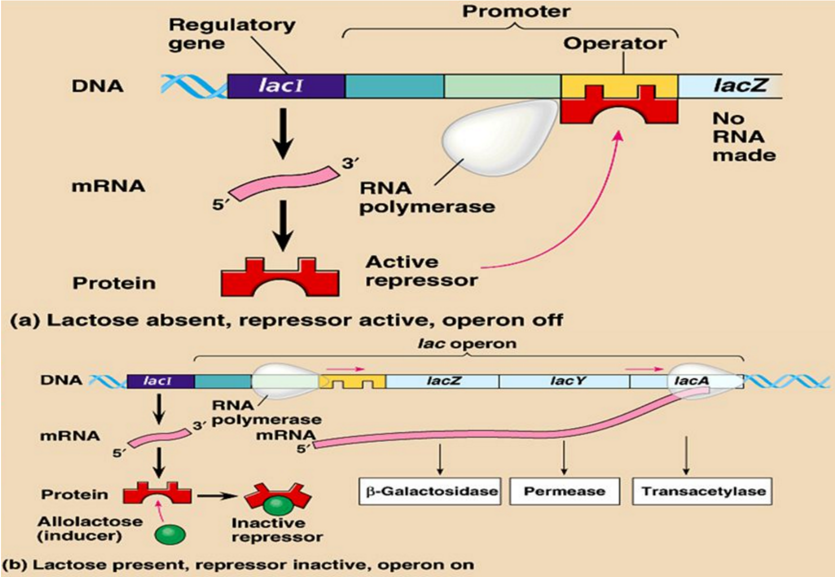
Fixed points of Boolean models

Matthew Macauley

Department of Mathematical Sciences
Clemson University
http://www.math.clemson.edu/~macaule/

Algebraic Systems Biology

The *lac* operon in *E. coli*



A 9-variable model of the *lac* operon (Chapter 1 of Robeva/Hodge, 2013)

Assumptions:

- Transcription and translation require 1 time step
- Degradation of mRNA and proteins take 1 time step
- High levels of lactose or allolactose imply (at least) medium levels in the next time step.

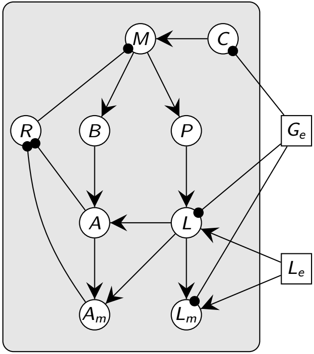
Variables:

- M (mRNA): $f_M = \overline{R} \wedge C$
- P (*lac* permease): $f_P = M$
- B (β-galactosidase): $f_B = M$
- C (catabolite activator protein, CAP): $f_C = \overline{G_e}$
- R (LacI repressor protein): $f_R = \overline{A} \wedge \overline{A_m}$
- A (high allolactose): $f_A = L \wedge B$
- A_m (at least medium allolactose): $f_{A_m} = A \vee L \vee L_m$
- L (high intracellular lactose): $f_L = \overline{G_e} \wedge P \wedge L_e$
- L_m (at least medium intracellular lactose): $f_{L_m} = \overline{G_e} \wedge (L \vee L_e)$

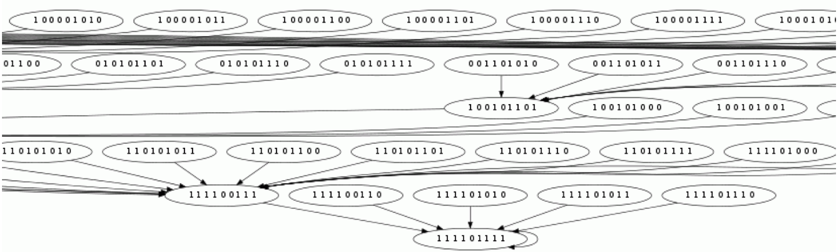
Parameters:

- G_e (high extracellular glucose): $f_{G_e} = G_e$
- L_e (high extracellular lactose): $f_{L_e} = L_e$

A 9-variable model of the *lac* operon (Chapter 1 of Robeva/Hodge, 2013)



$$
\begin{cases}
f_M = \overline{R} \wedge C \\
f_P = M \\
f_B = M \\
f_C = \overline{G_e} \\
f_R = \overline{A} \wedge \overline{A_m} \\
f_A = L \wedge B \\
f_{A_m} = A \vee L \vee L_m \\
f_L = \overline{G_e} \wedge P \wedge L_e \\
f_{L_m} = \overline{G_e} \wedge (L \vee L_e)
\end{cases}
$$



Finding the fixed points

The previous 9-variable model is about as big as Cyclone can handle.

However, many gene regulatory networks are much bigger. For example:

- A Boolean model (2006) of T helper cell differentiation has 23 nodes, and thus a state space of size $2^{23} = 8{,}388{,}608$.
- A Boolean model (2003) of the segment polarity genes in *Drosophila melanogaster* (fruit fly) has 60 nodes, and a state space of size $2^{60} \approx 1.15 \times 10^{18}$.

For these systems, we need to be able to analyze them without constructing the entire state space.

Our first goal is to find the fixed points. This amounts to solving a system of equations:

$$\begin{aligned} f_{x_1} &= x_1 \\ f_{x_2} &= x_2 \\ &\vdots \\ f_{x_n} &= x_n. \end{aligned}$$

This is a problem from computational algebraic geometry, over the finite field $\mathbb{F}_2 = \{0, 1\}$.

Finding the fixed points

Let's rename variables $(M, P, B, C, R, A, A_m, L, L_m) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$.

The fixed points are solutions to the following system of equations:

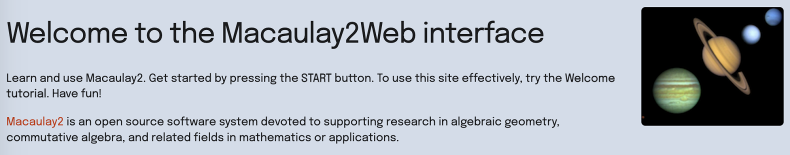
$$
\begin{array}{ll}
f_M = \overline{R} \wedge C = M & x_1 + x_4x_5 + x_4 = 0 \\
f_P = M = P & x_1 + x_2 = 0 \\
f_B = M = B & x_1 + x_3 = 0 \\
f_C = \overline{G_e} = C & x_4 + G_e + 1 = 0 \\
f_R = \overline{A} \wedge \overline{A_m} = R & x_5 + x_6x_7 + x_6 + x_7 + 1 = 0 \\
f_A = L \wedge B = A & x_6 + x_3x_8 = 0 \\
f_{A_m} = A \vee L \vee L_m = A_m & x_6 + x_7 + x_8 + x_9 + x_8x_9 + x_6x_8 + x_6x_9 + x_6x_8x_9 = 0 \\
f_L = \overline{G_e} \wedge P \wedge L_e = L & x_8 + x_2L_e(G_e + 1) = 0 \\
f_{L_m} = \overline{G_e} \wedge (L \vee L_e) = L_m & x_9 + (G_e + 1)(x_8 + x_6L_e + L_e) = 0
\end{array}
$$

We need to solve this system for all 4 possible parameter vectors:

$$(G_e, L_e) = (0,0),\ (0,1),\ (1,0),\ \text{and}\ \ (1,1).$$

Finding the fixed points using computational algebra

The Macaulay2 software system was written for researchers in algebraic geometry and commutative algebra.



It is freely available online:

`https://www.unimelb-macaulay2.cloud.edu.au/`

If we want to work in the polynomial ring $\mathbb{F}_2[x_1, \dots, x_9]$, we can type in:

```
R = ZZ/2[x1,x2,x3,x4,x5,x6,x7,x8,x9];
```

However, since $x_i^2 = x_i$ as functions, we want to work in the quotient ring $Q = R/J$:

```
J = ideal(x1^2-x1, x2^2-x2, x3^2-x3, x4^2-x4, x5^2-x5, x6^2-x6, x7^2-x7, x8^2-x8, x9^2-x9);
Q = R / J;
```

Finding the fixed points with Macaulay2, for $(G_e, L_e) = (0, 1)$

It is helpful to define a shortcut for AND and OR operators:

```
RingElement | RingElement :=(x,y)->x+y+x*y;
RingElement & RingElement :=(x,y)->x*y;
```

Next, let's set the parameters (constants), assuming low glucose and high lactose.

```
Ge = 0_Q
Le = 1_Q
```

Now we can define the functions of our 9-variable lac operon model.

```
f1 = (1+x5) & x4;
f2 = x1;
f3 = x1;
f4 = 1+Ge;
f5 = (1+x6) & (1+x7);
f6 = x8 & x3;
f7 = x6 | x8 | x9;
f8 = (1+Ge) & x2 & Le;
f9 = (1+Ge) & (x8 | Le);
```

The semicolons are optional. They supress the output being displayed.

Finding the fixed points with Macaulay2, for $(G_e, L_e) = (0, 1)$

We want to solve the system of nonlinear polynomials $\{f_1 + x_1 = 0, \ldots, f_9 + x_9 = 0\}$.

To do this, define the ideal generated by the polynomials $f_i + x_i$:

```
I = ideal(f1+x1, f2+x2, f3+x3, f4+x4, f5+x5, f6+x6, f7+x7, f8+x8, f9+x9)
```

Finally, compute a Gröbner basis of this ideal:

```
G = gens gb I
```

The output will look like this:

```
|x9+1, x8+1, x7+1, x6+1, x5, x4+1, x3+1, x2+1, x1+1|
```

This means that the following (much simpler!) system has same solution set:

$$\{x_9+1 = 0,\ x_8+1 = 0,\ x_7+1 = 0,\ x_6+1 = 0,\ x_5 = 0,\ x_4+1 = 0,\ x_3+1 = 0,\ x_2+1 = 0,\ x_1+1 = 0\}$$

Since we're working over $\mathbb{F}_2 = \{0, 1\}$, there is one solution:

$$(M, P, B, C, R, A, A_m, L, L_m) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) = (1, 1, 1, 1, 0, 1, 1, 1, 1).$$

This makes biological sense—the operon is ON.

Finding the fixed points with Macaulay2

Using the variables

$$(M, P, B, C, R, A, A_m, L, L_m) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$$

we can rerun the previous steps for the other three choices of parameter vector.

It is straightforward to check that there is a unique fixed points in each case:

- Parameter vector: $(G_e, L_e) = (0, 0)$

  Fixed point: $(0, 0, 0, 1, 1, 0, 0, 0, 0)$.

  Operon: OFF.

- Parameter vector: $(G_e, L_e) = (1, 0)$

  Fixed point: $(0, 0, 0, 0, 1, 0, 0, 0, 0)$.

  Operon: OFF.

- Parameter vector: $(G_e, L_e) = (1, 1)$

  Fixed point: $(0, 0, 0, 0, 1, 0, 0, 0, 0)$.

  Operon: OFF.

- Parameter vector: $(G_e, L_e) = (0, 1)$

  Fixed point: $(1, 1, 1, 1, 0, 1, 1, 1, 1)$.

  Operon: ON.

In each case, this is exactly what we expect biologically.

An alternate way to enter this model in Macaulay2

Another way to handle parameters is to treat them as variables that don't change.

For example, to work in the polynomial ring $\mathbb{F}_2[x_1, \ldots, x_9, G_e, L_e]$, we can type in:

```
f1 = (1+x5) & x4;
f2 = x1;
f3 = x1;
f4 = 1+Ge;
f5 = (1+x6) & (1+x7);
f6 = x8 & x3;
f7 = x6 | x8 | x9;
f8 = (1+Ge) & x2 & Le;
f9 = (1+Ge) & (x8 | Le);
Ge = Ge;
Le = Le;
```

The resulting Boolean model will have $2^{11} = 2048$ states, and it should have 4 fixed points.

We will leave the details as a HW exercise.

Gröbner bases vs. Gaussian elimination

A Gröbner basis is a special type of basis for an ideal of a polynomial ring.

It can be used as a generalization of Gaussian elimination, but for systems of nonlinear equations (i.e., polynomials).

In both cases:

- The input is a complicated system that we wish to solve.
- The output is a simple system that we can easily solve by hand.

Example. Consider the system $\begin{cases} x + 2y = 1 \\ 3x + 8y = 1 \end{cases}$

Gaussian elimination yields the following:

$$\left[\begin{array}{cc|c} 1 & 2 & 1 \\ 3 & 8 & 1 \end{array}\right] \longrightarrow \left[\begin{array}{cc|c} 1 & 2 & 1 \\ 0 & 2 & -2 \end{array}\right] \longrightarrow \left[\begin{array}{cc|c} 1 & 0 & 3 \\ 0 & 2 & -2 \end{array}\right] \longrightarrow \left[\begin{array}{cc|c} 1 & 0 & 3 \\ 0 & 1 & -1 \end{array}\right]$$

This is just a much simplier system with the same solution: $\begin{cases} x + 0y = 3 \\ 0x + y = -1 \end{cases}$

Back-substitution and Gaussian elimination

We don't need to do Gaussian elimination until the matrix is the identity; it only need be upper-triangular.

For example: $\begin{cases} x+z=2 \\ y-z=8 \\ 0=0 \end{cases}$

Similarly, when computational algebra software outputs a Gröbner basis, it will be in "upper-triangular form," and we can solve the system easily by back-substituting.

We'll do an example next, but for now, you can think of Gröbner bases as a mysterious "black box" that does what we want.

Later, as time allows, we might study them in more detail and understand what's going on behind the scenes.

Back-substitution and Gaussian elimination

Let's use computational algebra to solve the following system: $\begin{cases} x^2 + y^2 + z^2 = 1 \\ x^2 - y + z^2 = 0 \\ x - z = 0 \end{cases}$

```
R = RR[x,y,z]
I = ideal(x^2+y^2+z^2-1, x^2-y+z^2, x-z)
gens gb I
```

This gives an output of:

$$\begin{pmatrix} x - z & z^2 - .5y & y^2 + y - 1 \end{pmatrix}$$

This means that $y = \dfrac{-1 \pm \sqrt{5}}{2}$, and hence $x = z = \pm\sqrt{\dfrac{-1 + \sqrt{5}}{4}}$.

Note that there would be two additional solutions over $\mathbb{C}$. (*Why?*)

Exercise. What are the solutions over the following fields, given the Gröbner bases shown:

- $\mathbb{F}_2 = \{0, 1\}$: $\quad (1)$
- $\mathbb{F}_3 = \{0, 1, 2\}$: $\quad \begin{pmatrix} x - z & z^2 + y & y^2 + y - 1 \end{pmatrix}$