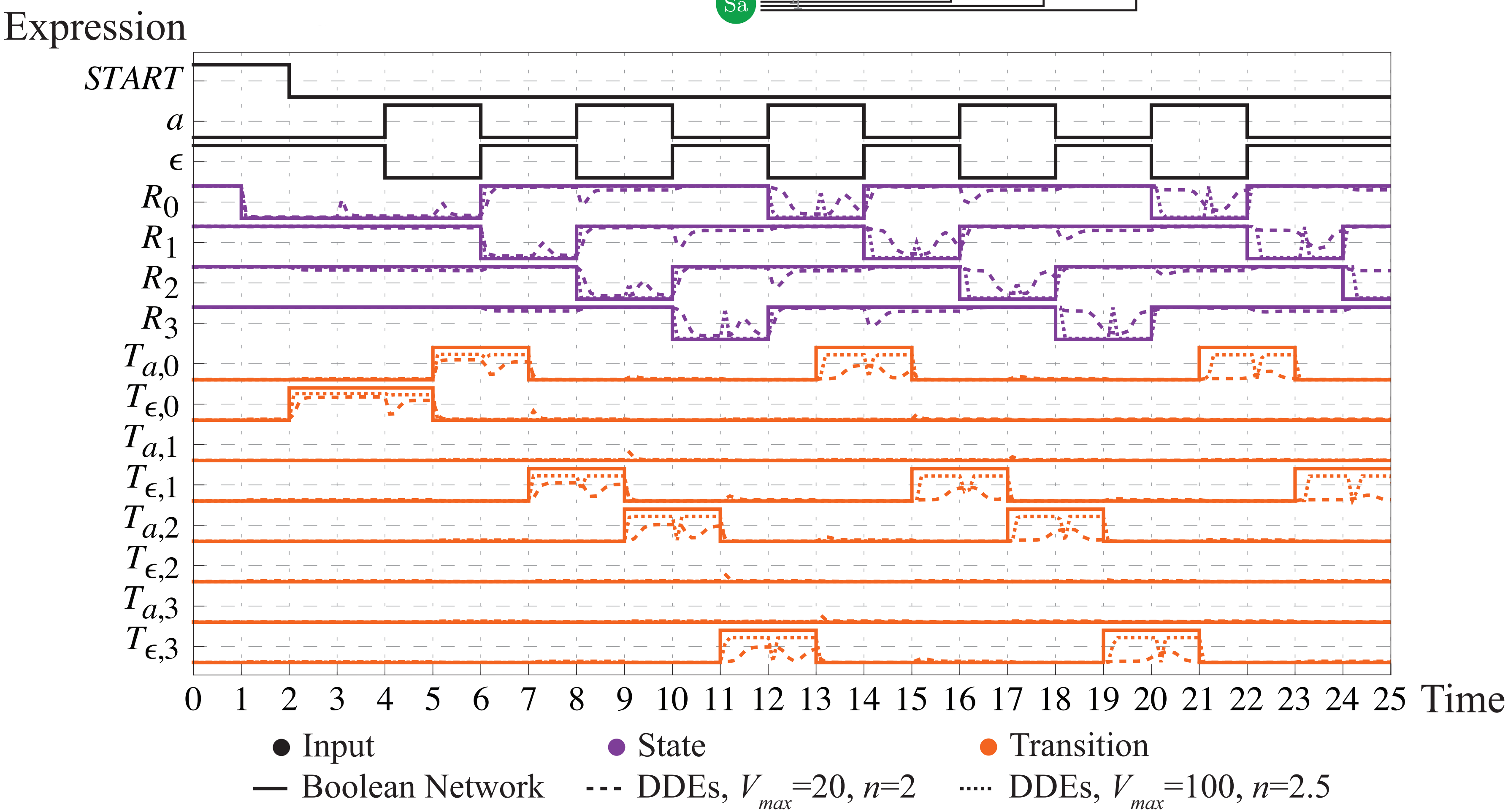
ε-signal. Inducer that is present in the absence of any other inducers.

Input	Final State
$aaaa\ldots aaaaaaaaaaaa\ldots a\ldots$	$q=2$
$a\ldots$	$q=2$
$a\ldots aaaaaaaa\ldots$	$q=0$

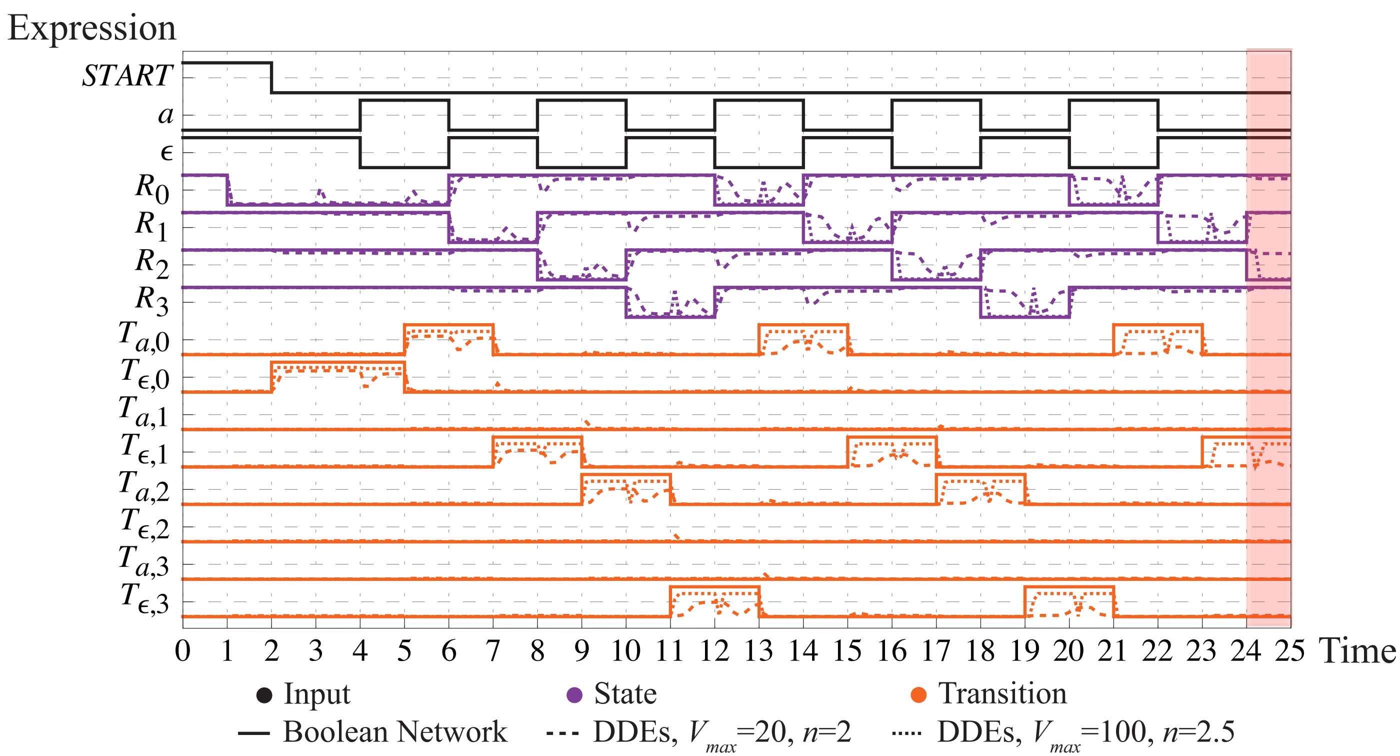
Example II: Modulo-Two Pulse Counter







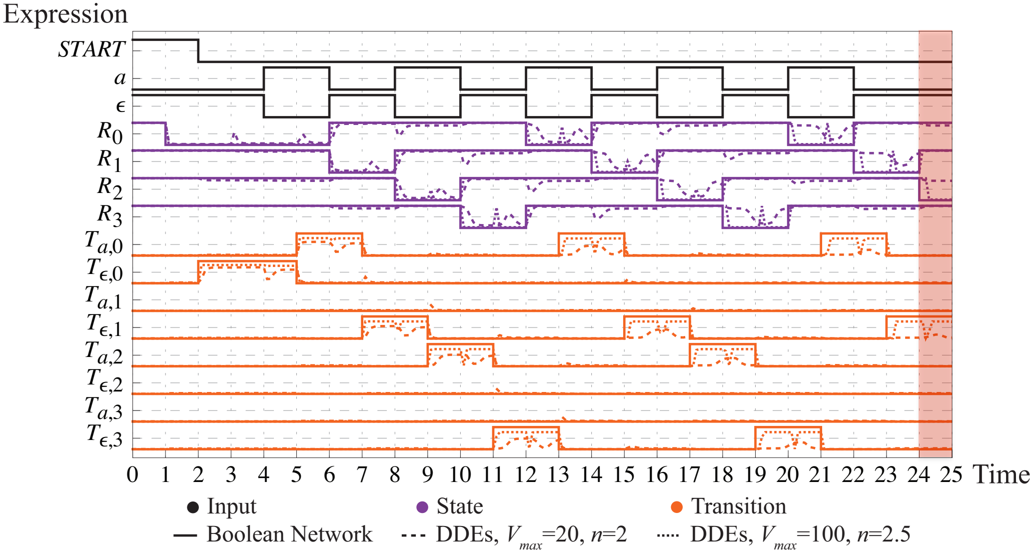
Counter
Comparing the DDE and BN



$$e_{avg} = max_{q \in Q} \frac{2}{\Delta t} \int_{12\Delta t}^{12.5\Delta t} |R_q(t) - \hat{R}_q(t)| dt$$

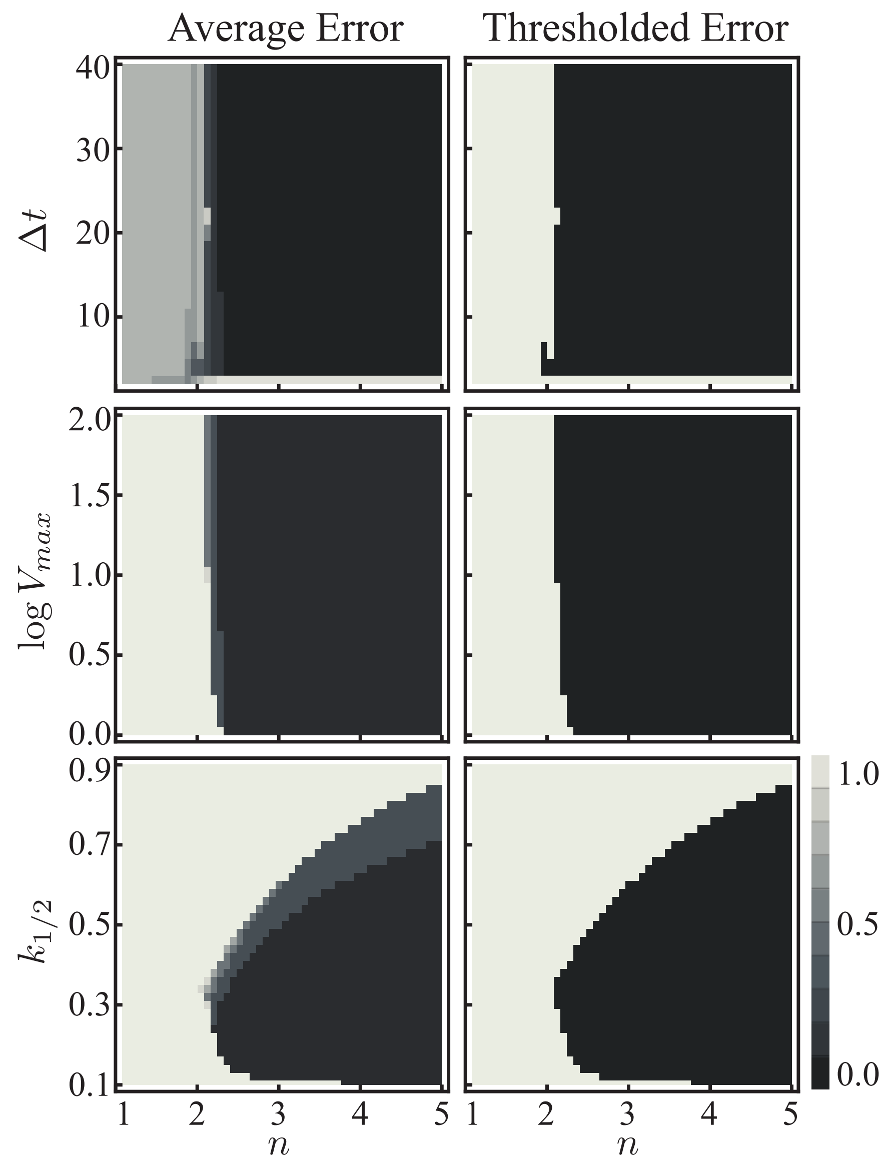
$$e_{thresh} = \begin{cases} 0, & e_{avg} < 1/2, \\ 1, & e_{avg} \geq 1/2 \end{cases} .$$

Counter
Comparing the DDE and BN

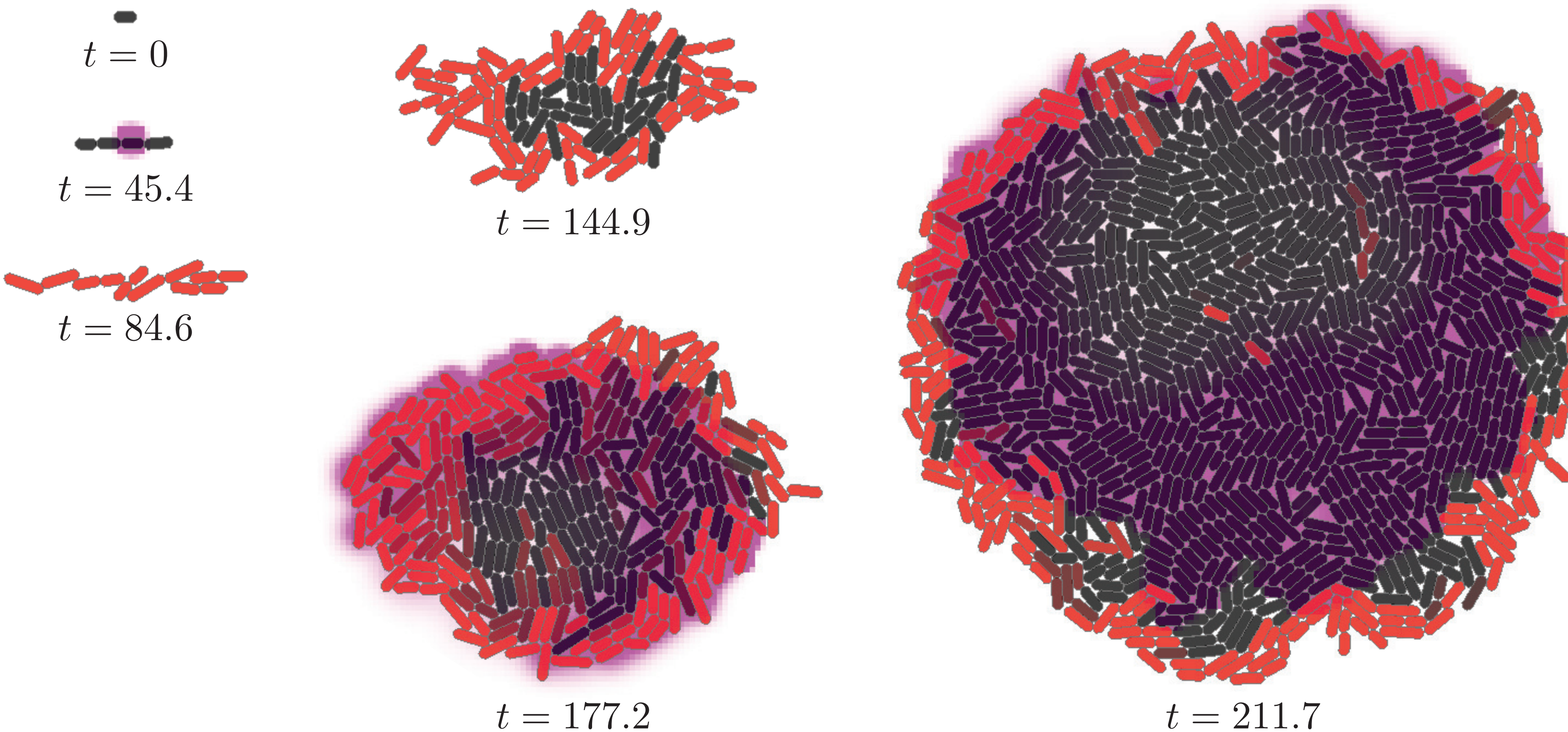


$$e_{avg} = max_{q \in Q} \frac{2}{\Delta t} \int_{12\Delta t}^{12.5\Delta t} |R_q(t) - \hat{R}_q(t)| dt$$

$$e_{thresh} = \begin{cases} 0, & e_{avg} < 1/2, \\ 1, & e_{avg} \geq 1/2 \end{cases}.$$



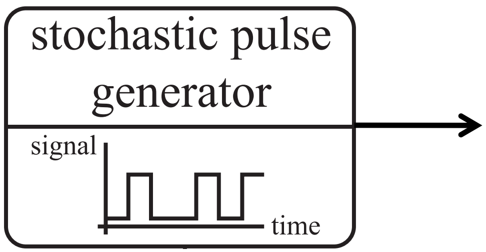
FSMs for Cellular Information Processing



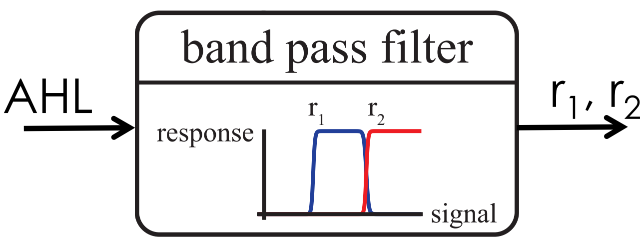
Example: Microcolony Edge Detection FSM Specification

Objective:

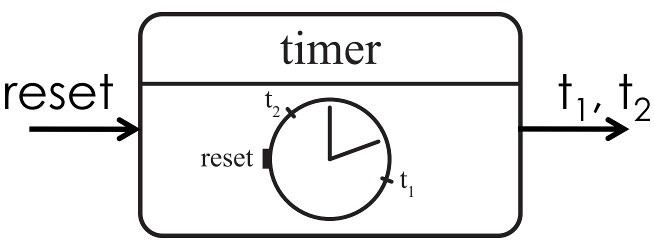
Design a genetic circuit to detect the edge of a growing microcolony. Assume cells have the following sensing/communication capabilities:



Turns on expression of a gene stochastically.



Turns on expression of two different genes according to the concentration of a diffusible molecule.



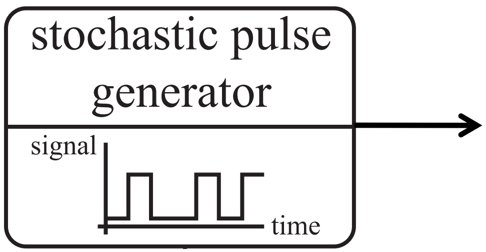
Turns on expression of two different genes at times t_1 and t_2 after reset.

Can be reset by expressing a “reset” gene.

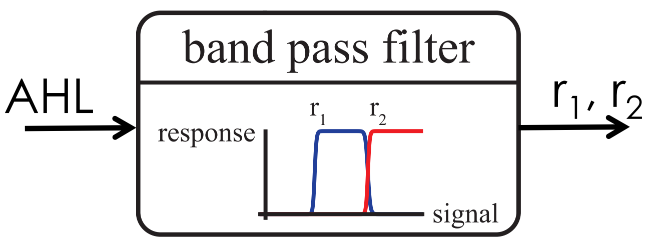
Example: Microcolony Edge Detection FSM Specification

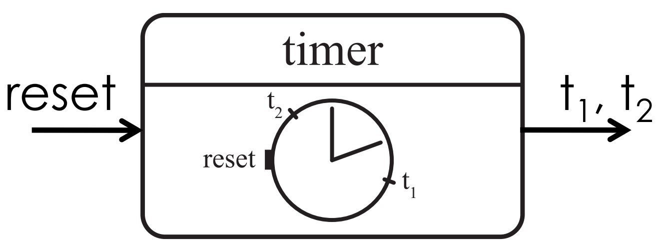
Objective:

Design a genetic circuit to detect the edge of a growing microcolony. Assume cells have the following sensing/communication capabilities:

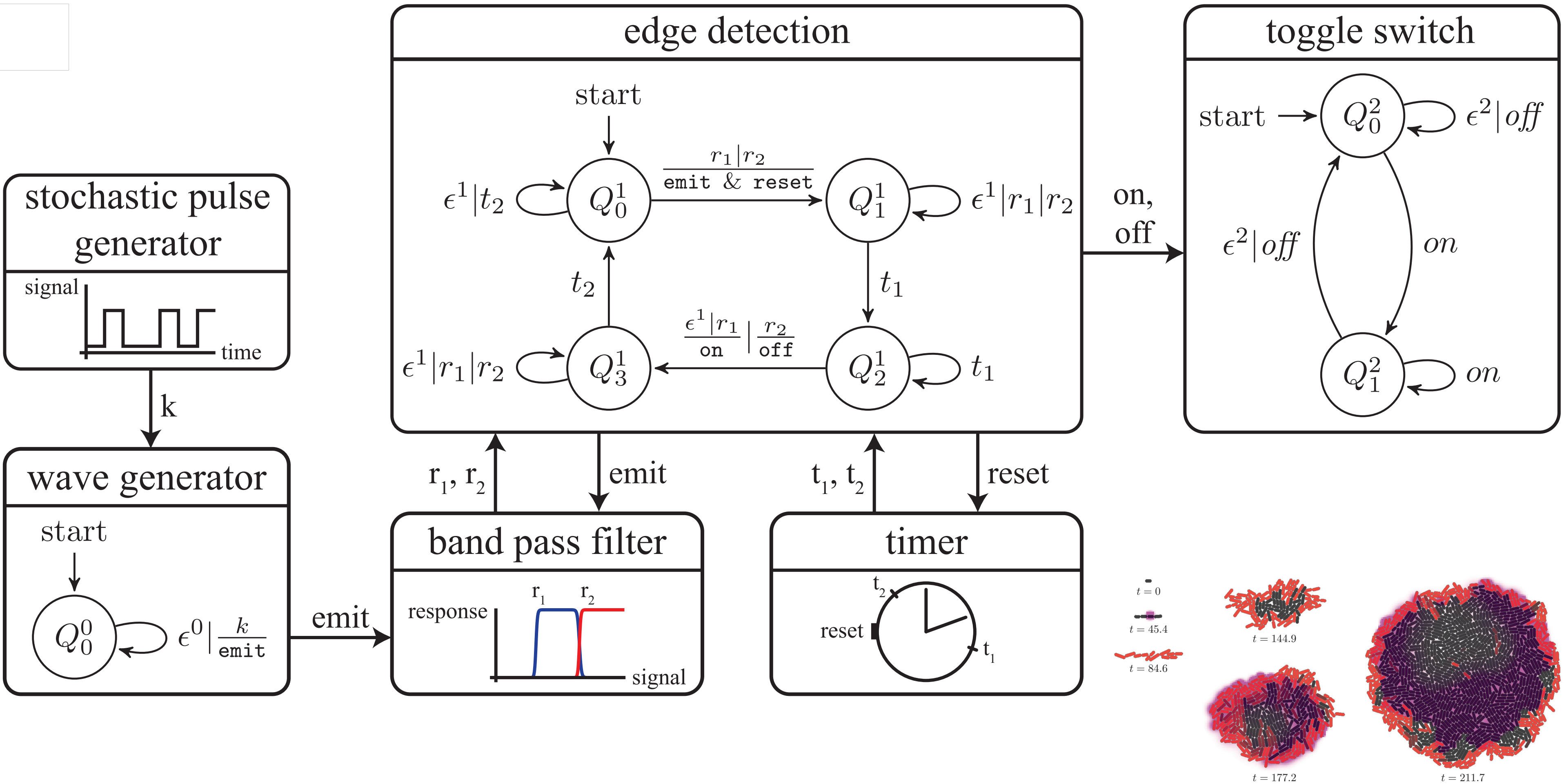


Idea: Determine “edgeness” based on a myopic stadium wave.

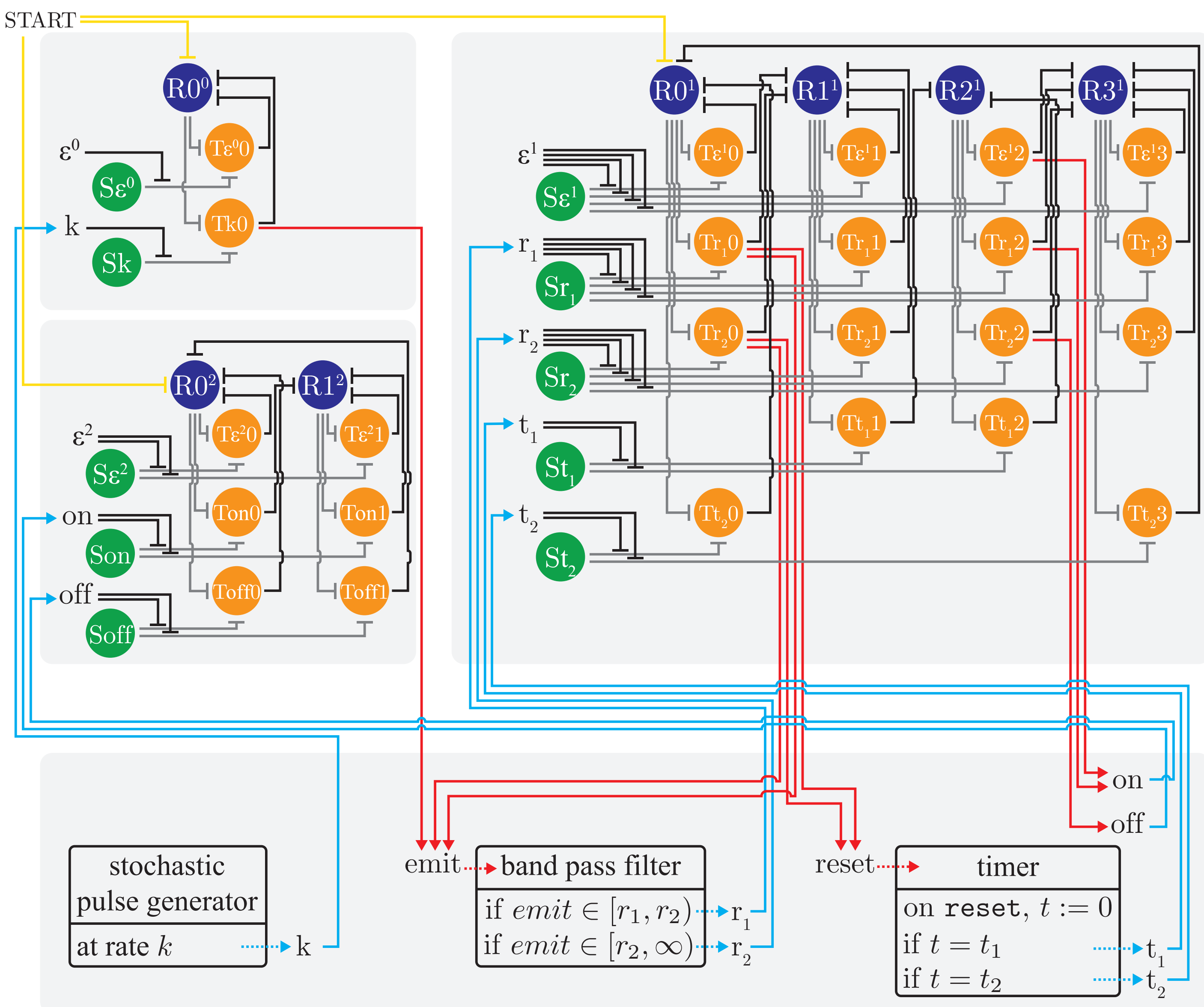




Example: Microcolony Edge Detection FSM Specification



Example: Microcolony Edge Detection FSM Specification



Example: Microcolony Edge Detection FSM Specification