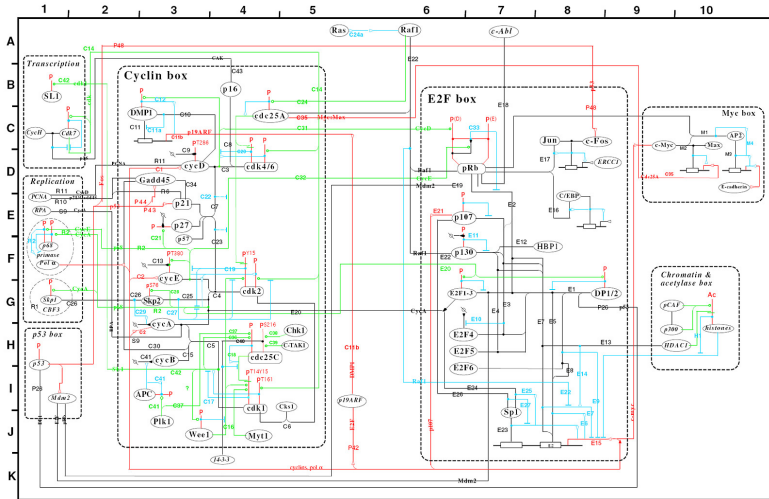# June 11th, 2010 - Boolean Networks

Bonny Guang, Madison Brandon, Rustin McNeill

June 11th, 2010

A Real-Life Wiring Diagram

Figure: Mammalian cell cycle with at least 60 nodes.

State Space Diagrams
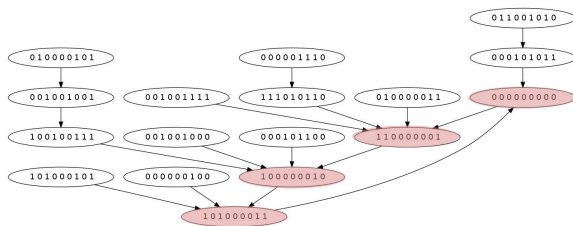
- $2^{60} \approx 1.2 \times 10^{18}$ states. Let's scale down to $n = 9$ nodes. (DVD online can only take 1000 states)
- Even for $n = 9$, when we try to see the state space graph...

Figure: Boolean state space graph, 9 nodes.

Impossible to even see the 9 states on one screen!

Figure: Cropping of 3% of graph
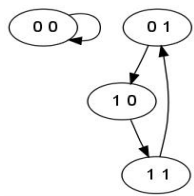
Simple Example

Figure: State space for a 2 by 2 system

x1	x2	f1	f2
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

Figure: Truth table for 2 by 2 system

Simple Example

$$f_1 = \begin{cases} (0,0) & \rightarrow & 0 \\ (0,1) & \rightarrow & 1 \\ (1,0) & \rightarrow & 1 \\ (1,1) & \rightarrow & 0 \end{cases}$$

with columns labeled **C** and **B**.

We can write any boolean function defined in this way as a polynomial $f : \mathbb{F}_2^2 \rightarrow \mathbb{F}_2^2$ where

$$f_1(x1, x2) = \sum_{i=1}^{4} b_i \prod_{j=1}^{2} (1 - (c_{i,j} - x_i))$$

where $c_{i,j}$ are the i,jth entries into the matrix C and b_i are the ith entries into the vector B.

Simple Example
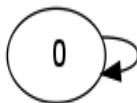
Then

$$
\begin{aligned}
f_1 = {}& 0 * (1 - (0 - x_1)(1 - (0 - x_1)) \\
& + 1 * (1 - (0 - x_1))(1 - (1 - x_2)) \\
& + 1 * (1 - (1 - x_1))(1 - (0 - x_2)) \\
& + 0 * (1 - (1 - x_1))(1 - (1 - x_2)) \\
\\
= {}& (1 - x_1)x_2 + x_1(1 - x_2) \\
\\
= {}& x1 + x2
\end{aligned}
$$

This is a polynomial in $\mathbb{F}_2^2$.

How to Find Fixed Points

Consider the single fixed point:

To determine a fixed point we solve

$$f_1 = x_1$$
$$f_2 = x_2$$
$$\vdots$$
$$f_n = x_n$$

Groebner Bases

Key Idea: We want to simplify the analysis of finding fixed points by computing a bases for the polynomials $f_1 - x_1, \ldots, f_n - x_n$

- Let f, g be polynomials in one variable. Then we can write $f = qg + r$ where q and r are unique polynomials.
- This means there is an algorithm for finding the remainder, r.
- But in general the remainder is not unique in division with polynomials in more than one variable.

Groebner bases allow for a unique remainder which means we can compute them by computer algebra system.

How Groebner Bases Simplify the Analysis

- Groebner bases are functions in triangular form where often the first equation is only in one variable. This means you can solve the system by back substitution.

- As an example consider the functions:

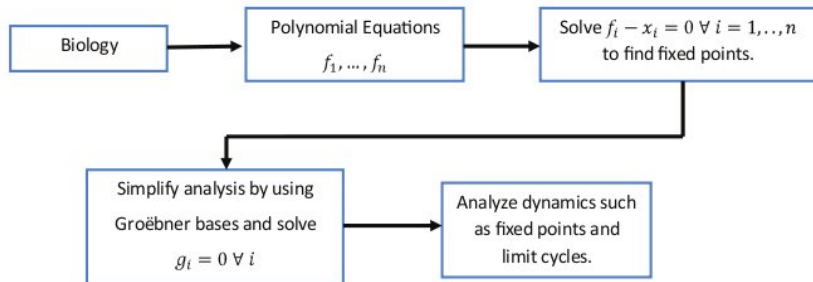$$f_1 = xy + x + y + 1$$
$$f_2 = xy + x + 1$$

- The Groebner basis is:

$$g_1 = y$$
$$g_2 = x + 1$$

which is easily solvable by back substitution.
$y = 0$ and $x = 1$.

Recap

Biology → Polynomial Equations $f_1, \dots, f_n$ → Solve $f_i - x_i = 0 \; \forall \; i = 1, .., n$ to find fixed points. → Simplify analysis by using Groëbner bases and solve $g_i = 0 \; \forall \; i$ → Analyze dynamics such as fixed points and limit cycles.

Goals - What We Hope to Accomplish

Over the next eight weeks we will

- Develop mathematical algorithms to analyze dynamics of large biological networks
- Write a software package using Macaulay2
- Incorporate our software with exisiting programs
  - GINsim
  - Polynome
  - Petrinet software

After the software is written

Verify the usefulness of the software

- ▶ Build a boolean model given a biological system
- ▶ Use the software we will write to analyze the dynamics of the system
  - ▶ Find the longest limit cycle
  - ▶ Calculate the number of components
  - ▶ Find all fixed points

Conclusion

- As we can express biological systems as polynomial functions we can implement abstract algebra to provide us with mathematical tools
- Developing the software is necessary to analyze large biological systems
- To develop software we will need mathematical algorithms that allow one to analyze large biological systems in a systematic way
- Software will be user friendly with easy to use interface
- Any biologist or other professional will be able to input the functions of a model that he or she wishes to know the key dynamics of