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

Turing Machine

Nondeterministic Pushdown Automata

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

- $(R, G)$
- $(R, Y)$
- $(Y, R)$
- $(G, R)$
Finite State Machines

Example: Traffic Light Controller

States: $L_R \times L_H$

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

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

Diagram:

- Start state: $(R, G)$
- $(R, Y)$
- $(Y, R)$
- $(G, R)$
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: $\{S_1, S_2, \overline{S}_1, \overline{S}_2\}$
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?
Example: Yeast-based Ultrasensitive Detector

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

\[ M = (Q, \Sigma, \delta, q_0, F') \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q )</td>
<td>set of states</td>
</tr>
<tr>
<td>( \Sigma )</td>
<td>set of input symbols</td>
</tr>
<tr>
<td>( \delta : Q \times \Sigma \rightarrow Q )</td>
<td>state transition function</td>
</tr>
<tr>
<td>( q_0 \in Q )</td>
<td>initial state</td>
</tr>
<tr>
<td>( F \subseteq Q )</td>
<td>set of accepting states</td>
</tr>
</tbody>
</table>

Semantics

Input: \( w = \sigma_1\sigma_2...\sigma_n \in \Sigma^* \)
Output: Accept or Not Accept

The machine begins in state \( q_0 \).

At each step \( i \) an input symbol \( \sigma_i \) 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**

\[ M = (Q, \Sigma, \delta, q_0, F') \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q)</td>
<td>set of states</td>
</tr>
<tr>
<td>(\Sigma)</td>
<td>set of input symbols</td>
</tr>
<tr>
<td>(\delta: Q \times \Sigma \rightarrow Q)</td>
<td>state transition function</td>
</tr>
<tr>
<td>(q_0 \in Q)</td>
<td>initial state</td>
</tr>
<tr>
<td>(F \subseteq Q)</td>
<td>set of accepting states</td>
</tr>
</tbody>
</table>

**Semantics**

Input: \( w = \sigma_1\sigma_2...\sigma_n \in \Sigma^* \)
Output: Accept or Not Accept

The machine begins in state \(q_0\).

At each step \(i\) an input symbol \(\sigma_i\) 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

\[ M = (Q, \Sigma, \delta, q_0, F') \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q )</td>
<td>set of states</td>
</tr>
<tr>
<td>( \Sigma )</td>
<td>set of input symbols</td>
</tr>
<tr>
<td>( \delta : Q \times \Sigma \to Q )</td>
<td>state transition function</td>
</tr>
<tr>
<td>( q_0 \in Q )</td>
<td>initial state</td>
</tr>
<tr>
<td>( F \subseteq Q )</td>
<td>set of accepting states</td>
</tr>
</tbody>
</table>

Semantics

Input: \( w = \sigma_1\sigma_2...\sigma_n \in \Sigma^* \)
Output: Accept or Not Accept

What strings does this machine accept?

\( A \)
\( BBBB \)
\( BABABA \)
\( AAAABBBBBBA \)
Finite State Machines

**Syntax**

\[ M = (Q, \Sigma, \delta, q_0, F') \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>set of states</td>
</tr>
<tr>
<td>( \Sigma )</td>
<td>set of input symbols</td>
</tr>
<tr>
<td>( \delta : Q \times \Sigma \rightarrow Q )</td>
<td>state transition function</td>
</tr>
<tr>
<td>( q_0 \in Q )</td>
<td>initial state</td>
</tr>
<tr>
<td>( F \subseteq Q )</td>
<td>set of accepting states</td>
</tr>
</tbody>
</table>

**Semantics**

Input: \( w = \sigma_1\sigma_2...\sigma_n \in \Sigma^* \)

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. 

Trans. regulated

Constitutive

DNA binding sequences

Core promoter
A. Constituitive Core promoter
   Trans. regulated DNA binding sequences

B. Transcriptional repression domain
   I. Small molecule recognition site/degron
   II. Programmable DNA binding domain
   III. Programmable DNA binding domain

Fluorescent marker
Biomolecular Parts

A. Trans. regulated DNA binding sequences
Constituitive Core promoter

B. I. Transcriptional repression domain
II. Small molecule recognition site/degron
III. Programmable DNA binding domain

Sensed molecule
Ø
Fluorescent marker
<table>
<thead>
<tr>
<th>Component Type</th>
<th>Biomolecular Realization</th>
<th>Gene Regulatory Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td><img src="" alt="Diagram" /></td>
<td>Y</td>
</tr>
<tr>
<td>Component Type</td>
<td>Biomolecular Realization</td>
<td>Gene Regulatory Network</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td>![Diagram](U to Y)</td>
<td>Y</td>
</tr>
<tr>
<td>Singly Regulated Gene</td>
<td>![Diagram](U to Y)</td>
<td>U → Y</td>
</tr>
<tr>
<td>Component Type</td>
<td>Biomolecular Realization</td>
<td>Gene Regulatory Network</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td>Singly Regulated Gene</td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td>Doubly Regulated Gene</td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td>Component Type</td>
<td>Biomolecular Realization</td>
<td>Gene Regulatory Network</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td><img src="transcriptional_unregulated" alt="Diagram" /></td>
<td><img src="transcriptional_unregulated_gene" alt="Diagram" /></td>
</tr>
<tr>
<td>Singly Regulated Gene</td>
<td><img src="singly_regulated" alt="Diagram" /></td>
<td><img src="singly_regulated_gene" alt="Diagram" /></td>
</tr>
<tr>
<td>Doubly Regulated Gene</td>
<td><img src="doubly_regulated" alt="Diagram" /></td>
<td><img src="doubly_regulated_gene" alt="Diagram" /></td>
</tr>
<tr>
<td>Small Molecule Sensor</td>
<td><img src="small_molecule_sensor" alt="Diagram" /></td>
<td><img src="small_molecule_sensor_gene" alt="Diagram" /></td>
</tr>
</tbody>
</table>

**Biomolecular Parts**
Gene Regulatory Networks

Syntax

\[ G = (V, U, E_r, H_r) \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V )</td>
<td>Set of <em>gene products</em></td>
</tr>
<tr>
<td>( U )</td>
<td>Set of <em>inducers</em></td>
</tr>
<tr>
<td>( E_r \subset V \times V )</td>
<td>repression relation</td>
</tr>
<tr>
<td>( H_r \subset U \times E_r )</td>
<td>inducible repression relation</td>
</tr>
</tbody>
</table>
Gene Regulatory Networks

**Syntax**

\[ G = (V, U, E_r, H_r) \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V )</td>
<td>Set of gene products</td>
</tr>
<tr>
<td>( U )</td>
<td>Set of inducers</td>
</tr>
<tr>
<td>( E_r \subset V \times V )</td>
<td>repression relation</td>
</tr>
<tr>
<td>( H_r \subset U \times E_r )</td>
<td>inducible repression relation</td>
</tr>
</tbody>
</table>

**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.,

\[ Y^t = \begin{bmatrix} R_0^t \\ R_1^t \\ S_a^t \\ S_b^t \\ T_{a,0}^t \\ T_{b,0}^t \\ T_{a,1}^t \\ T_{b,1}^t \end{bmatrix} \quad U^t = \begin{bmatrix} a^t \\ b^t \end{bmatrix} \]
## Boolean Network Dynamics

<table>
<thead>
<tr>
<th>Component Type</th>
<th>Gene Regulatory Network</th>
<th>Boolean Network Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td>( Y )</td>
<td>( Y^t = \text{on} )</td>
</tr>
<tr>
<td>Singly Regulated Gene</td>
<td>( U \longrightarrow Y )</td>
<td>( Y^{t+1} = \neg U^t )</td>
</tr>
<tr>
<td>Doubly Regulated Gene</td>
<td>( U_2 \downarrow U_1 \longrightarrow Y )</td>
<td>( Y^{t+1} = \neg(U_1^t \lor U_2^t) )</td>
</tr>
<tr>
<td>Small Molecule Sensor</td>
<td>( a \downarrow S_a \longrightarrow Y )</td>
<td>( S_a^t = \neg a^t )</td>
</tr>
</tbody>
</table>
### Boolean Network Dynamics

<table>
<thead>
<tr>
<th>Component Type</th>
<th>Gene Regulatory Network</th>
<th>Boolean Network Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td>( Y )</td>
<td>( Y^t = on )</td>
</tr>
<tr>
<td>Singly Regulated Gene</td>
<td>( U \rightarrow Y )</td>
<td>( Y^{t+1} = \neg U^t )</td>
</tr>
<tr>
<td>Doubly Regulated Gene</td>
<td>( U_2 \rightarrow \downarrow Y ) ( U_1 \rightarrow \downarrow Y )</td>
<td>( Y^{t+1} = \neg (U_1^t \lor U_2^t) )</td>
</tr>
<tr>
<td>Small Molecule Sensor</td>
<td>( S_a \rightarrow \downarrow Y )</td>
<td>( S_a^t = \neg a^t ) ( Y^{t+1} = \neg S_a^t )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( U^t )</th>
<th>( Y^{t+1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
### Boolean Network Dynamics

<table>
<thead>
<tr>
<th>Component Type</th>
<th>Gene Regulatory Network</th>
<th>Boolean Network Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td>Y</td>
<td>$Y^t = \text{on}$</td>
</tr>
<tr>
<td>Singly Regulated Gene</td>
<td>U → Y</td>
<td>$Y^{t+1} = \neg U^t$</td>
</tr>
<tr>
<td>Doubly Regulated Gene</td>
<td>U_2 \downarrow \ U_1 \rightarrow Y</td>
<td>$Y^{t+1} = \neg (U_1^t \lor U_2^t)$</td>
</tr>
<tr>
<td>Small Molecule Sensor</td>
<td>a \downarrow S_a \rightarrow Y</td>
<td>$S_a^t = \neg a^t$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$U^t$</th>
<th>$Y^{t+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
## Boolean Network Dynamics

<table>
<thead>
<tr>
<th>Component Type</th>
<th>Gene Regulatory Network</th>
<th>Boolean Network Equations</th>
<th>What is the truth table for the Double Regulated Gene and Small Molecular Sensor?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td>Y</td>
<td>$Y^t = \text{on}$</td>
<td></td>
</tr>
<tr>
<td>Singly Regulated Gene</td>
<td>U → Y</td>
<td>$Y^{t+1} = \neg U^t$</td>
<td></td>
</tr>
<tr>
<td>Doubly Regulated Gene</td>
<td>$U_1 \downarrow Y$</td>
<td>$Y^{t+1} = \neg (U_1^t \vee U_2^t)$</td>
<td></td>
</tr>
<tr>
<td>Small Molecule Sensor</td>
<td>$S_a \downarrow Y$</td>
<td>$S_a^t = \neg a^t$</td>
<td>$a^t$</td>
</tr>
</tbody>
</table>
### Boolean Network Dynamics

<table>
<thead>
<tr>
<th>Component Type</th>
<th>Gene Regulatory Network</th>
<th>Boolean Network Equations</th>
<th>What is the truth table for the Double Regulated Gene and Small Molecular Sensor?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td>Y</td>
<td>$Y^t = \text{on}$</td>
<td>-</td>
</tr>
</tbody>
</table>
| Singly Regulated Gene | $U \rightarrow Y$ | $Y^{t+1} = \neg U^t$ | $\begin{array}{c|c}
U^t & Y^{t+1} \\
0 & 1 \\
1 & 0 \\
\end{array}$ |
| Doubly Regulated Gene | $U_1 \rightarrow Y$ | $Y^{t+1} = \neg (U_1^t \lor U_2^t)$ | $\begin{array}{c|c|c}
U_1^t & U_2^t & Y^{t+1} \\
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}$ |
| Small Molecule Sensor | $S_a \rightarrow Y$ | $S_a^t = \neg a^t$ | $\begin{array}{c|c}
a^t & Y^{t+1} \\
0 & 0 \\
1 & 1 \\
\end{array}$ |
Objective:
Given a FSM $M$ as a specification, construct a GRN $g(M)$ that encodes the behavior the FSM.

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q$</td>
<td>set of states</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>set of input symbols</td>
</tr>
<tr>
<td>$\delta : Q \times \Sigma \rightarrow Q$</td>
<td>state transition function</td>
</tr>
<tr>
<td>$q_0 \in Q$</td>
<td>initial state</td>
</tr>
<tr>
<td>$F \subseteq Q$</td>
<td>set of accepting states</td>
</tr>
</tbody>
</table>
For each $q$ in $Q$, let $R_q$ be a singly regulated state gene, and “wire” START to $R_{q_0}$.

Each state $q$ in $Q$, is represented by a gene expression profile where $R_q$ is at a low level of expression, and all other state genes are at a high level of expression.
GRN General Construction Method

Figure 1: Simple two-state machine described as (A) a directed graph representation of a finite state machine, (B) a gene regulatory network made of repressing transcription factors and inducers, and (C) a biomolecular realization of the same GRN using the parts described in Figure 2. In the GRN representation orange circles denote transition species, purple circles denote state species, and green circles denote sensor species. In the GRN and biomolecular realization, the gene network is in state \( i \) when species \( R_i \) is at a low level expression, and following transition \( (q, A) \) when transition species \( T_q \) is at a high level of expression.

For each \( q \) in \( Q \), let \( R_q \) be a singly regulated state gene, and “wire” START to \( R_{q_0} \).

What are the expression levels of \( R_0 \) and \( R_1 \) with START on? START off?
For each $q$ in $Q$, let $R_q$ be a singly regulated state gene, and “wire” START to $R_{q_0}$.

<table>
<thead>
<tr>
<th>$START^t$</th>
<th>$R^t_{0+1}$</th>
<th>$R^t_{1+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
GRN General Construction Method

Specification:

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

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

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

For each $\sigma$ in $\Sigma$, let $S\sigma$ be a transcriptionally unregulated sensor gene for inducer $\sigma$.
For each \((q, \sigma, q')\) in \(Q \times \Sigma \times Q\), such that \(\delta(q, \sigma) q'\), let \(T \sigma q\) be a doubly regulated transition gene.
For each \((q, \sigma, q')\) in \(Q \times \Sigma \times Q\), such that \(\delta (q, \sigma) q'\), wire \(R_q\) to \(T \sigma q\).
What is the Boolean function that describes the expression level of $T_{a,0}$ at time $t+1$?

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

Figure 1: Simple two-state machine described as (A) a directed graph representation of a finite state machine, (B) a gene regulatory network made of repressing transcription factors and inducers, and (C) a biomolecular realization of the same GRN using the parts described in Figure 2. In the GRN representation orange circles denote transition species, purple circles denote state species, and green circles denote sensor species. In the GRN and biomolecular realization, the gene network is in state $i$ when species $R_i$ is at a low level expression, and following transition $(q, R)$ when transition species $T_q$ is at a high level of expression.
What is the Boolean function that describes the expression level of Ta0 at time $t+1$?

\[
T_{a,0}^{t+1} = \neg R_0^t = \text{START}^{t-1}
\]
Figure 1: Simple two-state machine described as (A) a directed graph representation of a finite state machine, (B) a gene regulatory network made of repressing transcription factors and inducers, and (C) a biomolecular realization of the same GRN using the parts described in Figure 2. In the GRN representation, orange circles denote transition species, purple circles denote state species, and green circles denote sensor species. In the GRN and biomolecular realization, the gene network is in state $i$ when species $R_i$ is at a low level of expression, and following transition $(q, \sigma, q')$ when transition species $T_\sigma q$ is at a high level of expression.

For each $(q, \sigma, q')$ in $Q \times \Sigma \times Q$, such that $\delta(q, \sigma) q'$, wire $S \sigma$ to $T_\sigma q$. 

**GRN General Construction Method**

**Specification:**

For each $(q, \sigma, q')$ in $Q \times \Sigma \times Q$, such that $\delta(q, \sigma) q'$, wire $S \sigma$ to $T_\sigma q$. 

**START**

- **R0**
  - **Ta0**
  - **Sa**
  - **Tb0**
  - **Sb**

- **R1**
  - **Ta1**
  - **Tb1**
GRN General Construction Method

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

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

Specification:

start

A
B

Figure 1: Simple two-state machine described as (A) a directed graph representation of a finite state machine, (B) a gene regulatory network made of repressing transcription factors and inducers, and (C) a biomolecular realization of the same GRN using the parts described in Figure 2. In the GRN representation orange circles denote transition species, purple circles denote state species, and green circles denote sensor species. In the GRN and biomolecular realization, the gene network is in state $i$ when species $R_i$ is at a low level expression, and following transition $(q, a)$ when transition species $T_q$ is at a high level of expression.
What is the Boolean function that describes the expression level of each transition gene at time $t+1$?

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

$$T_{b,0}^{t+1} = \neg R_0^t \land b^t$$

$$T_{a,1}^{t+1} = \neg R_1^t \land a^t$$

$$T_{b,1}^{t+1} = \neg R_1^t \land b^t$$
For each \((q, \sigma, q')\) in \(Q \times \Sigma \times Q\), such that \(\delta(q, \sigma) = q'\), wire \(T \sigma q\) to \(Rq'\).
For each \((q, \sigma, q')\) in \(Q \times \Sigma \times Q\), such that \(\delta(q, \sigma) = q'\), wire \(T \sigma q\) to \(R q'\).
For each \((q, \sigma, q')\) in \(Q \times \Sigma \times Q\), such that \(\delta(q, \sigma) = q'\), wire \(T \sigma q\) to \(Rq'\).
For each \((q, \sigma, q')\) in \(Q \times \Sigma \times Q\), such that \(\delta(q, \sigma) q'\), wire \(T \sigma q\) to \(Rq'\).
For each \((q, \sigma, q')\) in \(Q \times \Sigma \times Q\), such that \(\delta(q, \sigma) q'\), wire \(T \sigma q\) to \(Rq'\).
GRN General Construction Method

Specification:

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

DONE! Almost...
An input sequence $w$ in $\Sigma^*$ is represented by a trajectory over START and the inducers, e.g.

$\begin{array}{c|c|c}
\text{START} & \text{AA...A} & \text{BB...B} \\
\hline
A & \text{AA...A} & \text{BB...B} \\
B & \text{AA...A} & \text{BB...B}
\end{array}$
Let $h_{BN}(w,t)$ be the input trajectory to $g(M)$ where $w = \sigma_{c1} \sigma_{c2} \ldots \sigma_{cm}$.

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

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

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

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 $g(M)$

\begin{itemize}
\item $S_a$
\item $S_b$
\item START
\item $R_0$
\item $R_1$
\item $T_{a0}$
\item $T_{a1}$
\item $T_{b0}$
\item $T_{b1}$
\item $A$
\item $B$
\item $Q_0$
\item $Q_1$
\end{itemize}

\begin{itemize}
\item $A$
\item $B$
\item $R_0$
\item $R_1$
\item $T_{ij}$
\item $TA_0$
\end{itemize}

\begin{itemize}
\item $A$
\item $B$
\end{itemize}
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 $g(M)$

```
A           A
Q0           Q1
B        B
R0        R1
TA0        TA1
TB1        TB0
```

```
START
A
B
R0
R1
Tij
```

```
sa

b
```

```
0
A
B
1
```

```
start
```
Theorem. Given a finite state machine $M$, the GRN $g(M)$ simulates $M$ when modeled as a Boolean network.
**Theorem.** Assuming a Boolean network model, GRNs are computationally equivalent to FSMs.
Representations of the Two-State Machine

A.

\[ Q = \{0, 1\} \quad \delta(0, a) \mapsto 1 \]
\[ \Sigma = \{a, b\} \quad \delta(0, b) \mapsto 0 \]
\[ F = \{1\} \quad \delta(1, a) \mapsto 1 \]
\[ q_0 = 0 \quad \delta(1, b) \mapsto 0 \]
## Delay Differential Equations Model

<table>
<thead>
<tr>
<th>Component Type</th>
<th>Biomolecular Realization</th>
<th>GRN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td><img src="image1" alt="Diagram" /></td>
<td>Y</td>
</tr>
<tr>
<td>Singly Regulated Gene</td>
<td><img src="image2" alt="Diagram" /></td>
<td>U → Y</td>
</tr>
<tr>
<td>Doubly Regulated Gene</td>
<td><img src="image3" alt="Diagram" /></td>
<td>U_1 → Y, U_2 → Y</td>
</tr>
<tr>
<td>Small Molecule Sensor</td>
<td><img src="image4" alt="Diagram" /></td>
<td>S_a → Y</td>
</tr>
</tbody>
</table>

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**

\[ G = (V, E_r, E_a) \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V )</td>
<td>set of gene products or inducers</td>
</tr>
<tr>
<td>( E_r \subset V \times (V \cup E_r \cup E_a) )</td>
<td>repression relation</td>
</tr>
<tr>
<td>( E_a \subset V \times (V \cup E_r \cup E_a) )</td>
<td>activation relation</td>
</tr>
</tbody>
</table>

**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.,

\[
Y^t = \begin{bmatrix}
R_0(t) \\
R_1(t) \\
S_a(t) \\
S_b(t) \\
T_{a,0}(t) \\
T_{b,0}(t) \\
T_{a,1}(t) \\
T_{b,1}(t)
\end{bmatrix},
\quad U(t) = \begin{bmatrix}
a(t) \\
b(t)
\end{bmatrix}
\]
\[ \begin{align*}
V_{\text{max}}, \beta & \quad \text{protein production and degradation rates} \\
k_p & \quad \text{small molecule binding affinity} \\
k_{1/2} & \quad \text{input for half-maximum gene production} \\
n & \quad \text{Hill coefficient} \\
\tau & \quad \text{time delay, approximates transcription/translation dynamics} \quad \tau = 1
\end{align*} \]

\[ \begin{align*}
\frac{d}{dt} S_a(t) & = \quad V_{\text{max}} - (\beta + k_p a(t)) S_a(t) \\
\frac{d}{dt} Y(t) & = \quad V_{\text{max}} \left( \frac{1}{1 + \left( \frac{S_a(t-\tau)}{k_{1/2}} \right)^n} \right) - \beta Y(t)
\end{align*} \]
## Delay Differential Equations Model

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

\[ V_{max} = \beta \quad k_p \gg \beta \]

### Equations:
\[
\frac{d}{dt} S_a(t) = V_{max} - (\beta + k_p a(t)) S_a(t)
\]

- **Production**
- **Degradation**

Can you interpret this as a chemical reaction network?
Delay Differential Equations Model

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

\[ V_{max} = \beta, \quad k_p \gg \beta \]

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

Production  Degradation

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{max}$, $\beta$</td>
<td>protein production and degradation rates</td>
</tr>
<tr>
<td>$k_p$</td>
<td>small molecule binding affinity</td>
</tr>
<tr>
<td>$k_{1/2}$</td>
<td>input for half-maximum gene production</td>
</tr>
<tr>
<td>$n$</td>
<td>Hill coefficient</td>
</tr>
<tr>
<td>$\tau$</td>
<td>time delay, approximates transcription/translation dynamics</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
S_a(t) &= V_{max} - (\beta + k_p a(t)) S_a(t) \\
0 &= V_{max} - (\beta + k_p a^*) S_a^* \\
\emptyset &\xrightarrow{V_{max}} S_a \quad \beta + k_p a(t) \quad \emptyset
\end{align*}
\]
**Delay Differential Equations Model**

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

**Equations**

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

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

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

\[
\text{Ø} \xrightarrow{V_{max}} S_a \xrightarrow{\beta + k_p a(t)} \text{Ø}
\]
### Delay Differential Equations Model

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

Original set of equations:

\[
\begin{align*}
\frac{d}{dt}S_a(t) &= V_{max} - (\beta + k_p a(t))S_a(t) \\
\frac{d}{dt}Y(t) &= \frac{V_{max}}{1 + \left(\frac{S_a(t-\tau)}{k_{1/2}}\right)^n} - \beta Y(t)
\end{align*}
\]
$V_{\text{max}}, \beta$  protein production and degradation rates
$k_p$  small molecule binding affinity
$k_{1/2}$  input for half-maximum gene production
$n$  Hill coefficient
$\tau$  time delay, approximates transcription/translation dynamics  \( \tau = 1 \)

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

Production

Degradation

What happens when

- \( S_a^* = 0 \) ?
- \( S_a^* \to \infty \) ?
- \( S_a^* = k_{1/2} \) ?
Delay Differential Equations Model

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

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

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

\(V_{\text{max}} = \beta\), \(k_p \gg \beta\)
### Delay Differential Equations Model

**Parameters and Equations**

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

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

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

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

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

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

Y^* = \frac{V_{\text{max}}}{\beta} = Y_{\text{max}}
Delay Differential Equations Model

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

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

\[
S^*_a \to \infty \quad 0 = \frac{V_{\text{max}}}{1 + \left( \frac{S^*_a}{k_{1/2}} \right)^n} - \beta Y^*
\]
### Delay Differential Equations Model

<table>
<thead>
<tr>
<th>Component/Type</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biomolecular Realization</td>
<td>$\frac{d}{dt} Y(t) = \frac{V_{max}}{1 + \left( \frac{S_a(t-\tau)}{k_{1/2}} \right)^n} - \beta Y(t)$</td>
</tr>
<tr>
<td>Doubly Regulated Gene</td>
<td>$S^* \rightarrow \infty \quad 0 = \frac{V_{max}}{1 + \left( \frac{S^<em><em>a}{k</em>{1/2}} \right)^n} - \beta Y^</em>$</td>
</tr>
<tr>
<td>Singly Regulated Gene</td>
<td>$S_a \rightarrow \infty \quad 0 = \frac{V_{max}}{1 + \left( \frac{S_a}{k_{1/2}} \right)^n} - \beta Y^*$</td>
</tr>
</tbody>
</table>

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

\[ V_{max} = \beta \]
\[ k_p \gg \beta \]

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

\[ S^* \rightarrow \infty \quad 0 = \frac{V_{max}}{1 + \left( \frac{S^*_a}{k_{1/2}} \right)^n} - \beta Y^* \]

\[ = 0 - \beta Y^* \]

\[ Y^* = 0 \]
Delay Differential Equations Model

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

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

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

\( V_{\text{max}} = \beta \)
\( k_p \gg \beta \)

\( V_{\text{max}} \), \( \beta \), \( k_p \), \( k_{1/2} \), \( n \), \( \tau \) are parameters in the model.
Delay Differential Equations Model

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

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

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

\[ Y^* = \frac{1}{2} \cdot \frac{V_{max}}{\beta} = \frac{1}{2} Y_{max} \]
### Delay Differential Equations Model

**$V_{max}, \beta$**  
protein production and degradation rates

**$k_p$**  
small molecule binding affinity

**$k_{1/2}$**  
input for half-maximum gene production

**$n$**  
Hill coefficient

**$\tau$**  
time delay, approximates transcription/translation dynamics  
\[ \tau = 1 \]

---

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

![Dose Response Graphs](image)

- **$k_{1/2}$**
- **$n$**

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{max}$</td>
<td>$\beta$</td>
</tr>
<tr>
<td>$k_p$</td>
<td>small molecule binding affinity</td>
</tr>
<tr>
<td>$k_{1/2}$</td>
<td>input for half-maximum gene production</td>
</tr>
<tr>
<td>$n$</td>
<td>Hill coefficient</td>
</tr>
<tr>
<td>$\tau$</td>
<td>time delay, approximates transcription/translation dynamics</td>
</tr>
</tbody>
</table>
## Delay Differential Equations Model

<table>
<thead>
<tr>
<th>Component Type</th>
<th>Biomolecular Realization</th>
<th>GRN</th>
<th>Delay Differential Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcriptionally Unregulated Gene</td>
<td><img src="image1" alt="Diagram" /></td>
<td>Y</td>
<td>( \frac{d}{dt} Y(t) = V_{max} - \beta Y(t) )</td>
</tr>
<tr>
<td>Singly Regulated Gene</td>
<td><img src="image2" alt="Diagram" /></td>
<td>U → Y</td>
<td>( \frac{d}{dt} Y(t) = \frac{V_{max}}{1 + \left( \frac{U(t-\tau)}{k_{1/2}} \right)^n} - \beta Y(t) )</td>
</tr>
<tr>
<td>Doubly Regulated Gene</td>
<td><img src="image3" alt="Diagram" /></td>
<td>U_1 → Y</td>
<td>( \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) )</td>
</tr>
<tr>
<td>Small Molecule Sensor</td>
<td><img src="image4" alt="Diagram" /></td>
<td>S_a → Y</td>
<td>( \frac{d}{dt} S_a(t) = V_{max} - (\beta + k_p a(t)) S_a(t) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \frac{d}{dt} Y(t) = \frac{V_{max}}{1 + \left( \frac{S_a(t-\tau)}{k_{1/2}} \right)^n} - \beta Y(t) )</td>
</tr>
</tbody>
</table>
Let $h_{\text{DDE}}(w, \Delta t, t)$ be the input trajectory to DDE model of $g(M)$ with pulse width $\Delta t$, where $w=\sigma_{c_1}\sigma_{c_2}...\sigma_{cm}$.

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

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

$$\sigma_j(t) = \begin{cases} 
1, & \exists c_i \text{ s.t. } j = c_i \text{ and } t \in [2i, 2i+1) \\
0, & \text{otherwise}
\end{cases}$$
\textbf{Example II: Modulo-Two Pulse Counter}

\begin{itemize}
  \item [**ε-symbol**] Applied whenever another input symbol is not supplied.
\end{itemize}

<table>
<thead>
<tr>
<th>Input</th>
<th>Final State</th>
</tr>
</thead>
<tbody>
<tr>
<td>$aaaa . . . aaaaaaaaaaaaaaaaa . . . a . . .$</td>
<td>$q = 2$</td>
</tr>
<tr>
<td>$a . . .$</td>
<td>$q = 2$</td>
</tr>
<tr>
<td>$a . . . aaaaaaaaaa . . .$</td>
<td>$q = 0$</td>
</tr>
</tbody>
</table>
\(\varepsilon\)-symbol. Applied at any time step where another input symbol is not supplied.

\(\varepsilon\)-signal. Inducer that is present in the absence of any other inducers.

<table>
<thead>
<tr>
<th>Input</th>
<th>Final State</th>
</tr>
</thead>
<tbody>
<tr>
<td>(aaaa \ldots a\ldots a\ldots a\ldots a\ldots)</td>
<td>(q = 2)</td>
</tr>
<tr>
<td>(a \ldots)</td>
<td>(q = 2)</td>
</tr>
<tr>
<td>(a \ldots a\ldots a\ldots a\ldots a\ldots)</td>
<td>(q = 0)</td>
</tr>
</tbody>
</table>
Example II: Modulo-Two Pulse Counter

Expression

START

- Input
- State
- Boolean Network
- DDEs, $V_{\text{max}} = 20, n=2$
- DDEs, $V_{\text{max}} = 100, n=2.5$
- Transition

Time
Equation 65 describes the error model respectively, where gene expression at a fixed time after applying a control input in a DDE model to the ideal Boolean network. Transitioning from states 0 to 3 according to prescribed pulses of inducer input, trajectories of the DDE model are compared against the trajectory produced by the ideal Boolean network model for an input of five pulses of equal duration of inducer.

In networks like the modulo two pulse counter, more reliable performance can be achieved by allowing the machine to settle into a periodic orbit before changing inputs.

In general, we would like to know, given a finite state machine (FSM) and a GRN implementation of that FSM, how robust is the GRN implementation to changes in the parameters. In most cases, the gene expression levels decrease over time, and this is particularly evident with transition genes. By increasing the Hill coefficients $\alpha$ and $\beta$, we can approximate a sharper sigmoidal response. This is reflected in Figure 5C, where the behavior of the DDE model for $V_{\max} = 100, n=2$ is shown.

Initially, both DDE simulations track quite well. However, by time $t = 20$, the gene expression at states 0, 1, 2, 3 for $V_{\max} = 100, n=2.5$ begins to deviate from the ideal Boolean network model. This is not surprising, and for various methods the gene expression at states 0, 1, 2, 3 for $V_{\max} = 100, n=2.5$ begins to deviate from the ideal Boolean network model. This is not surprising, and for various methods this can be seen as a significant lag or change in maximum amplitude of expression. This is reflected in Figure 5C, where the gene expression at states 0, 1, 2, 3 for $V_{\max} = 100, n=2.5$ begins to deviate from the ideal Boolean network model.

For a fixed $k$, the control input is defined as $S = 3$, and the machine accepts all sequences of input symbols that end in an even number of contiguous $0$ symbols. In this way, an accepted input sequence consisting of an even number of contiguous $0$ symbols will end in state $R_1$, while an accepted input sequence consisting of an odd number of contiguous $0$ symbols will end in state $R_2$. In the case of $V_{\max} = 100, n=2.5$, the machine accepts all sequences of input symbols that end in an even number of contiguous $0$ symbols. In this way, an accepted input sequence consisting of an even number of contiguous $0$ symbols will end in state $R_1$, while an accepted input sequence consisting of an odd number of contiguous $0$ symbols will end in state $R_2$.

In Figure 4, the DDE simulations qualitatively track the trajectories of the Boolean network model. However, by time $t = 20$, the gene expression at states 0, 1, 2, 3 for $V_{\max} = 100, n=2.5$ begins to deviate from the ideal Boolean network model. This is not surprising, and for various methods this can be seen as a significant lag or change in maximum amplitude of expression. This is reflected in Figure 5C, where the gene expression at states 0, 1, 2, 3 for $V_{\max} = 100, n=2.5$ begins to deviate from the ideal Boolean network model.
Equation 65 describes error model respectively, $a$ to expression to upstream transcription factors. The duration of the input pulse may also be increased relative closely than the model where

\[
\tau_{\text{tra}} \text{ trajectories of the DDE model against the trajectory produced by the ideal Boolean network model for an }
\]

input sequence can be split into a sequence of contiguous modulo two pulse counter is shown in Figure 5A. There are two input symbols to this machine, one method for examining how well the behavior of the DDE model approximates the ideal Boolean network. Comparing the DDE and BN Counter Examples.

Example I: Modulo-Two Pulse Counter

Comparing the DDE and BN

Example II: Modulo-Two Pulse Counter

Comparing the DDE and BN

$\text{Expression}$

\[
\begin{align*}
\text{START} & \\
a & \\
\epsilon & \\
R_0 & \\
R_1 & \\
R_2 & \\
R_3 & \\
T_{a,0} & \\
T_{e,0} & \\
T_{a,1} & \\
T_{e,1} & \\
T_{a,2} & \\
T_{e,2} & \\
T_{e,3} & \\
\end{align*}
\]

\begin{align*}
\text{Input} & \\
\text{State} & \\
\text{Transition} & \\
\text{Boolean Network} & \\
\text{DDEs, } V_{\text{max}} = 20, n = 2 & \\
\text{DDEs, } V_{\text{max}} = 100, n = 2.5 & \\
\end{align*}

$\text{Average Error}$

$\text{Thresholded Error}$

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

\[
e_{\text{thresh}} = \begin{cases} 
0, & e_{avg} < 1/2, \\
1, & e_{avg} \geq 1/2.
\end{cases}
\]
FSMs for Cellular Information Processing

A. $t = 0$

B. $t = 45.4$

C. $t = 84.6$

D. $t = 144.9$

E. $t = 177.2$

F. $t = 211.7$
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:

- **stochastic pulse generator**
  - Turns on expression of a gene stochastically.

- **band pass filter**
  - Turns on expression of two different genes according to the concentration of a diffusible molecule.

- **timer**
  - Turns on expression of two different genes at times $t_1$ and $t_2$ after reset.
  - Can be reset by expressing a “reset” gene.
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

- **wave generator**
- **stochastic pulse generator**
- **time**
- **signal**
- **k**
- **start**

- **band pass filter**
  - **response**
  - **signal**

- **timer**
  - **reset**
  - **t**

- **edge detection**
  - **start**
  - **emit & reset**
  - **t**

- **toggle switch**
  - **on**, **off**

**A.**
- \( t = 0 \)
- \( t = 45.4 \)
- \( t = 84.6 \)
- \( t = 144.9 \)
- \( t = 177.2 \)
- \( t = 211.7 \)
Example: Microcolony Edge Detection
FSM Specification

stochastic pulse generator at rate $k$

emit... band pass filter
if $emit \in [r_1, r_2)$, $r_1 \to r_2$
if $emit \in [r_2, \infty)$, $r_2 \to \infty$

reset
on reset, $t := 0$
if $t = t_1$, $t_1 \to t_1$
if $t = t_2$, $t_2 \to t_2$

timer

Example: Microcolony Edge Detection
FSM Specification