



Homework # 2

Due March 27, 2015

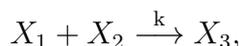
The theory of reaction kinetics underpins our understanding of biological and chemical systems. It is a simple and elegant formalism: chemical reactions define rules according to which reactants form products; each rule fires at a rate that is proportional to the quantities of the corresponding reactants that are present. On the computational front, there has been a wealth of research into efficient methods for simulating chemical reactions, ranging from ordinary differential equations (ODEs) to stochastic simulation. On the mathematical front, entirely new branches of theory have been developed to characterize the dynamics of chemical reaction networks.

Most of this work is from the vantage point of analysis: a set of chemical reaction exists, designed by nature and perhaps modified by human engineers; the objective is to understand and characterize its behavior. Comparatively little work has been done at a conceptual level in tackling the inverse problem of synthesis: how can one design a set of chemical reactions that implement specific behavior?

This homework will consider the computational power of chemical reactions from both a deductive and an inductive point of view.

Analyzing Chemical Reaction Networks

A molecular system consists of a set of chemical reactions, each specifying a rule for how types of molecules combine. For instance,

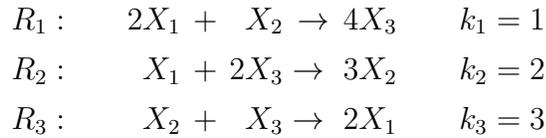


specifies that one molecule of X_1 combines with one molecule of X_2 to produce one molecule of X_3 . The rate at which the reaction fires is proportional to (1) the concentrations of the participating molecular types; and (2) a rate constant k . (This value is not constant at all; rather it is dependent on factors such as temperature and volume; however, it is independent of molecular quantities, and so called a “constant.”)

Given a set of such reactions, we can model the behavior of the system in two ways:

1. In a **continuous** sense, in terms of molecular concentrations, with differential equations;
2. In a **discrete** sense, in terms of molecular quantities, through probabilistic discrete-event simulation.

Consider the reactions:



For a continuous model, let x_1, x_2 and x_3 denote the concentrations of X_1, X_2 , and X_3 , respectively. (Recall that concentration is number of molecules per unit volume.) The behavior of the system is described by the following set of differential equations:

$$\begin{aligned} \frac{dx_1}{dt} &= -x_1^2 x_2 - 2x_1 x_3^2 + 6x_2 x_3 \\ \frac{dx_2}{dt} &= -x_1^2 x_2 + 6x_1 x_3^2 - 3x_2 x_3 \\ \frac{dx_3}{dt} &= 4x_1^2 x_2 - 2x_1 x_3^2 - 3x_2 x_3 \end{aligned}$$

For the discrete model, let the state be $\mathbf{S} = [x_1, x_2, x_3]$, where x_1, x_2 and x_3 denote the numbers of molecules of types X_1, X_2 , and X_3 , respectively. (Here we use actual integer quantities, not concentrations.) The firing probabilities for R_1, R_2 and R_3 are computed as follows:

$$\begin{aligned} p_1(x_1, x_2, x_3) &= \frac{\frac{1}{2}x_1(x_1 - 1)x_2}{\frac{1}{2}x_1(x_1 - 1)x_2 + x_1 x_3(x_3 - 1) + 3x_2 x_3}, \\ p_2(x_1, x_2, x_3) &= \frac{x_1 x_3(x_3 - 1)}{\frac{1}{2}x_1(x_1 - 1)x_2 + x_1 x_3(x_3 - 1) + 3x_2 x_3}, \\ p_3(x_1, x_2, x_3) &= \frac{3x_2 x_3}{\frac{1}{2}x_1(x_1 - 1)x_2 + x_1 x_3(x_3 - 1) + 3x_2 x_3}. \end{aligned}$$

Suppose that $\mathbf{S} = [3, 3, 3]$. Then the firing probabilities for R_1, R_2 and R_3 are

$$\begin{aligned} p_1(3, 3, 3) &= \frac{9}{9 + 18 + 27} = \frac{1}{6}, \\ p_2(3, 3, 3) &= \frac{18}{9 + 18 + 27} = \frac{1}{3}, \\ p_3(3, 3, 3) &= \frac{27}{9 + 18 + 27} = \frac{1}{2}, \end{aligned}$$

respectively.

N.B. In the continuous model, the rate of change of type is proportional to x^n where x is the concentration of a reaction and n is the coefficient. In the discrete model, the probability is proportional to $\binom{x}{n}$. This is a subtle difference. See the paper by Gillespie for an explanation.

Problem [5 points]

Suppose that we define the following “outcomes”:

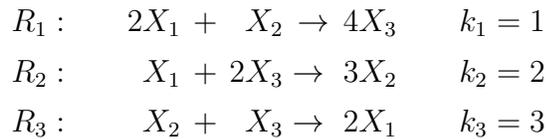
- C_1 : states $\mathbf{S} = [x_1, x_2, x_3]$ with $x_1 > 5$.
- C_2 : states $\mathbf{S} = [x_1, x_2, x_3]$ with $x_2 \geq 7$.
- C_3 : states $\mathbf{S} = [x_1, x_2, x_3]$ with $x_3 < 5$.

Beginning from the state $\mathbf{S} = [3, 5, 7]$, compute (or estimate) $\Pr(C_1)$, $\Pr(C_2)$, and $\Pr(C_3)$.

□

Now, instead of “outcomes”, let’s analyze probabilities in a more general sense.

Consider again the reactions:



Let the state be $\mathbf{S} = [x_1, x_2, x_3]$, where x_1, x_2 and x_3 denote the numbers of molecules of types X_1, X_2 , and X_3 , respectively.

Suppose that systems begins in the state $S = [3, 3, 3]$ (with probability 1). After one step:

- it is in state $[1, 2, 7]$ with probability $\frac{1}{6}$.
- it is in state $[2, 6, 1]$ with probability $\frac{1}{3}$.
- it is in state $[5, 2, 2]$ with probability $\frac{1}{2}$.

So, considering the first type, X_1 , its discrete probability distribution after one step is

- $Pr[X_1 = 1] = \frac{1}{6}$,
- $Pr[X_1 = 2] = \frac{1}{3}$,
- $Pr[X_1 = 5] = \frac{1}{2}$,

After many steps, the system can be in many different states, with different quantities of X_1 . (Of course, some of these states may have the same quantity of X_1 .) The probability distribution may look something like:

- $Pr[X_1 = 0] = 0.012$,
- $Pr[X_1 = 1] = 0.025$,
- $Pr[X_1 = 2] = 0.036$,
- $Pr[X_1 = 3] = 0.061$,

- $Pr[X_1 = 4] = 0.12,$
- $Pr[X_1 = 5] = 0.19,$
- $Pr[X_1 = 6] = 0.24,$
- $Pr[X_1 = 7] = 0.19,$
- $Pr[X_1 = 8] = 0.116,$
- $Pr[X_1 = 9] = 0.010.$

(Note that this is not a real calculation.)

Problem [5 points]

For the set of reactions above, beginning from the state $[3, 5, 7]$ compute (or estimate) the mean and variance for the probability distributions of X_1 , X_2 and X_3 – each separately – after 5 steps. □

Randomness is inherent to all forms of biochemical computation: at any given instant, the choice of which reaction fires next is a matter of chance. Certain biochemical systems appear to exploit this randomness for evolutionary advantage, choosing between different outcomes with a probability distribution – in effect, hedging their bets with a portfolio of responses that is carefully tuned to the environmental conditions.

For instance, the *lambda* bacteriophage, a virus that infects the *E. coli* bacteria, chooses one of two survival strategies: either it integrates its genetic material with that of its host and then replicates when the bacterium divides (call this “stealth” mode); or else it manipulates the molecular machinery of its host to make many copies of itself, killing the bacterium in the process, and thereby releasing its progeny into the environment (call this the “hijack” strategy). The choice of which strategy to pursue, while based on environmental inputs, is probabilistic: in some cases, the virus chooses the first strategy, say with probability p and the second with probability $1 - p$. Clearly the virus is hedging its bets, an approach that provides significant advantages in an evolutionary context.

A model for the biochemistry of *lambda* is at:

<http://cctbio.ece.umn.edu/ee5393-2015-spring/lambda.r>

A set of initial values for the molecular types is at:

<http://cctbio.ece.umn.edu/ee5393-2015-spring/lambda.in>

With this model, we can assume that *lambda* has entered “stealth” mode when $cI2 > 145$; it has entered “hijack” mode when $Cro2 > 55$.

Problem [5 points]

For the reactions and initial values given, compute the probability that *lambda* has entered “stealth” mode vs. “hijack” mode for a range of values of the molecular type $MOI = 1 \dots 10$.

For this problem, you can either use your own code for simulating chemical reactions from Homework 1 or use my code:

<http://cctbio.ece.umn.edu/ee5393-2014-spring/aleae.zip>.

□

Synthesizing Chemical Reaction Networks

The task of synthesizing a set of chemical reactions to compute a desired function is conceptually open-ended. Like most engineering problems, there are often many possible solutions.

In class, we saw a chemical reaction network that performs multiplication. What follows are some other examples of chemical reaction networks that compute simple functions. In describing the functions that the modules implement, we add subscripts to the quantities of molecular types to denote *when* these quantities exist: zero indicates that this is the initial quantity, whereas infinity indicates that it is the quantity after the module has finished.

Exponentiation:

$$Y_{\infty} = 2^{X_0}$$

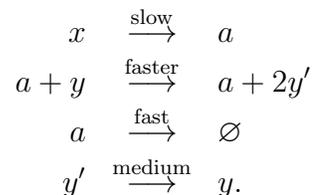
This module consumes molecules of an input type one at a time, doubling the quantity of an output type for each. Its behavior is described by the following pseudocode:

```

1  ForEach x {
2    Y = 2 * Y;
3  }.

```

The reactions are:



Initially, Y is one and all other quantities (except X) are zero.

Logarithm:

$$Y_{\infty} = \log_2(X_0)$$

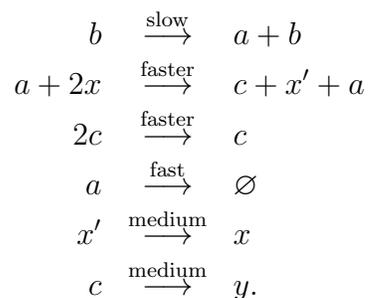
This module is similar to the exponentiation module, except that instead of doubling the output, the input is forced to halve itself; each time it does so, the output is incremented by one. Its behavior is described by the following pseudocode:

```

1  While Not(X==1) {
2    X = X/2;
3    Y = Y+1;
4  }.

```

The reactions are:



Initially, B is a small but non-zero quantity and all other quantities (except X) are zero.

Problem [10 points]

Produce a chemical reaction network that computes

$$Z_{\infty} = X_0 \log_2 Y_0$$

Demonstrate that your solution works (either mathematically, or by simulating it continuously or stochastically).

Problem [10 points]

Produce a chemical reaction network that computes

$$Y_{\infty} = 2^{\log_2 X_0} \tag{1}$$

$$\tag{2}$$

(No points for noticing that $Y_{\infty} = X_0$. Your network must compute this as shown!) Demonstrate that your solution works (either mathematically, or by simulating it continuously or stochastically).

Problem [15 points]*(no collaboration)*

Consider the following representation of real numbers. A real value x between 0 and 1 is represented as $\frac{x_1}{x_1+x_2}$, where x_1 and x_2 are positive integers.

Construct a set of chemical reactions that multiplies two real numbers a and b represented this way, producing a resulting number c , also represented this way. Let the quantities or concentrations of molecular types A_1 and A_2 represent a , those of B_1 and B_2 represent b , and those of C_1 and C_2 represent c . Demonstrate that your solution works (either mathematically, or by simulating it continuously or stochastically).

Two-Terminal Switches

This problem is based on C. E. Shannon's Masters Thesis from 1938, "A Symbolic Analysis of Relay and Switching Circuits." This classical paper was the first work that connected between logic and circuit design.

In his seminal Master's Thesis, Claude Shannon made the connection between Boolean algebra and switching circuits. He considered two-terminal switches corresponding to electromagnetic relays. An example of a two-terminal switch is shown in the top part of Figure 1. The switch is either ON (closed) or OFF (open). A Boolean function can be implemented in terms of connectivity across a network of switches, often arranged in a series/parallel configuration. An example is shown in the bottom part of Figure 1.

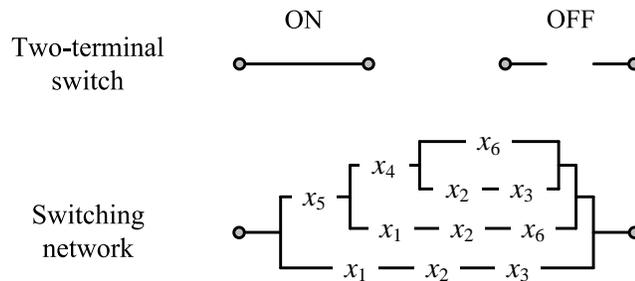


Figure 1: Two-terminal switching network implementing the Boolean function $x_1x_2x_3 + x_1x_2x_5x_6 + x_2x_3x_4x_5 + x_4x_5x_6$.

Each switch is controlled by a Boolean variable. If the variable is 1 (0) then the corresponding switch is ON (OFF). The Boolean function for the network evaluates to 1 if there is a closed path between the left and right nodes. It can be computed by taking the sum (OR) of the product (AND) of literals along each path. The function is

$$x_1x_2x_3 + x_1x_2x_5x_6 + x_4x_5x_2x_3 + x_4x_5x_6.$$

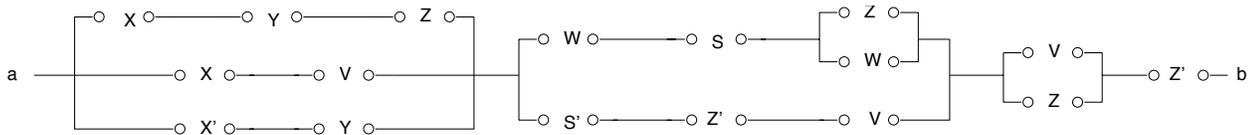
We use the modern convention: 1 denotes a closed circuit, 0 denotes an open circuit, + (addition) denotes a parallel connection and \cdot (multiplication) denotes a series connection. (Shannon used the opposite of the modern convention for everything: 1 for open, + for a series connection, etc. Forget about that.)

Using De Morgan's Theorem and the theorems given on page 476 of Shannon's Masters Thesis, simplify the following circuits as much as possible and draw the simplified

circuits. The number of switching elements achievable in the final answer is given for you.

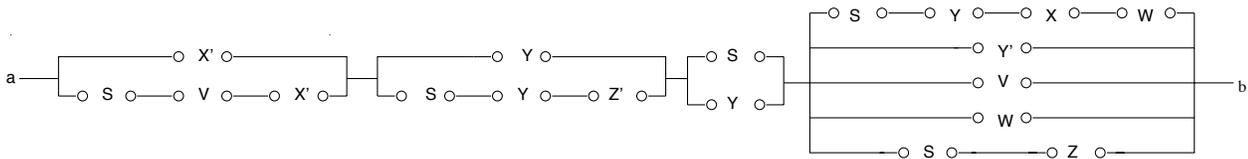
Problem [5 points]

This circuit can be simplified down to 6 switching elements. (Note that 4 of the switching elements shown are for negated variables.)



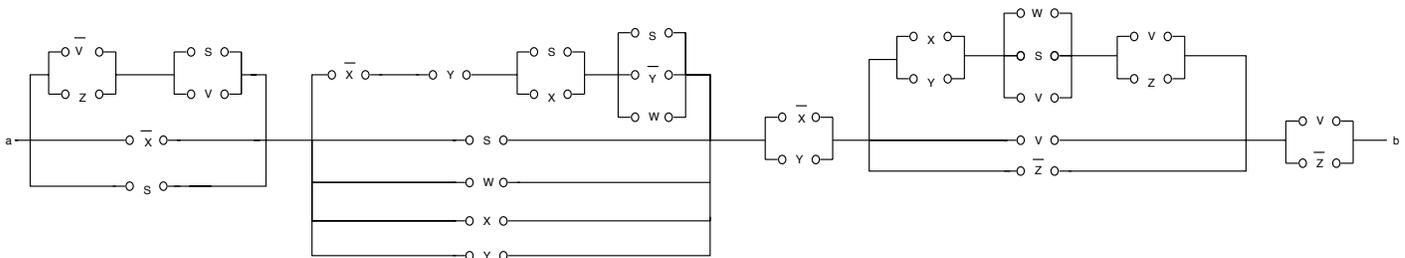
Problem [5 points]

This circuit can be simplified down to 6 switching elements. (Note that 4 of the switching elements shown are for negated variables.)



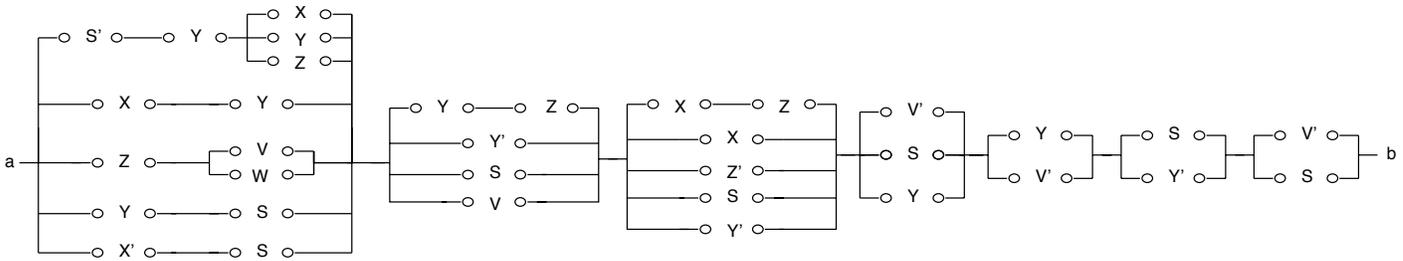
Problem [5 points]

This circuit can be simplified down to 9 switching elements. (Note that 7 of the switching elements shown are for negated variables.)



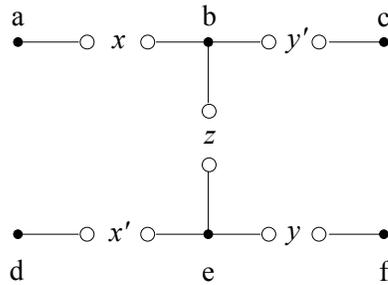
Problem [5 points]

This circuit can be simplified down to 8 switching elements (7 for extra credit). (Note that 9 of the switching elements shown are for negated variables.)



Two-Terminal Switches, but Multiple Circuit Terminals (no collaboration)

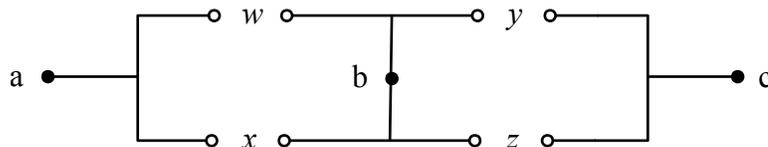
A generalization of the switching circuit model is a circuit with *multiple circuit terminals*. A function F_{ab} is 1 if there is a closed path between terminals a and b , and 0 otherwise. With multiple circuit terminals, a, b, c, d, \dots , different functions can be implemented between *pairs* of terminals. For instance, for the circuit



we have

$$\begin{aligned}
 F_{af} &= xyz \\
 F_{bd} &= x'z \\
 F_{cf} &= 0.
 \end{aligned}$$

and so on. For the circuit



we have

$$\begin{aligned}F_{ab} &= x + w \\F_{bc} &= y + z \\F_{ac} &= (x + w)(y + z).\end{aligned}$$

As these examples show, you can use either a variable or its complement to control a switch. (Each terminal is, in fact, any stretch of wire.)

Problem [**5 points**]

Construct a circuit with 3 switches that implements the functions

$$\begin{aligned}f_1 &= xy \\f_2 &= x'y\end{aligned}$$

Problem [**5 points**]

Construct a circuit with 4 switches that implements the functions

$$\begin{aligned}f_1 &= xy + x'y' \\f_2 &= x'y + xy'\end{aligned}$$

Problem [**5 points**]

Construct a circuit with 6 switches that implements the functions

$$\begin{aligned}f_1 &= x(y + z) \\f_2 &= y(x + z) \\f_3 &= z(x + y) \\f_4 &= x + yz \\f_5 &= y + xz \\f_6 &= z + xy\end{aligned}$$

Problem [**15 points**]

Construct a circuit with as few switches as possible that implements all functions of

two variables. A solution with 8 switches gets full credit. These functions are

$$\begin{aligned}f_0 &= 0 \\f_1 &= x \\f_2 &= x' \\f_3 &= y \\f_4 &= y' \\f_5 &= xy \\f_6 &= xy' \\f_7 &= x'y \\f_8 &= x'y' \\f_9 &= x + y \\f_{10} &= x + y' \\f_{11} &= x' + y \\f_{12} &= x' + y' \\f_{13} &= xy' + x'y \\f_{14} &= xy + x'y' \\f_{15} &= 1\end{aligned}$$

The constant 1 function is trivial: choose same terminal, e.g., connect a back to a . So forget about that one. For all the others, including the constant 0 function, you must select a pair of distinct terminals. As in the example above, the constant 0 function is implemented between a pair of terminals that are never connected.