Welcome to EE381V
Programming with Molecules

Spring 2017

T, Th 5:00-6:30pm
JES A305A
About me

Dr. David Soloveichik
david.soloveichik@utexas.edu

Office: POB 3.434

Education:
2002 BS in Computer Science (Harvard)
2008 PhD in Computation and Neural Systems (Caltech)
Spent time at UW and UCSF
My interests

**Natural computing:** models of computing inspired by nature
Computation is not a man-made phenomenon. From our brains to the regulatory networks of bacteria, nature provides fascinating examples of information processing, which is quite different from electronic computers.

**Molecular programming:** engineering smart molecules
Using nucleic-acid "strand displacement cascades" we build molecular interactions for synthetic biology, nanotechnology, and bioengineering in our wet-lab. We use chemistry as a "programming language".

**Distributed computing:** Formal models of distributed computing help us to discover the potential and limits of chemical information processing. We study models inspired by self-assembly and chemical reaction networks.
Teaching Assistant

Mengshi Zhang
Graduate Student (Electrical and Computer Engineering)
Research Interests: Software Testing, Data Mining

mengshi.zhang@utexas.edu
Office Hours

W 2:00-3:00pm in POB 3.434 (or by appointment)

W 4-5pm in POB 5.110

TA: Mengshi Zang
What this class is **NOT** about:

- computational chemistry
- simulating molecular structure and dynamics
- fast algorithms for simulating chemical kinetics

computational chemistry
What this class is **NOT** about:

**What is a DNA Computer?**

**INVENTOR:** Dr. Leonard Adleman

- DNA computer is a molecular computer that works biochemically to solve complex problems and different possible solutions are created all at once.
- It computes using enzymes that react with DNA strands and cause chain reactions.

creating a "chemical supercomputer"
What this class **IS** about:

“models of computing *inspired by* chemical and biological systems”

theory course: we will spend some time studying definitions, theorems and proofs
What this class IS about:

"Computer science and engineering has mastered complexity for electronic computation — can we do the same for engineering molecular devices and systems?"

"Molecular programming involves the specification of structures, circuits, and behaviors both within living and non-living systems—systems in which computing and decision-making will carried out by chemical processes themselves."
1. Chemical reaction networks
2. Algorithmic self-assembly
3. Neural Networks
1. Chemical reaction networks
Why Study CRNs?

- Fundamental model of chemical kinetics used in the natural sciences

- Fundamental model of population dynamics in ecology

- Sensor networks (population protocols)

- Fundamental mathematical structure: (vector addition systems, Petri nets, commutative semigroups, bounded context-free languages, uniform recurrence equations, ...)

$\text{CO}_2^{(aq)} + \text{H}_2\text{O} \leftrightarrow \text{H}_2\text{CO}_3$

$\Omega$
Why Understand Computation with CRNs?

- Embed programming into environments not compatible with traditional von Neumann computer architectures
formal model: CRNs

\[ A + X \rightarrow A + Y \]
\[ Y + Y \rightarrow X + X \]
\[ A \rightarrow X \]

distributed computing:
rate independent computation, probabilistic computation
dynamical systems
thermodynamics
Examples of Deterministic Computation

\[ f(x) = 2x \]
\[ X \rightarrow Y + Y \]

\[ f(x_1, x_2) = \min(x_1, x_2) \]
\[ X_1 + X_2 \rightarrow Y \]
\[ X_1 \rightarrow Y \]
\[ X_2 \rightarrow Y \]

\[ f(x_1, x_2) = \max(x_1, x_2) \]
\[ X_1 \rightarrow L_1 + Y \]
\[ X_2 \rightarrow L_2 + Y \]
\[ L_1 + L_2 \rightarrow K \]
\[ Y + K \rightarrow \emptyset \]

Note: \( \max(x_1, x_2) = x_1 + x_2 - \min(x_1, x_2) \)
Examples of Questions We'll Ask:

Note that these computations are correct no matter what order the reactions happen to occur.
Which functions can be computed this way?

How does allowing some probability of error increase the computational power?

Which functions can be computed fast?
Lotka Volterra aka Predator-Prey

\[ X_1 + X_2 \rightarrow 2 \ X_2 \]
\[ X_1 \rightarrow 2 \ X_1 \]
\[ X_2 \rightarrow \emptyset \]
Can we understand information processing in biological regulatory pathways?

“Hairball”

protein interaction network of Syphilis bacteria (Treponema pallidum)
Example: Approximate Majority
distributed algorithm design problem

• n people in a room
• each starts with belief X (Dem) or Y (Rep)
• goal: quickly reach consensus, with initial majority winning whp
• pairwise interaction rules; any pair of people equally likely to interact next
Example: Approximate Majority

\[
\begin{align*}
X + Y &\rightarrow X + B \\
X + Y &\rightarrow B + Y \\
X + B &\rightarrow X + X \\
Y + B &\rightarrow Y + Y
\end{align*}
\]

“Chemical Caucus”

Angluin, Aspnes, Eisenstat DISC’07
Example: Approximate Majority

The expected time to converge is provably $\Theta(\log n)$ interactions per person.

Starting configuration: half $X$ and half $Y$.

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X + Y \rightarrow X + B$</td>
<td></td>
</tr>
<tr>
<td>$X + Y \rightarrow B + Y$</td>
<td></td>
</tr>
<tr>
<td>$X + B \rightarrow X + X$</td>
<td></td>
</tr>
<tr>
<td>$Y + B \rightarrow Y + Y$</td>
<td></td>
</tr>
</tbody>
</table>
Approximate Majority in Biology

“Epigenetic Memory by Nucleosome Modification”

Dodd, Micheelsen, Sneppen, Thon, Cell 129, 813-822 (2007)
Approximate Majority Emulation Zoo

Luca Cardelli, “Morphisms of reaction networks that couple structure to function” (2014)
Can We Build Arbitrary Chemical Reaction Networks?

- Molecules are complexes of DNA strands
- Interactions programmed by choice of sequence (Watson-Crick binding)

"DNA strand displacement cascades" are **complete** for chemical reaction networks

Soloveichik, Seelig, Winfree *PNAS* 2010
Strand displacement in the lab

molecular logic circuits

- Largest autonomous biochemical networks built from scratch
  Qian, Winfree, Science 2011

controlling assembly of nanoscale structures

- Prescribed nanoscale structures seen under atomic force microscope
  Yin, Choi, Calvert, Yurke, Pierce Nature 2008

molecular artificial neural networks

- Biochemical system doing inference
  Qian, Winfree, Bruck Nature 2011

strand displacement cascades

strand displacement in mammalian cells

- Logic on biological signals
  Hemphill, Deiters J Am Chem Soc 2013
2. Algorithmic Self-Assembly
Motivation 1: Understand Algorithmic Principles of Development
Motivation 2:
Bottom-up Fabrication
Another example

The “discrete Sierpinski triangle”
Fig. 2. Using a binary counter to self-assemble a demultiplexer. Logic levels for an example input-output pair are shown: only the row that exactly matches the input pattern is set to “1”. To make a pattern with \( N \) rows, \( 10 + \log N \) tiles are used.
Example of Questions We'll Ask:

Which patterns/structures are easy to self-assembly (require few tile types) and which are hard (require many tile types)?

Is cooperatively (requiring multiple bonds) necessary for complex patterns/structures?
DNA tiles

Ned Seeman
Sierpinski Triangle

Crystal that thinks as it grows, which controls the growth process
3. Neural Networks
Idealized Neuron

100 billion neurons in human brain
Linear Threshold Gate
(McCulloch-Pitts Model)

\[ f(x) \in \{0, 1\} \]

\[ F(X) = w_0 + \sum_{i=1}^{n} w_i x_i \]

\[ f(X) = \text{sgn} (F(X)) = \begin{cases} 
0 & \text{if } F(X) < 0 \\
1 & \text{if } F(X) \geq 0
\end{cases} \]
### Complexity of LT circuits vs AON circuits

**Linear Threshold gates**

**AND, OR, NOT gates** (unbounded fan-in/out)

#### XOR (parity) computation over $n$ variables:

<table>
<thead>
<tr>
<th></th>
<th>depth 2</th>
<th>arbitrary depth</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AON</strong></td>
<td>$2^{n-1} + 1$</td>
<td>$\lceil 2.5(n - 1) \rceil$</td>
</tr>
<tr>
<td></td>
<td>(optimal)</td>
<td>(lower-bound: $2n-1$)</td>
</tr>
<tr>
<td><strong>LT</strong></td>
<td>$\lceil n/2 \rceil + 1$</td>
<td>$\lceil \log_2(n + 1) \rceil$</td>
</tr>
<tr>
<td></td>
<td>(optimal)</td>
<td>(optimal)</td>
</tr>
</tbody>
</table>
Some Models of Learning

Perceptron Learning Algorithm

Hopfield Networks

Boltzmann Machines

Can we capture some key aspects of human memory in a formal model?
Software Tools We'll Use

(may change)
Free license:

For students from Cockrell School of Engineering:  
http://www.engr.utexas.edu/itg/products/548-mathematica-30128485

For students from College of Natural Science:  
http://license.cns.utexas.edu/
Guarded Command Programming + Cell Signaling + Micro-Colony Simulation

What can you gro?

```plaintext
p.mode = 4 & (get_signal(s1) > 0.5 & get_signal(s2) > 0.5):
  set("growth_rate", 0.1)
```
Tile Assembly Simulator

ISU TAS:
(Windows, Linux, OS X)
Prerequisites

• **Formal proofs (discrete math)**
• Undergraduate probability
• Digital circuits, automata, elementary differential equations

No biology or chemistry background is assumed
Grading

Weekly Homework (65%)
Group Project (30%)
Class Participation (5%)

Homework is due at the beginning of class. The penalty for late homework will be \(-20\%\) per 24 hour period. This penalty is assessed after normal grading and is \textit{cumulative} with any points lost (e.g. a homework that would normally receive 80\% of total points would receive only 40\% if handed in within 48 hours). Late homework must be submitted via e-mail to the instructor and TA.
Course Materials on Canvas