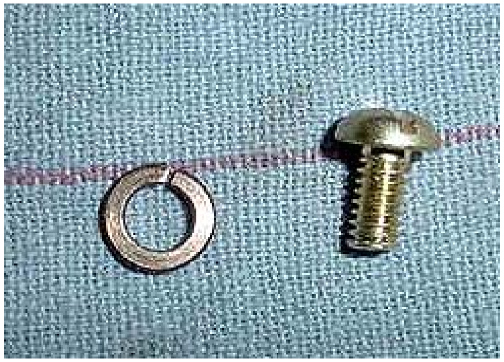
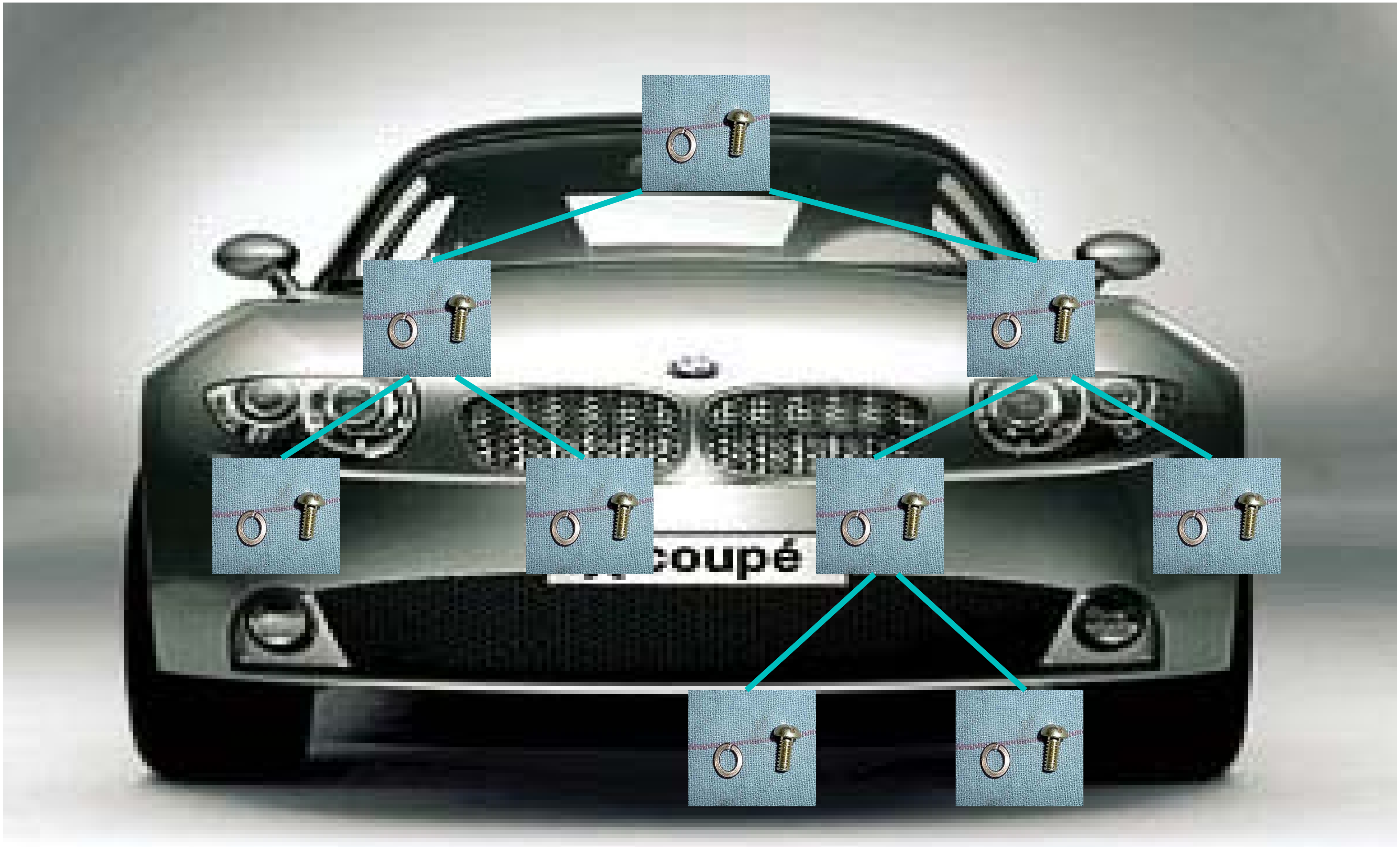
Biological Networks

Jehoshua (Shuki) Bruck

From Screws to Systems…

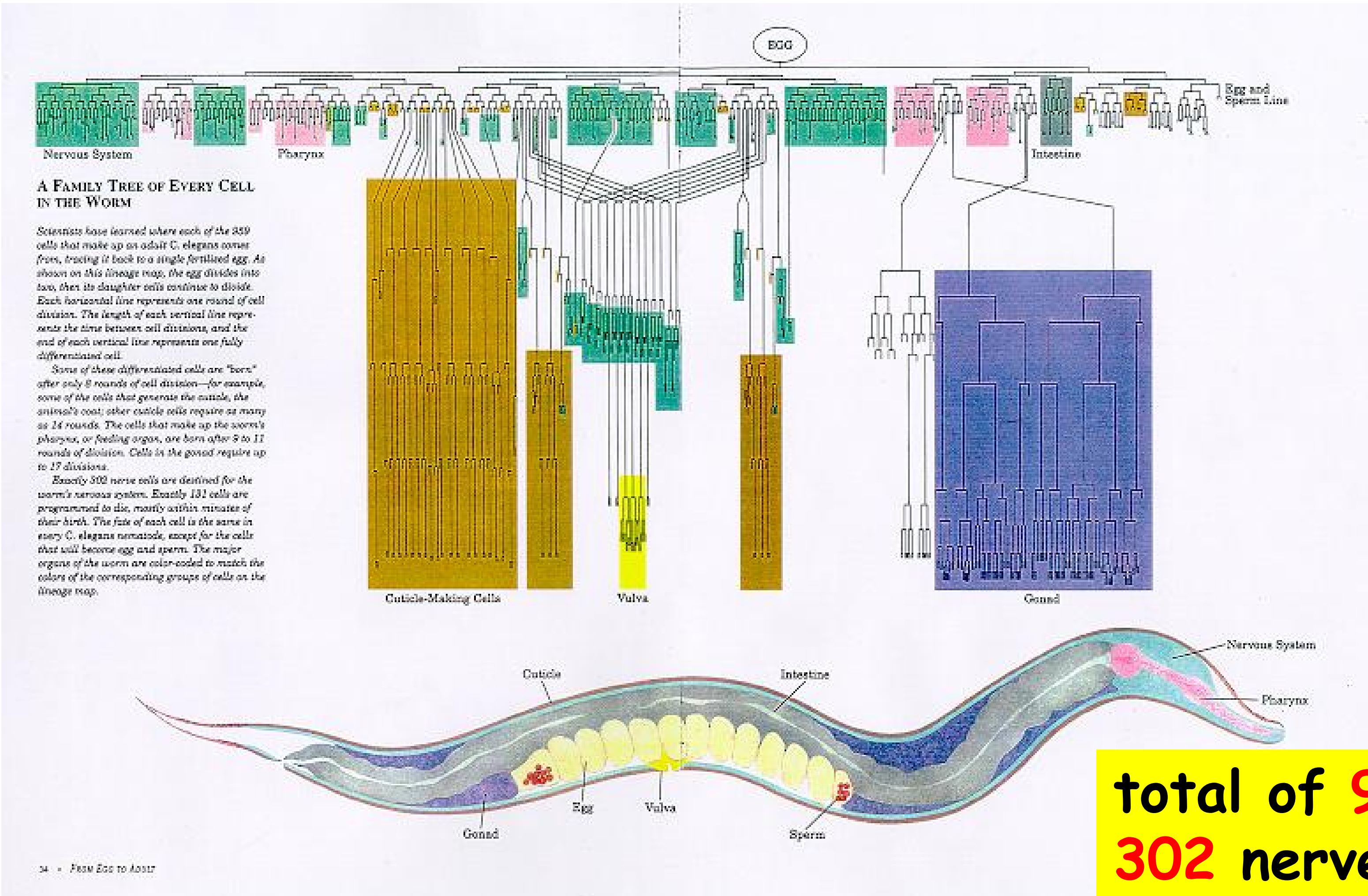


The Lineage of BMW



It happens in biological systems!!!

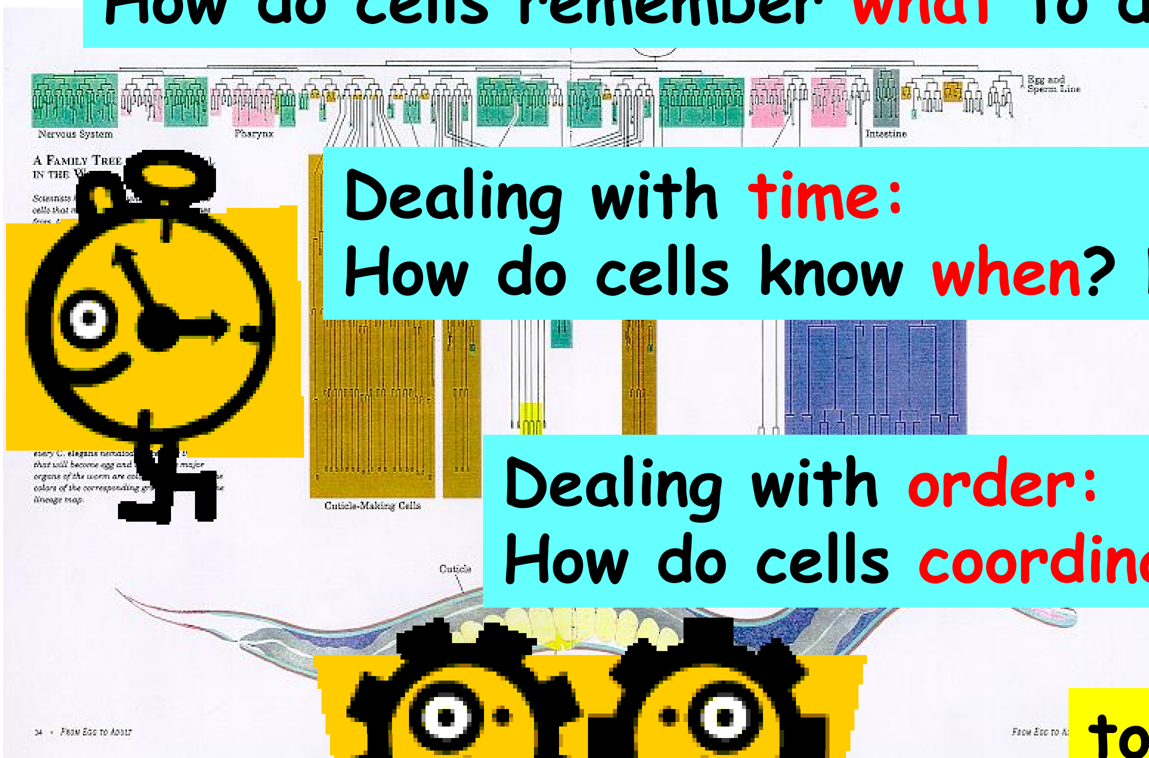
C. Elegans Lineage



total of 959 cells
302 nerve cells
131 cells are destined to die

C. Elegans Lineage - Simple Questions

Dealing with **identity**:
How do cells remember **what** to do?



Dealing with **time**:
How do cells know **when**? No clock…

Dealing with **order**:
How do cells **coordinate** their actions?

total of **959** cells
302 nerve cells
131 cells are destined to die

Control via Stochastic Chemical Reactions

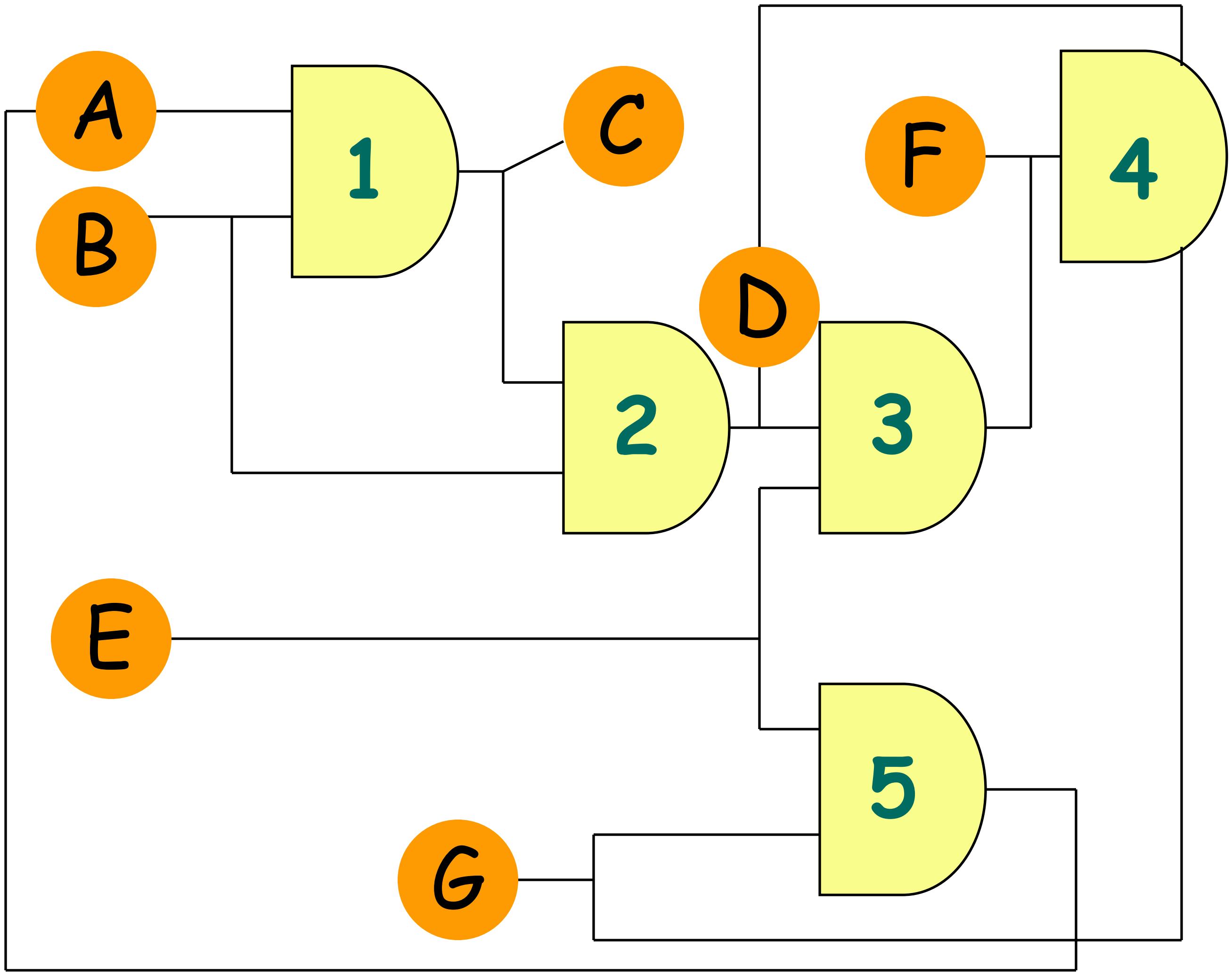
$$A + B \xrightarrow{k_1} C$$

$$B + C \xrightarrow{k_2} D$$

$$D + E \xrightarrow{k_3} F$$

$$F \xrightarrow{k_4} D + G$$

$$E + G \xrightarrow{k_5} A$$



Chemical Reactions Networks

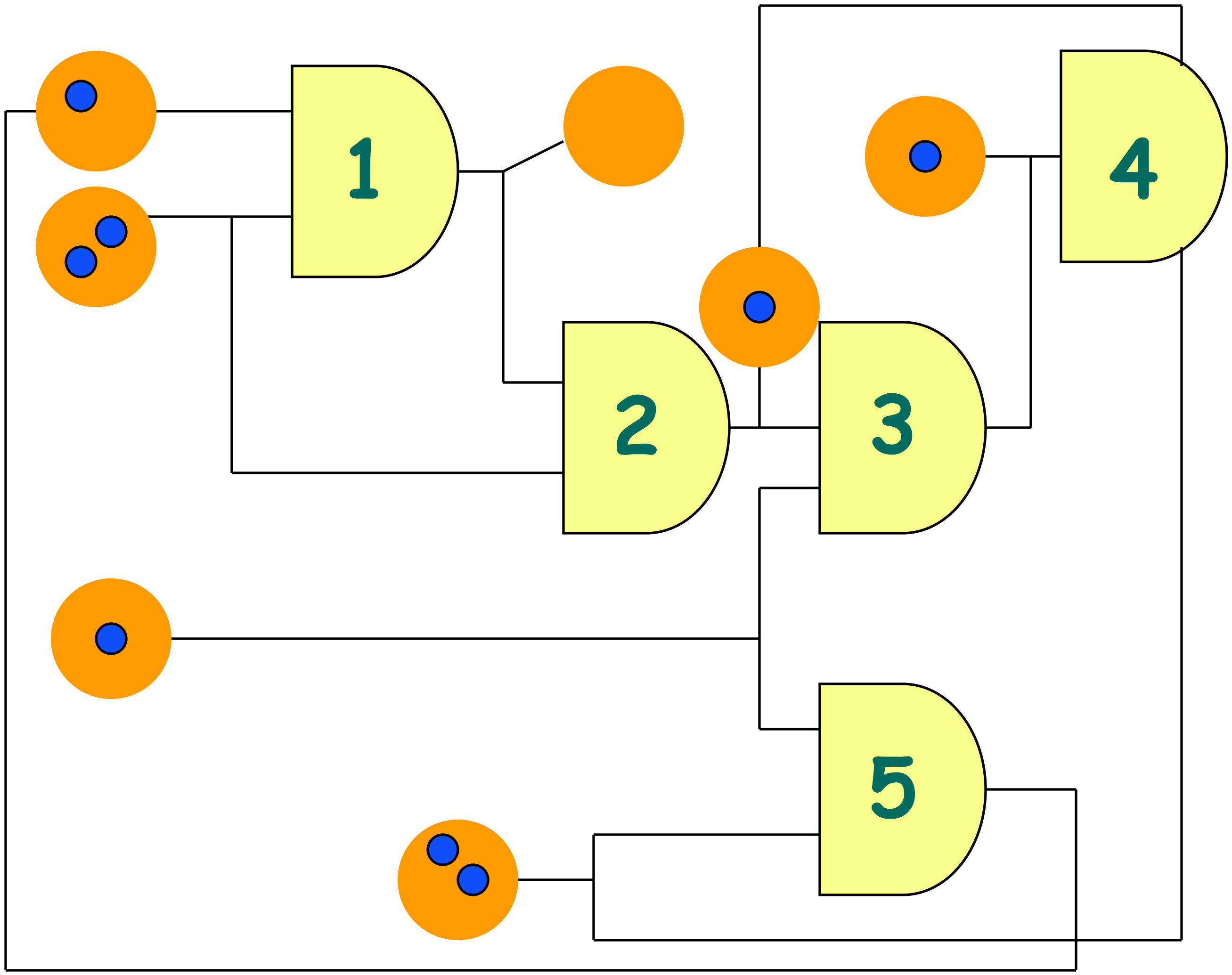
$$A + B \xrightarrow{k_1} C$$

$$B + C \xrightarrow{k_2} D$$

$$D + E \xrightarrow{k_3} F$$

$$F \xrightarrow{k_4} D + G$$

$$E + G \xrightarrow{k_5} A$$



Chemical Reactions Networks

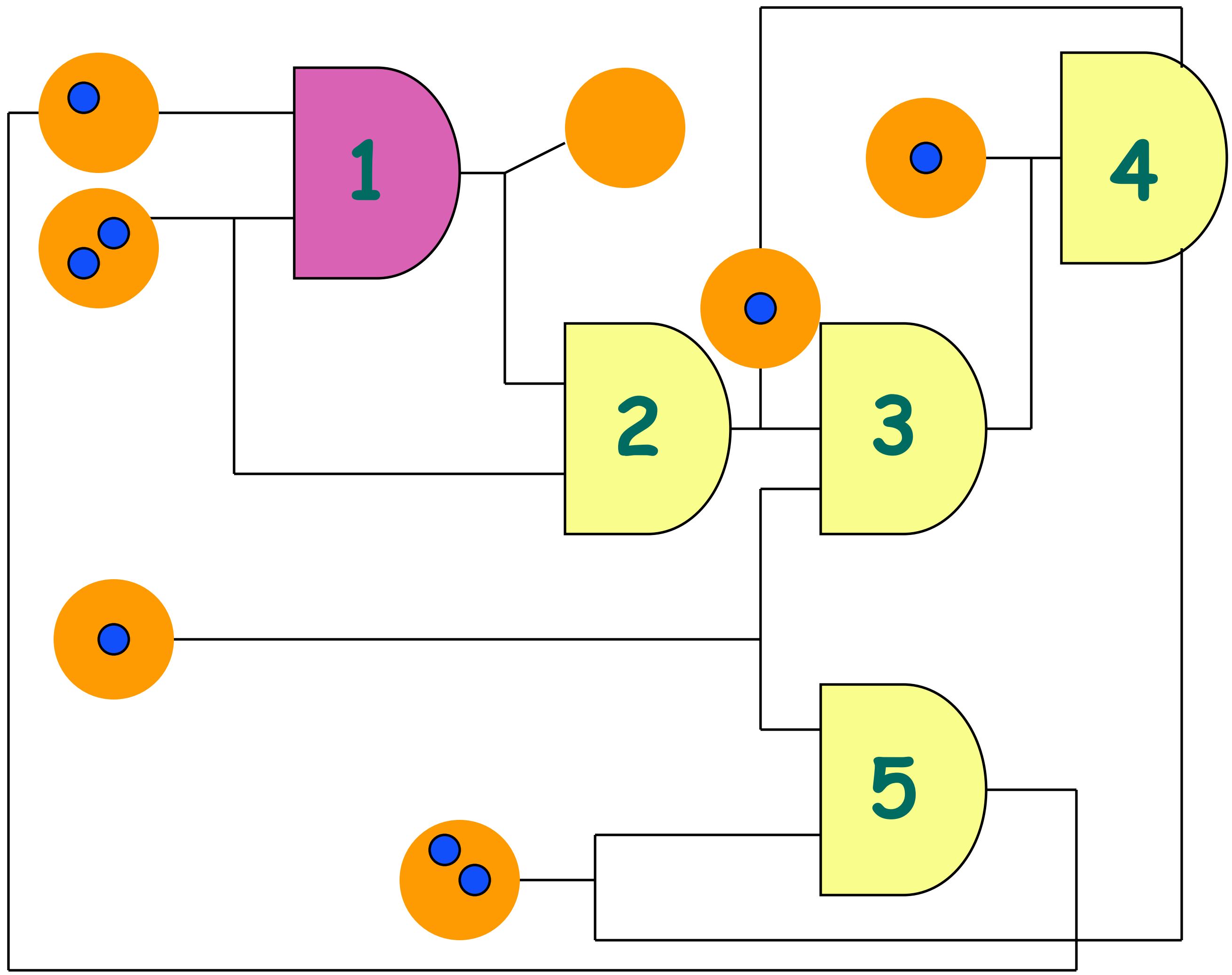
$$A + B \xrightarrow{k_1} C$$

$$B + C \xrightarrow{k_2} D$$

$$D + E \xrightarrow{k_3} F$$

$$F \xrightarrow{k_4} D + G$$

$$E + G \xrightarrow{k_5} A$$



Chemical Reactions Networks

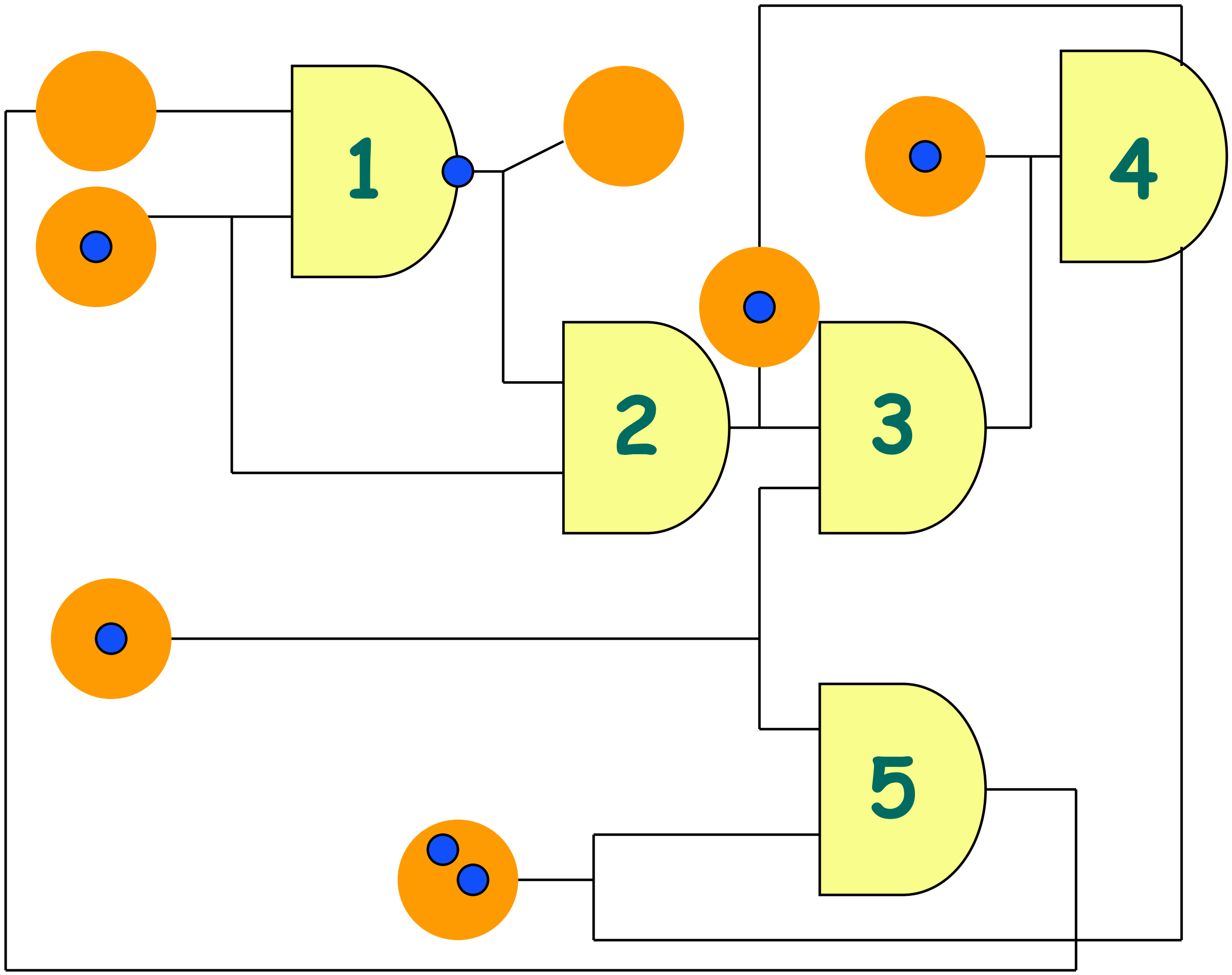
$$A + B \xrightarrow{k_1} C$$

$$B + C \xrightarrow{k_2} D$$

$$D + E \xrightarrow{k_3} F$$

$$F \xrightarrow{k_4} D + G$$

$$E + G \xrightarrow{k_5} A$$



Chemical Reactions Networks

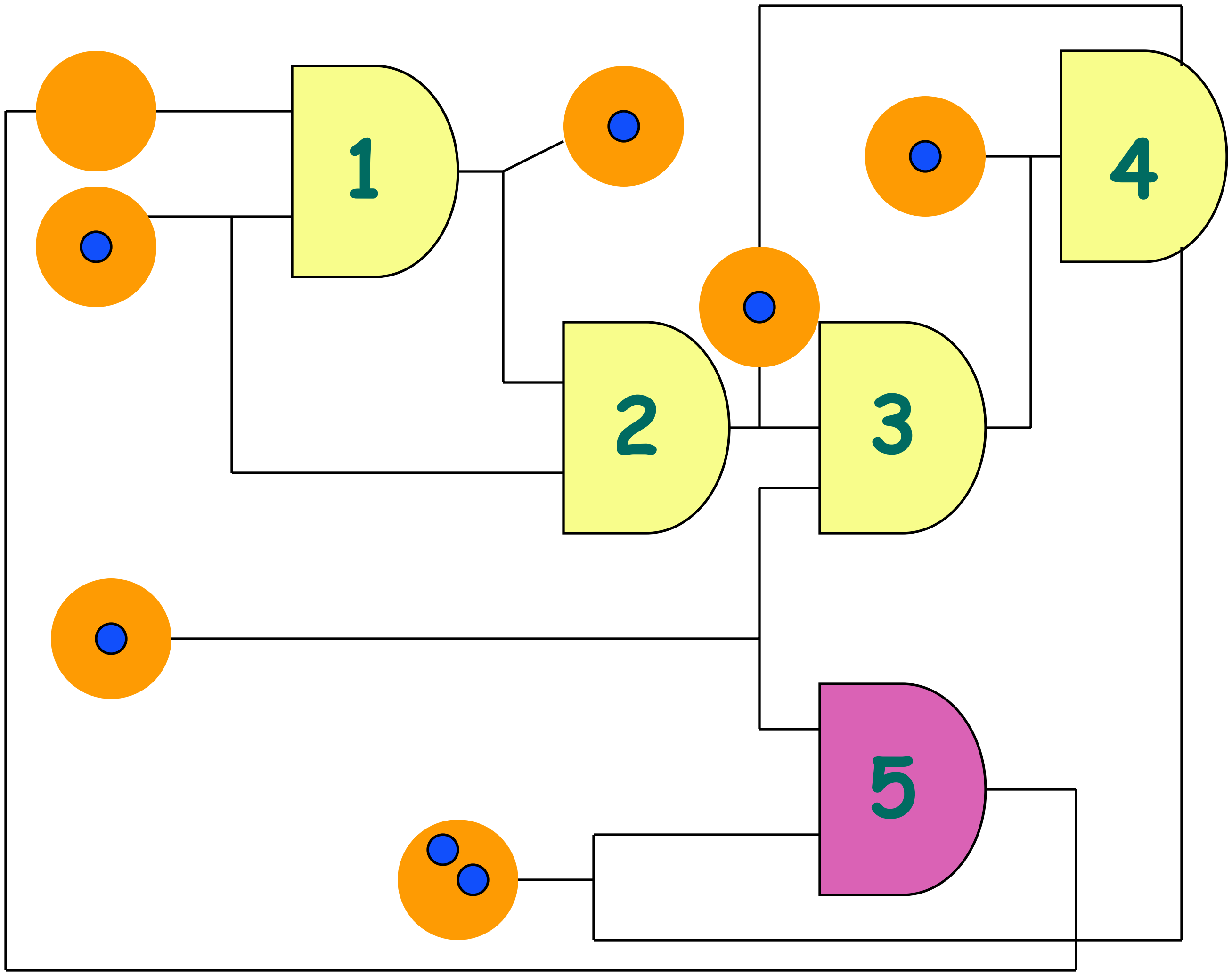
$$A + B \xrightarrow{k_1} C$$

$$B + C \xrightarrow{k_2} D$$

$$D + E \xrightarrow{k_3} F$$

$$F \xrightarrow{k_4} D + G$$

$$E + G \xrightarrow{k_5} A$$



Chemical Reactions Networks

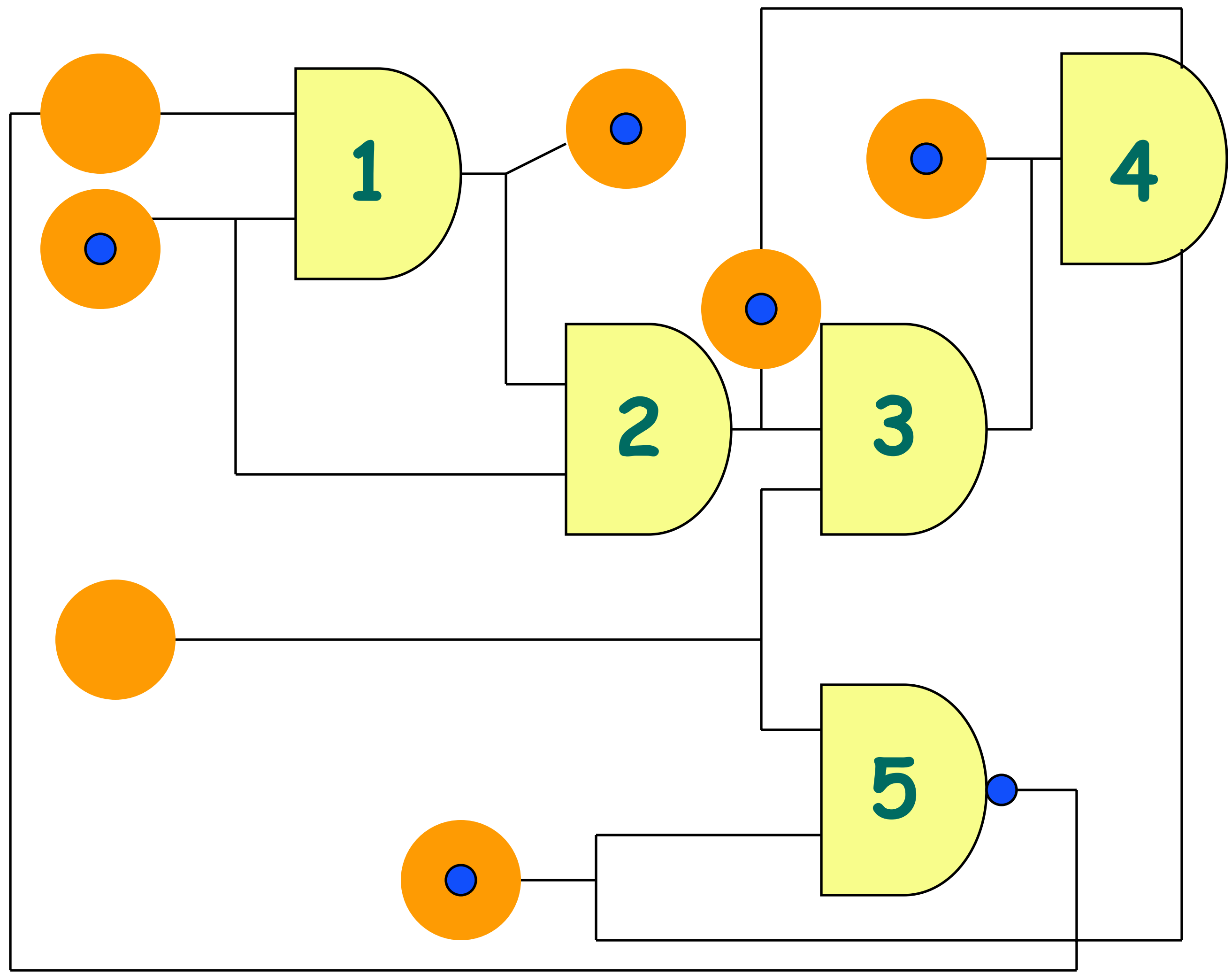
$$A + B \xrightarrow{k_1} C$$

$$B + C \xrightarrow{k_2} D$$

$$D + E \xrightarrow{k_3} F$$

$$F \xrightarrow{k_4} D + G$$

$$E + G \xrightarrow{k_5} A$$



Chemical Reactions Networks

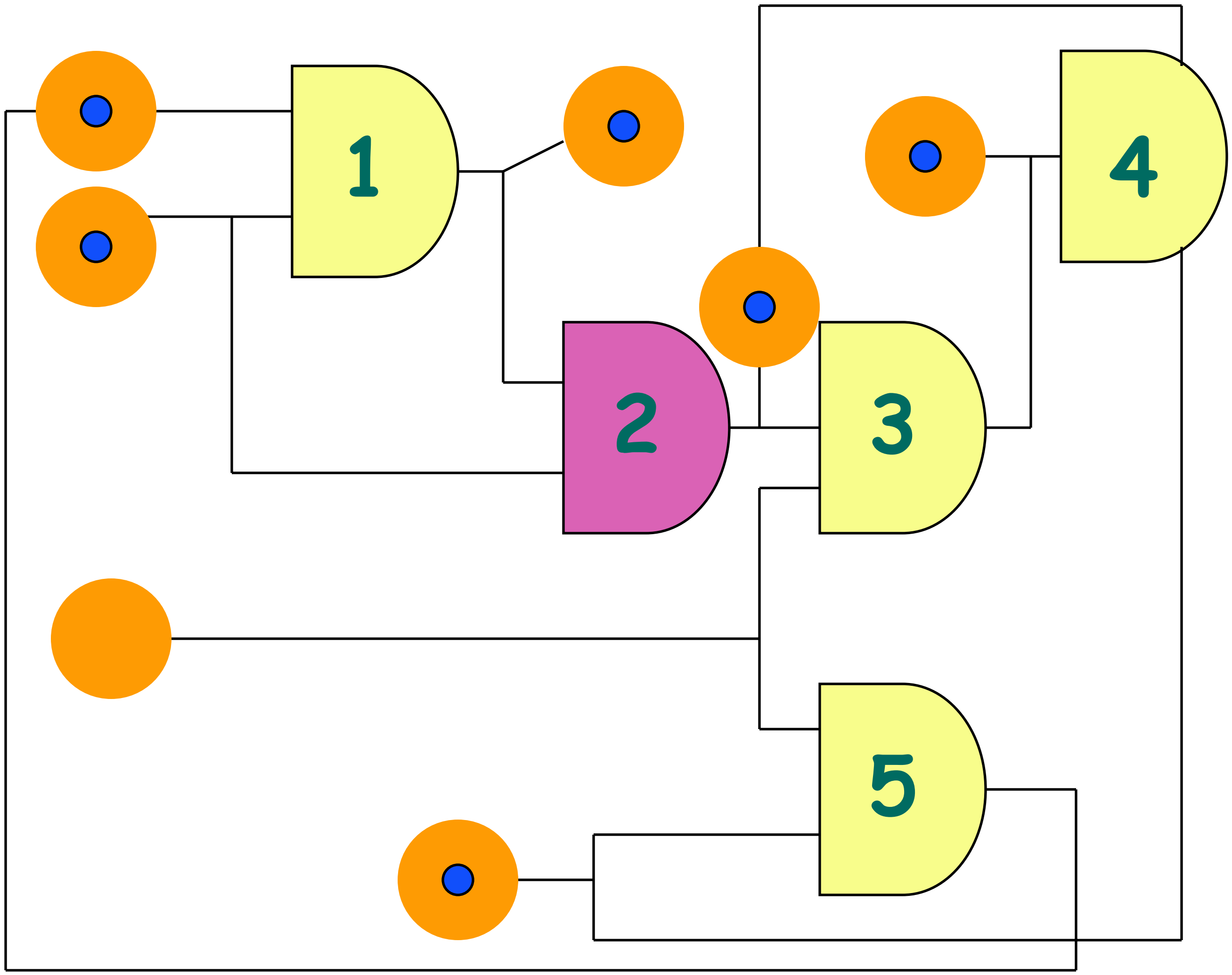
$$A + B \xrightarrow{k_1} C$$

$$B + C \xrightarrow{k_2} D$$

$$D + E \xrightarrow{k_3} F$$

$$F \xrightarrow{k_4} D + G$$

$$E + G \xrightarrow{k_5} A$$



Chemical Reactions Networks

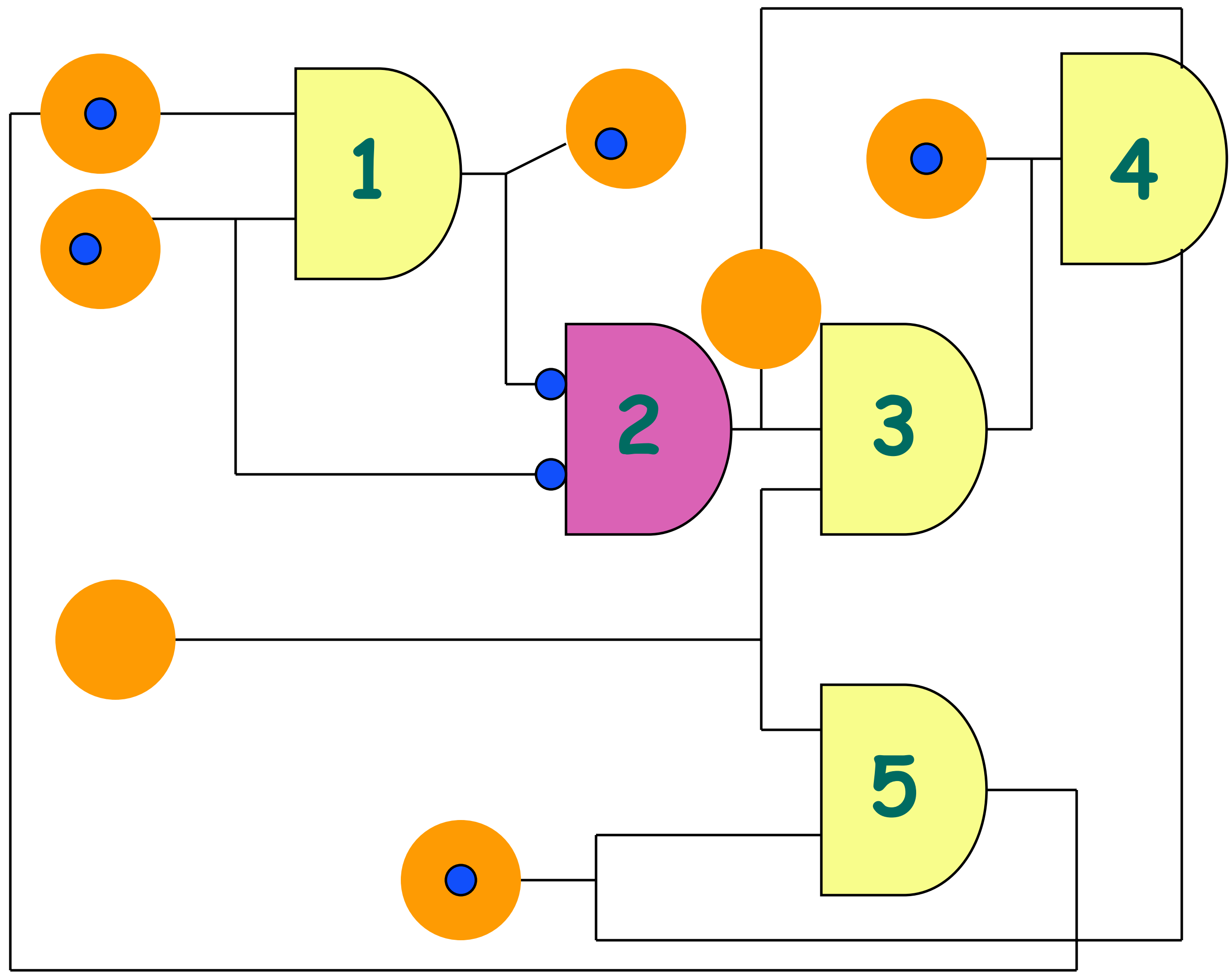
$$A + B \xrightarrow{k_1} C$$

$$B + C \xrightarrow{k_2} D$$

$$D + E \xrightarrow{k_3} F$$

$$F \xrightarrow{k_4} D + G$$

$$E + G \xrightarrow{k_5} A$$

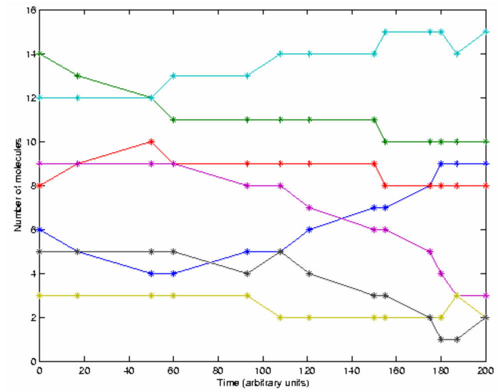


Solving the Puzzle

Mapping and Prediction

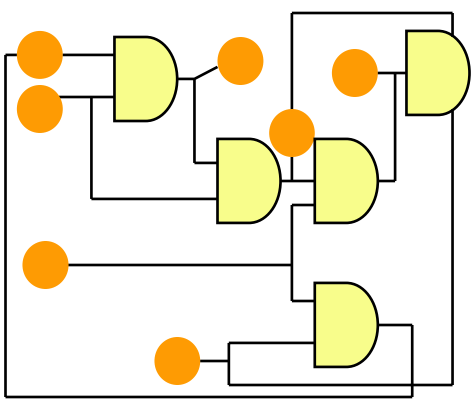
- What are the key players in in a gene regulatory system?
- What are their relevant interactions?
- **Success:** predictive model

$$A + B \xrightarrow{k_1} C$$
$$B + C \xrightarrow{k_2} D$$
$$D + E \xrightarrow{k_3} F$$
$$F \xrightarrow{k_4} D + G$$
$$E + G \xrightarrow{k_5} A$$



Principles and Abstractions

- What are the key computational principles in gene regulations?
- A formal language for design and analysis
- **Success:** understanding / compression a calculus for Biology



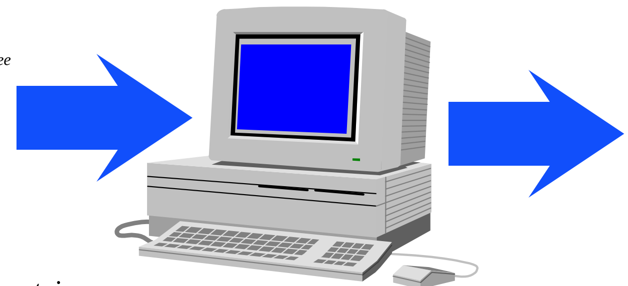
Mapping and Prediction

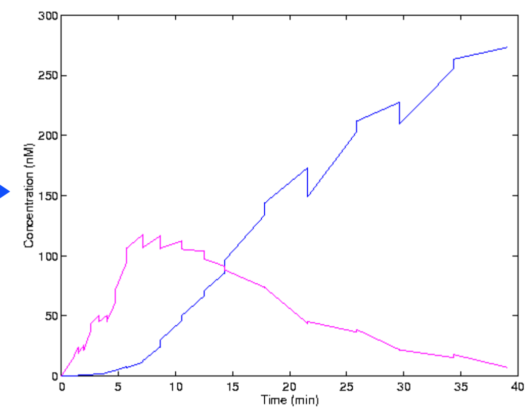
Gillespie, 1976; McAdams and Arkin, 1997
Gibson and Bruck, 2000; Riedel and Bruck 2005

Physical chemistry

Trajectories

$$RNAP*DNA_{closed} \xrightarrow{k_1 2} RNAP*DNA_{open,0}$$
$$RNAP*DNA_{open,n} \xrightarrow{k_2} RNAP*DNA_{open,n+1}$$
$$RNAP*DNA_{open,MAX} \xrightarrow{k_3} RNAP_{free} + DNA_{free} + mRNA_{free}$$
$$Ribosome + mRNA_{free} \xrightarrow{k_4} Ribosome*mRNA_0$$
$$RNase + mRNA_{free} \xrightarrow{k_5} RNase*mRNA$$
$$RNase*mRNA \xrightarrow{k_6} RNase$$
$$Ribosome*mRNA_n \xrightarrow{k_7} Ribosome*mRNA_{n+1}$$
$$Ribosome*mRNA_{MAX} \xrightarrow{k_8} Ribosome_{free} + mRNA_{free} + protein$$
$$protein \xrightarrow{k_9} no\_protein$$
$$Volume = Initial\_volume(1+k_{10}t)$$

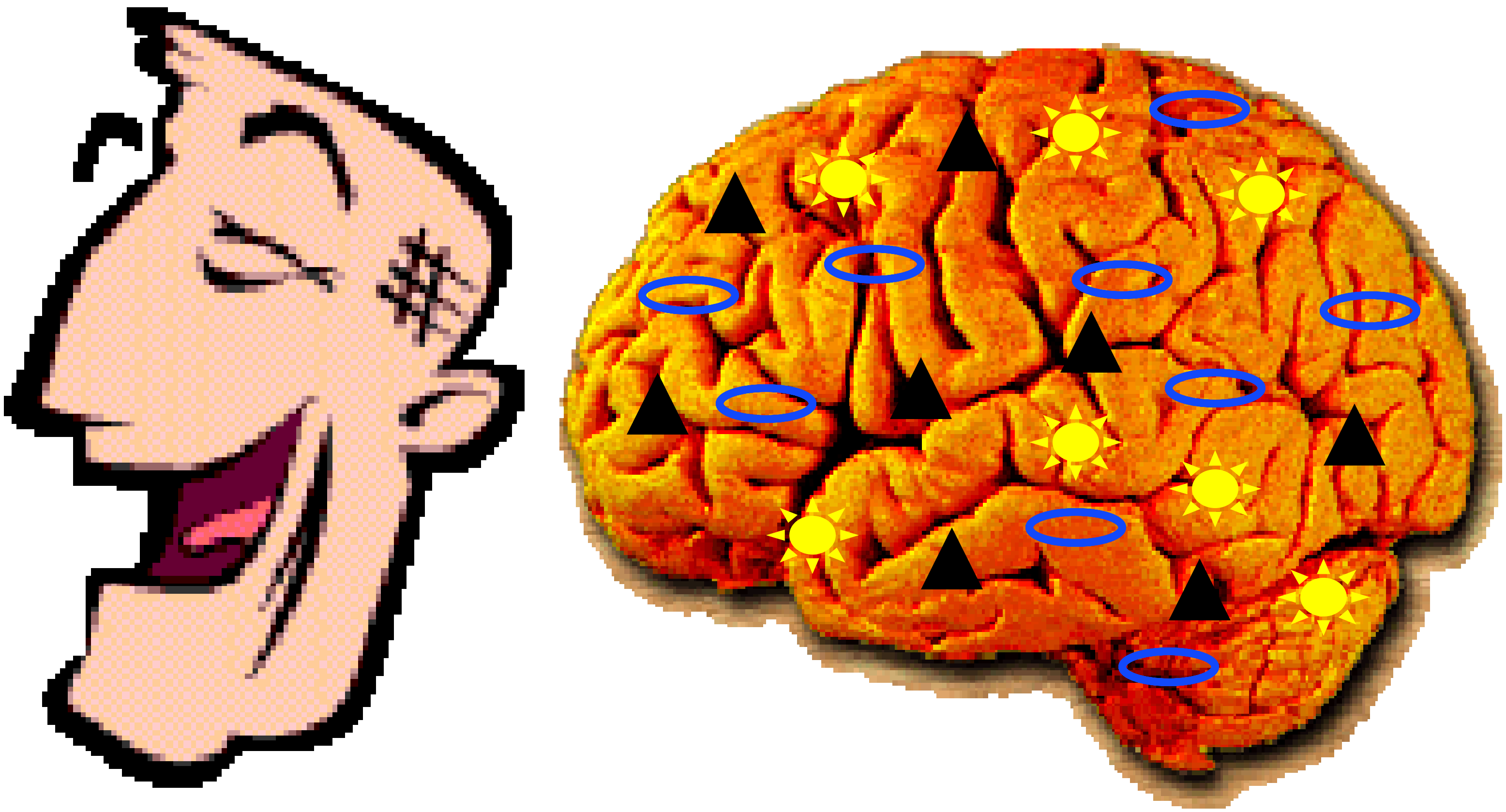




Generating trajectories from stochastic chemical equations

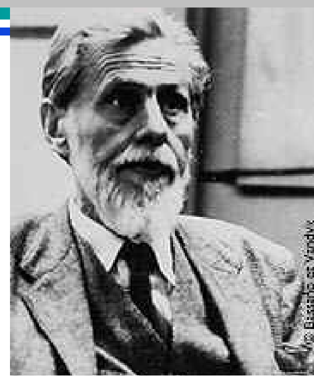
We can “see” trajectories and know how compute them faster

Descriptive Biology: Is It Sufficient?



Early Work on Abstractions

Neurophysiologist, MD



Logician, Autodidact

Warren McCulloch arrived in early 1942 to the University of Chicago, invited Pitts, who was still homeless, to live with his family.

In the evenings McCulloch and Pitts collaborated. Pitts was familiar **with the work of Gottfried Leibniz on computing** and they considered the question of whether the nervous system could be considered a kind of **universal computing device as described by Leibniz.**

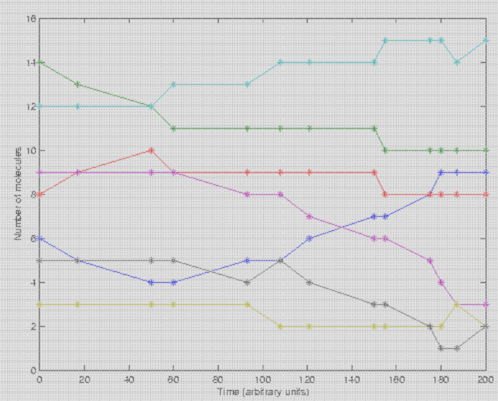
This led to their 1943 seminal neural networks paper:
A Logical Calculus of Ideas Immanent in Nervous Activity.

Solving the Biology Puzzle

Mapping and Prediction

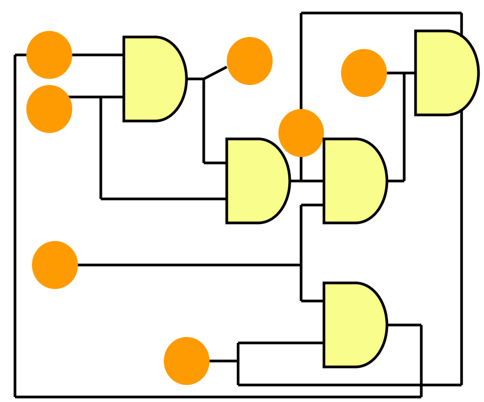
- What are the key players in in a gene regulatory system?
- What are their relevant interactions?
- Success: predictive model

$$A + B \xrightarrow{k_1} C$$
$$B + C \xrightarrow{k_2} D$$
$$D + E \xrightarrow{k_3} F$$
$$F \xrightarrow{k_4} D + G$$
$$E + G \xrightarrow{k_5} A$$



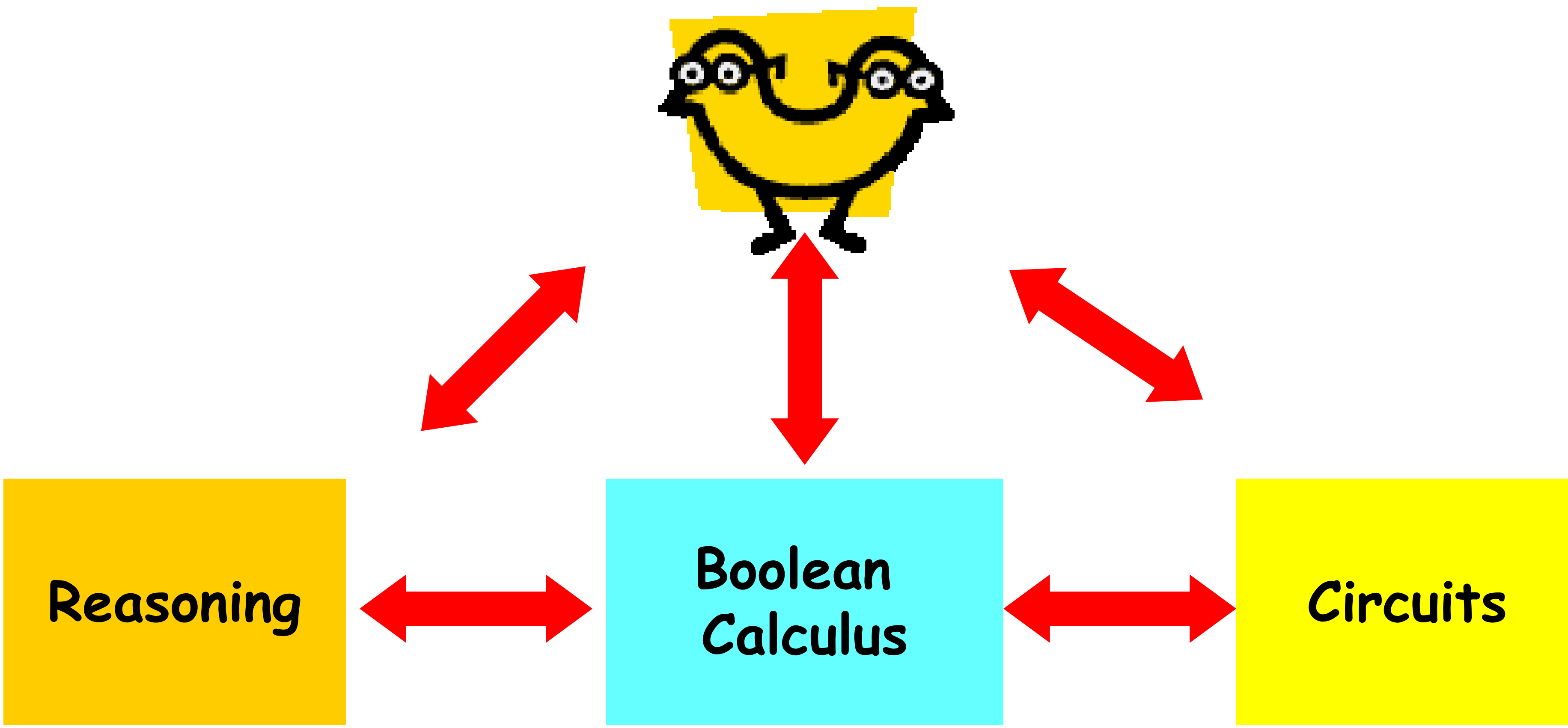
Principles and Abstractions

- What are the key computational principles in gene regulations?
- A formal language for design and analysis
- Success:
  understanding / compression
  a calculus for Biology



Key to the Wonderful Progress in Design:
Abstractions in Information Systems

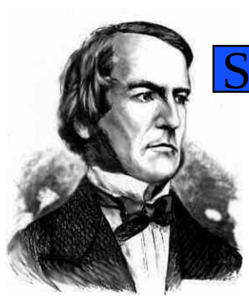
Reasoning to Calculations to Physics

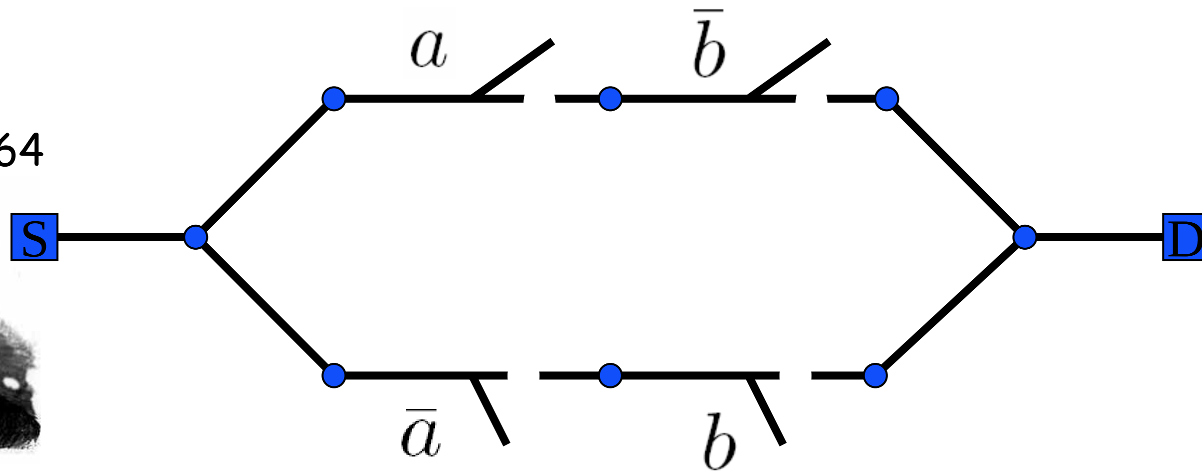


Key to the Progress in Design:
Abstractions in Information Systems

Logic to Boolean Calculus to Physical Circuits

Boole
1815-1864



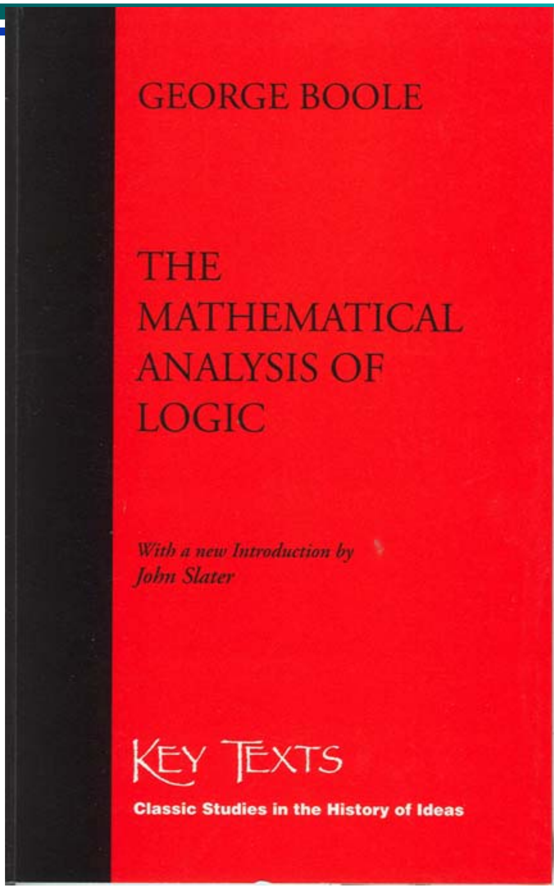


Shannon
1916-2001

1847
Connected Logic
with Algebra
Boolean Algebra
Logical Calculation

1938
Boolean Algebra to
Electrical Circuits
Logic Design

Text to Algebra
George Boole, 1854



the interpretation of v. But it has been thought better to write them separately, for greater ease and convenience. And it is further to be borne in mind, that although three different forms are given for the expression of each of the *particular* propositions, everything is really included in the first form.

TABLE.

The class X	x	
The class not-X	$1 - x$	
All Xs are Ys All Ys are Xs	$x = y$	
All Xs are Ys	$x(1 - y) = 0$	
No Xs are Ys	$xy = 0$	
All Ys are Xs Some Xs are Ys	$y = vx$	vx = some Xs $v(1 - x) = 0$.
No Ys are Xs Some not-Xs are Ys	$y = v(1 - x)$	$v(1 - x)$ = some not-Xs $vx = 0$.
Some Xs are Ys	$v = xy$ or $vx = vy$ or $vx(1 - y) = 0$	v = some Xs or some Ys vx = some Xs, vy = some Ys $v(1 - x) = 0$, $v(1 - y) = 0$.
Some Xs are not Ys	$v = x(1 - y)$ or $vx = v(1 - y)$ or $vxy = 0$	v = some Xs, or some not-Ys vx = some Xs, $v(1 - y)$ = some not-Ys $v(1 - x) = 0$, $vy = 0$.

The Algebra (Boolean Calculus)

Boole, DeMorgan, Jevons, Peirce, Schroder (18xx)

Postulate System: Huntington (1904)

Algebraic system: set of elements B,
two binary operations + and ·
B has at least two elements (0 and 1)

If the following postulates are true
then it is a **Boolean Algebra**:

(i) identity $a + 0 = a; \quad a \cdot 1 = a$

(ii) complement $a + \overline{a} = 1; \quad a \cdot \overline{a} = 0$

(iii) commutative $a + b = b + a; \quad a \cdot b = b \cdot a$

(vi) distributive $a + b \cdot c = (a + b) \cdot (a + c); \quad a \cdot (b + c) = a \cdot b + a \cdot c$

Shannon MSc Thesis, 1938

c_{k+1}	c_k	$c_{j+1}c_j$	c_2c_1	Carried numbers
	a_k---$a_{j+1}a_j$---$a_2a_1a_0$			First number
	b_k	$b_{j+1}b_j$	$b_2b_1b_0$	Second number
c_{k+1}	s_k---$s_{j+1}s_j$---s_2,s_1s_0			Sum

...t not

Who invented the binary representation of numbers?

c_{j+1} is one if two or ...

$$c_{j+1} = S_{2,3}(a_j, b_j, c_j), \quad j = 1, 2, \ldots k .$$

Using the method of symmetric functions, and shifting down for s_j gives the circuits of Figure 35. Eliminating superfluous elements we arrive at Figure 36.

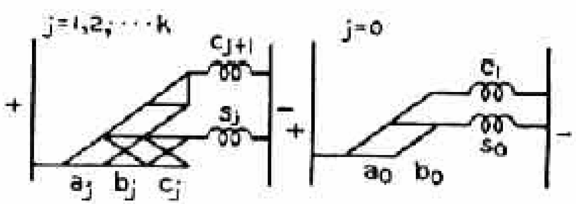


Figure 35. Circuits for electric adder

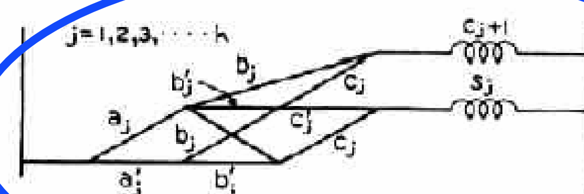
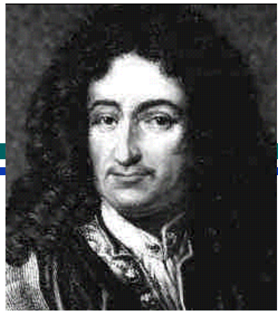


Figure 36. Simplification of figure 35

Gottfried Leibniz
1646-1716



Leibniz - Binary System

And accordingly, numbers are expressed as follows:

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

10000	16
10001	17
10010	18
10011	19
10100	20
10101	21
10110	22
10111	23
11000	24
11001	25
11010	26
11011	27
11100	28
11101	29
11110	30
11111	31
100000	32
etc.	etc

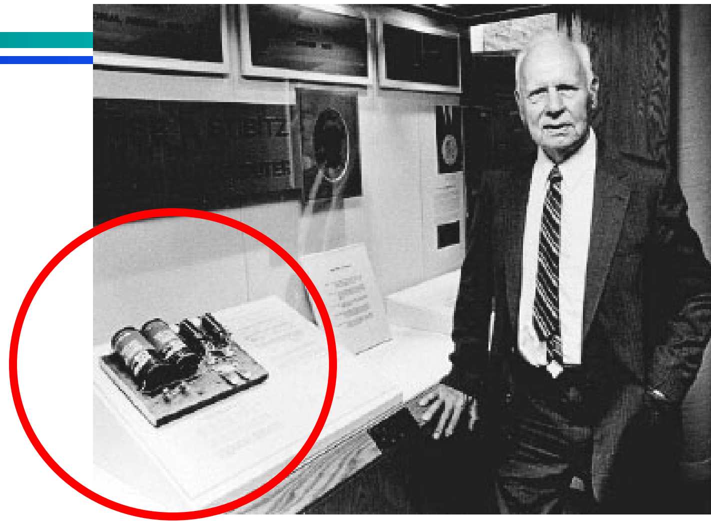
Gottfried Leibniz
1646-1716

Leibniz - Binary System

Binary addition algorithm

§72 As for addition, it is simply done by counting and making periods when there are numbers to add together, adding up each column as usual, which will be done as follows: count the unities of the column for example, for 29, look how this number is written in the table, to wit, by 11101; thus you write 1 under the column and put periods under the second, third and fourth column thereafter. These periods denote that it is necessary to count out one unity further in the column following.

The First Digital Adder
George Stibitz, 1904-1995



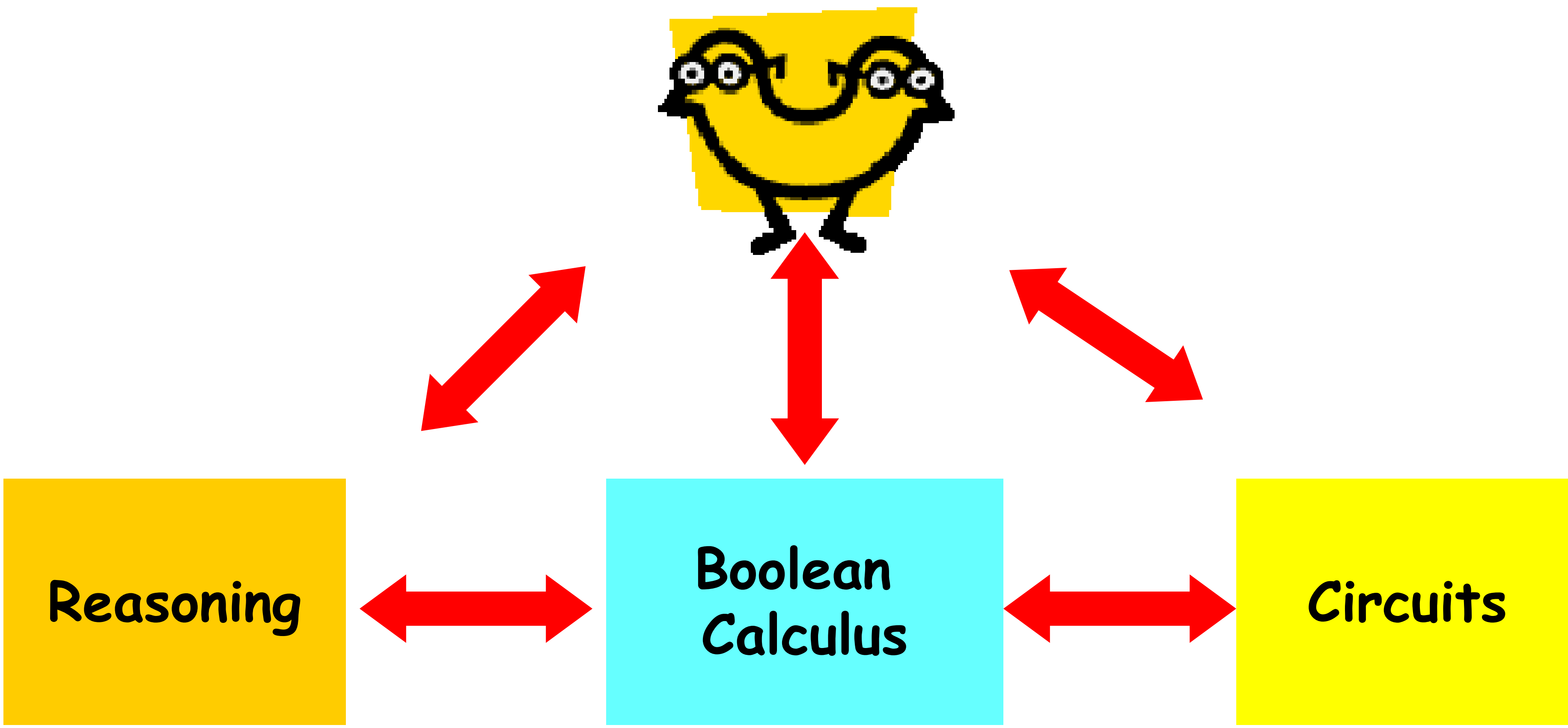
He worked at Bell Labs in New York.

In the fall of 1937 Stibitz used surplus relays, tin can strips, flashlight bulbs, and other common items to construct his "Model K" (K stands for kitchen table).

Model K was designed to display the result of the addition of two bits.

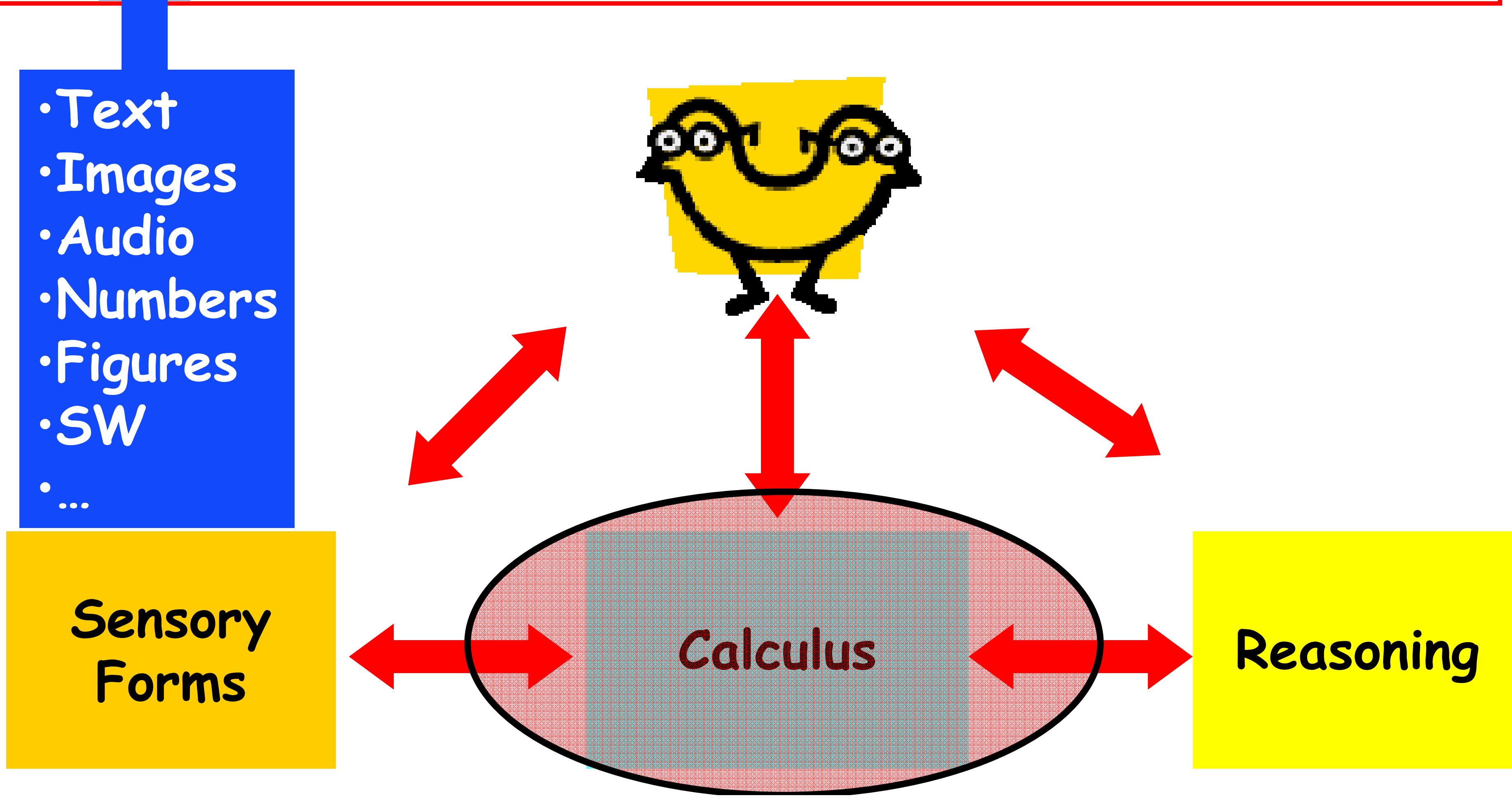
Key to the Wonderful Progress in Design: Abstractions in Information Systems

Reasoning to Calculations to Physics



Key Challenge to the Progress in Analysis Abstractions in Information Systems

Sensory Forms to Calculations to Reasoning

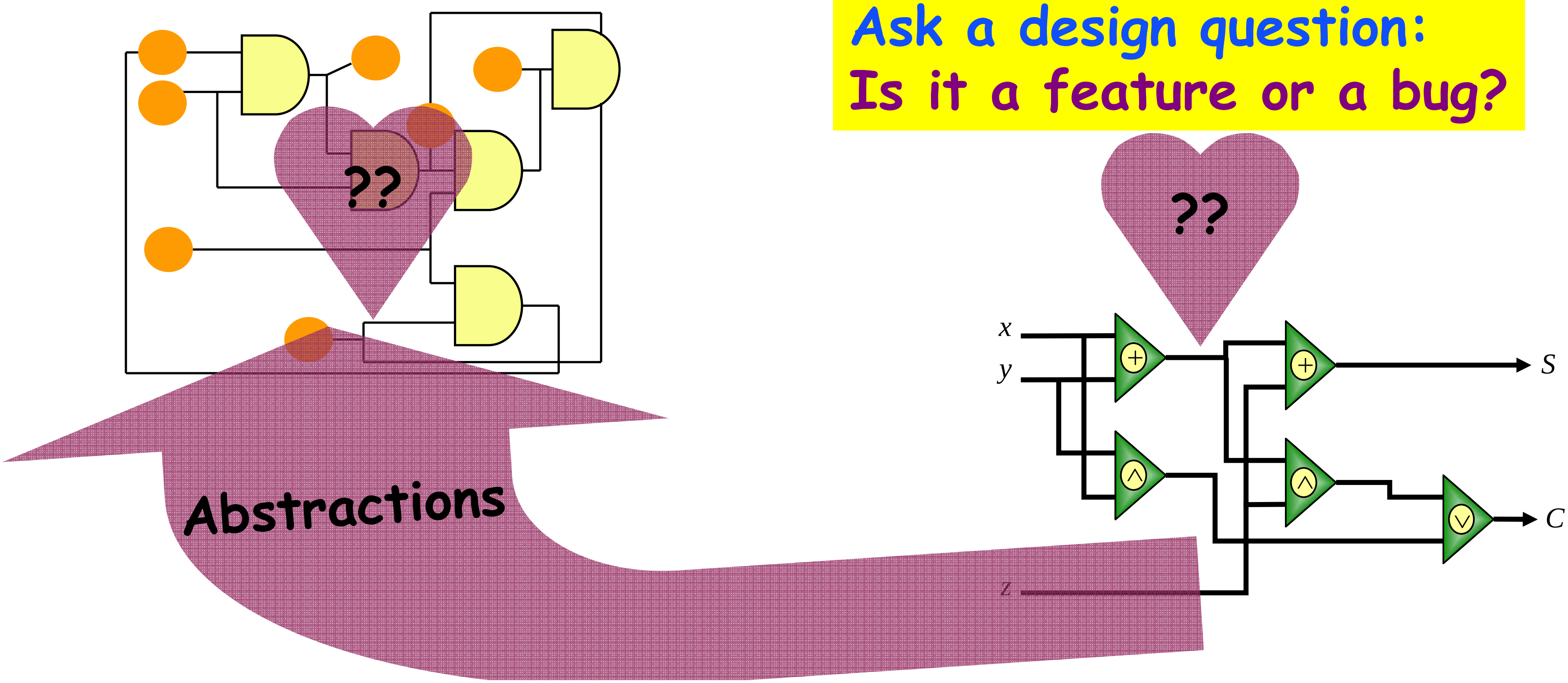


Key Challenge to the Progress in Analysis
Abstractions in Information Systems

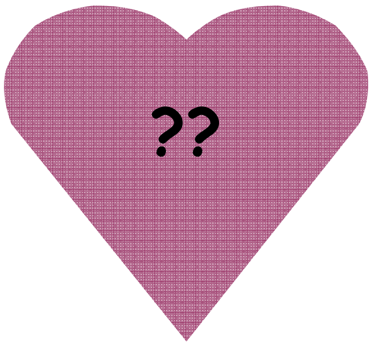
Sensory Forms to Calculations to Reasoning

Biology | **Engineering**

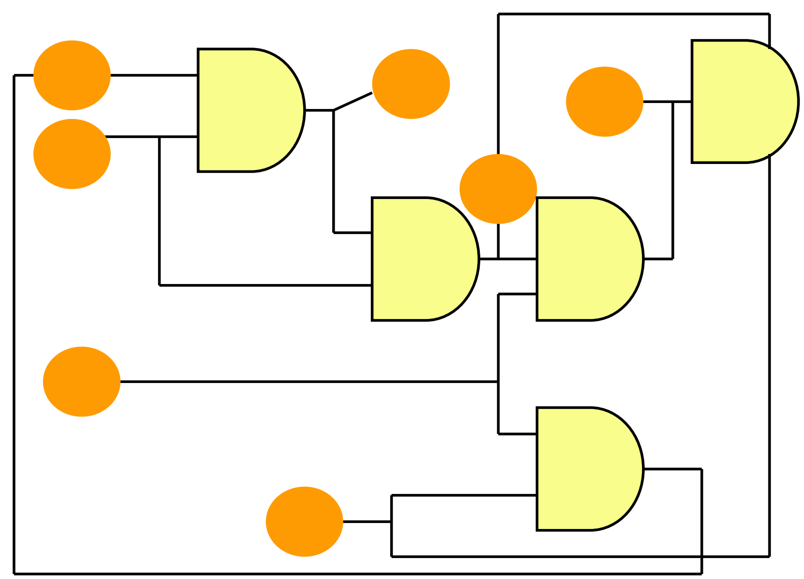
Ask a design question:
Is it a feature or a bug?



A Feature or a Bug?



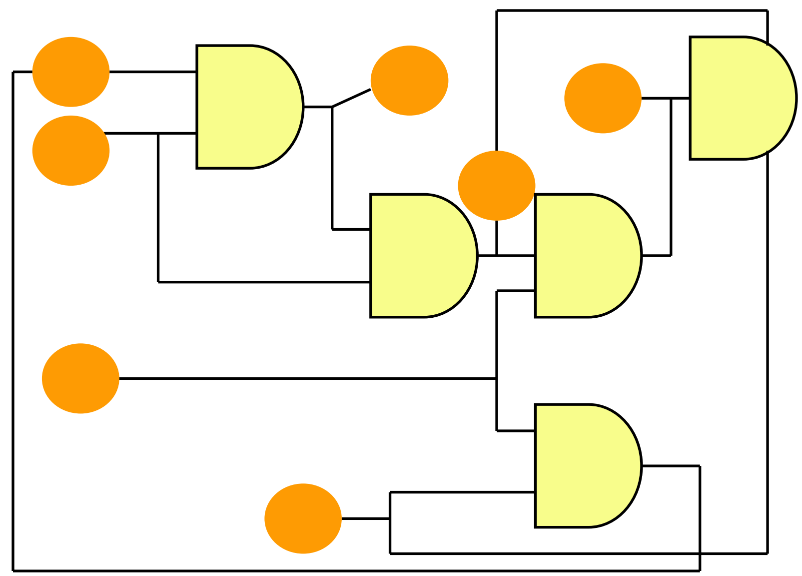
- Cyclic vs. acyclic (feedback)
- Stochastic vs. deterministic

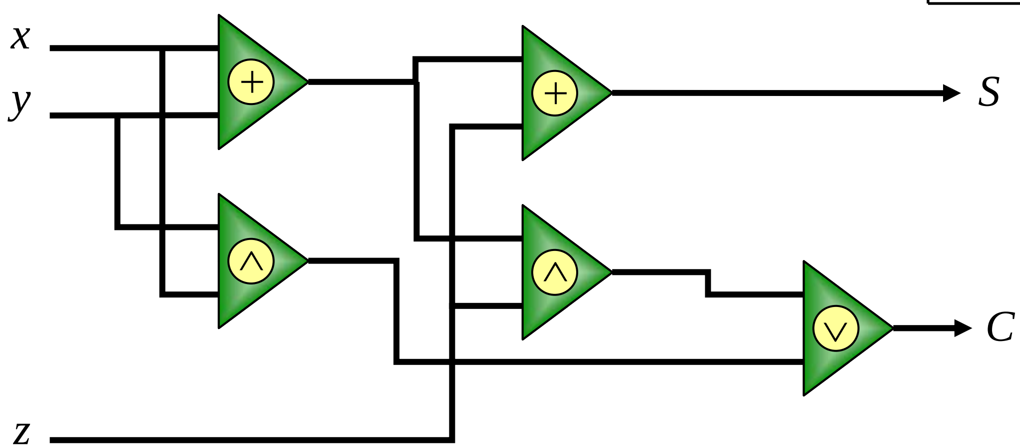


Bio Circuits vs. Combinational Logic Circuits

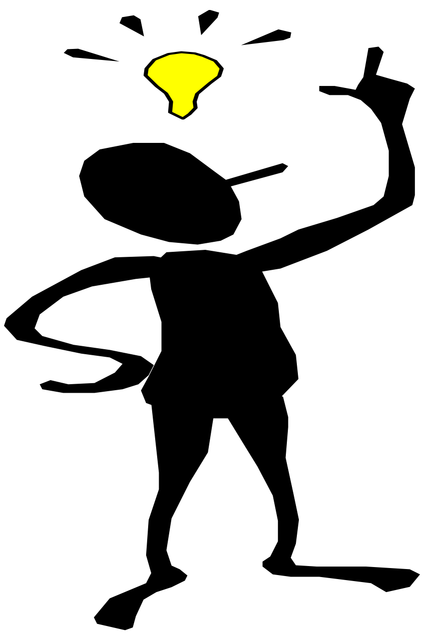
Joint work with Marc Riedel

- Cyclic vs. acyclic (feedback)
- Stochastic vs. deterministic





Are Cycles a Feature or a Bug?



Hypothesis ?????

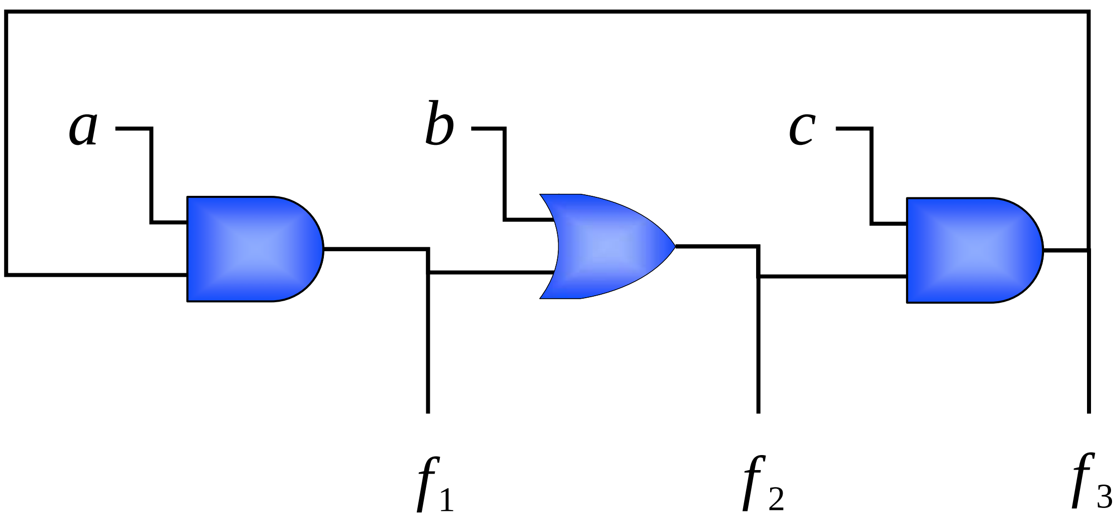
Cycles might help in

- Reducing cost
- Increasing performance

Circuits With Cycles

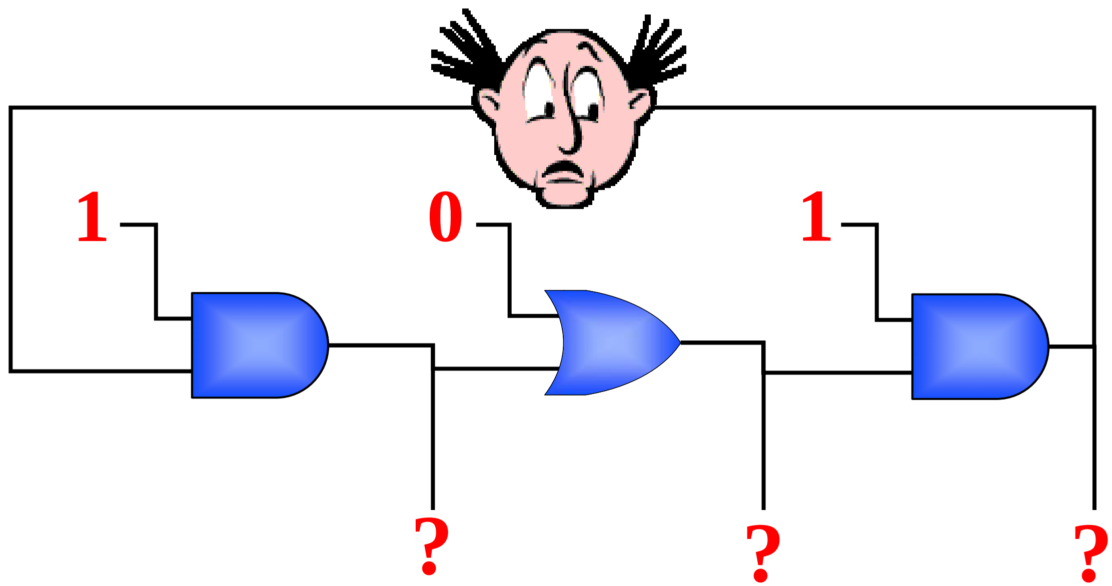
Generally exhibit **time-dependent** behavior

May have **unstable/unknown** outputs



Circuits With Cycles

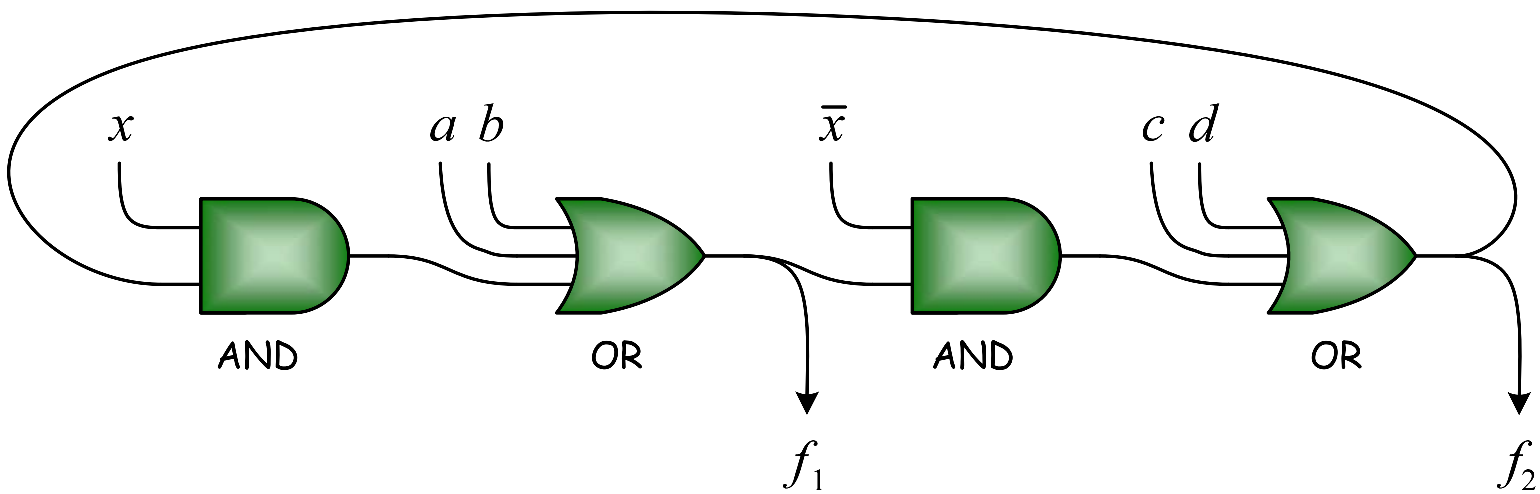
Generally exhibit **time-dependent** behavior
May have **unstable/unknown** outputs



0: non-controlling for OR
1: non-controlling for AND

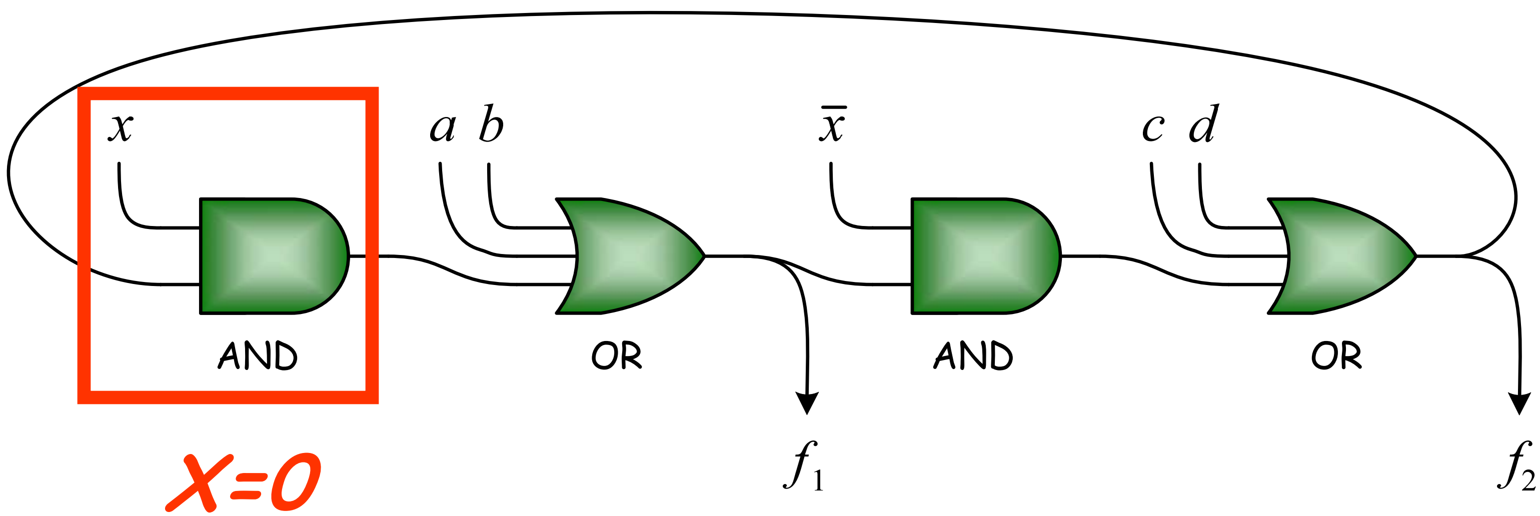
Cyclic Circuits Can be Combinational
McCaw's 1963

Cyclic, 4 AND/OR gates, 5 variables, 2 functions:



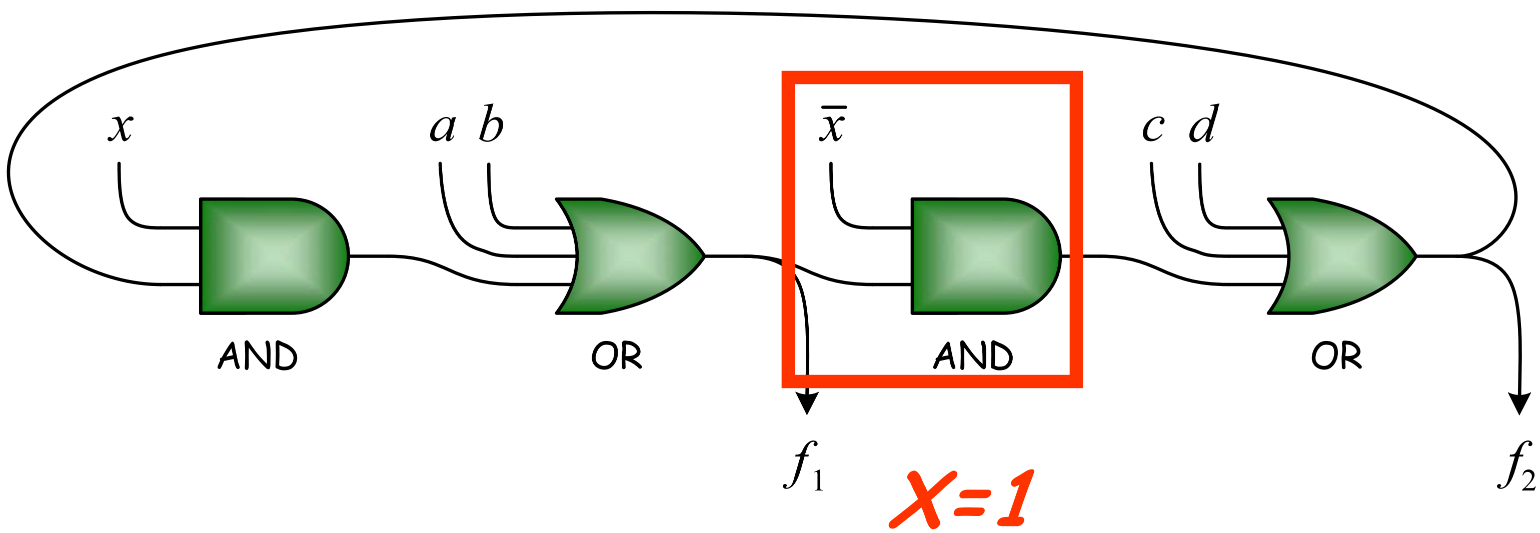
Cyclic Circuits Can be Combinational
McCaw's 1963

Cyclic, 4 AND/OR gates, 5 variables, 2 functions:



Cyclic Circuits Can be Combinational
McCaw's 1963

Cyclic, 4 AND/OR gates, 5 variables, 2 functions:



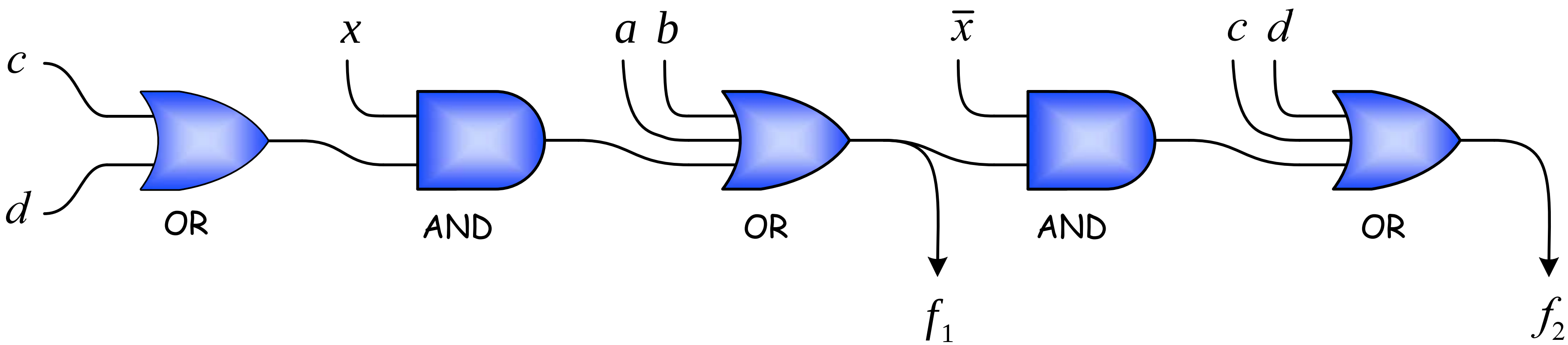
$$f_1 = a + b + x(c + d)$$

$$f_2 = c + d + \bar{x}(a + b)$$

McCaw's Circuit (1963)

Smallest possible equivalent acyclic circuit?

5 AND/OR gates; improvement factor is 4/5

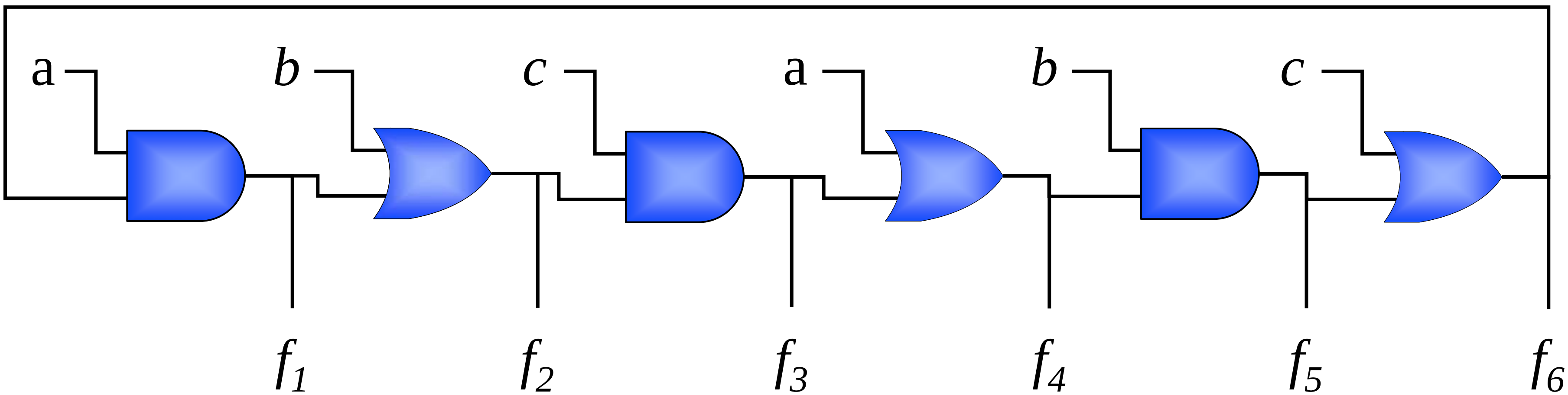


$$f_1 = a + b + x(c + d)$$

$$f_2 = c + d + \overline{x}(a + b)$$

Cyclic Combinational Circuits

Cyclic circuits can be combinational
Short 1961, McCaw 1963, Kautz 1970, Huffman 1971, Rivest 1977



Improvement factor is 2/3 (Rivest 1977)

Improvement factor of ½ (Riedel & Bruck 2003)

The Role of Cycles in Circuit Design?

Best paper award in 2003 Design Automation Conference

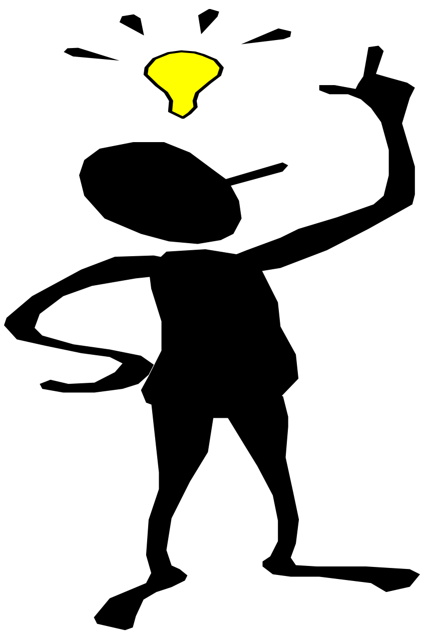
- Developed the theory and synthesis techniques for cyclic combinational circuits
  Synthesis is based on symbolic analysis

- Caltech Cyclify = a software package for the design of combinational circuits with cycles

- Integrated Caltech Cyclify with the Berkeley design tools

- Evaluated benchmark circuits and compared with current design tools

Cycles in Circuits is a Feature!



Cycles help in

- Reducing cost
- Increasing performance

Optimization for Cost (Area)

Cost: Number of NAND2/NOR2 gates

Benchmark	Berkeley SIS	Caltech CYCLIFY	Improvement
5xp1	203	182	10.34%
ex6	194	152	21.65%
planet	943	889	5.73%
s386	231	222	3.90%
bw	302	255	15.56%
cse	344	329	4.36%
pma	409	393	3.91%
s510	514	483	6.03%
duke2	847	673	20.54%
styr	858	758	11.66%
s1488	1084	1003	7.47%

Optimization for Performance (Delay) and Fixed Cost

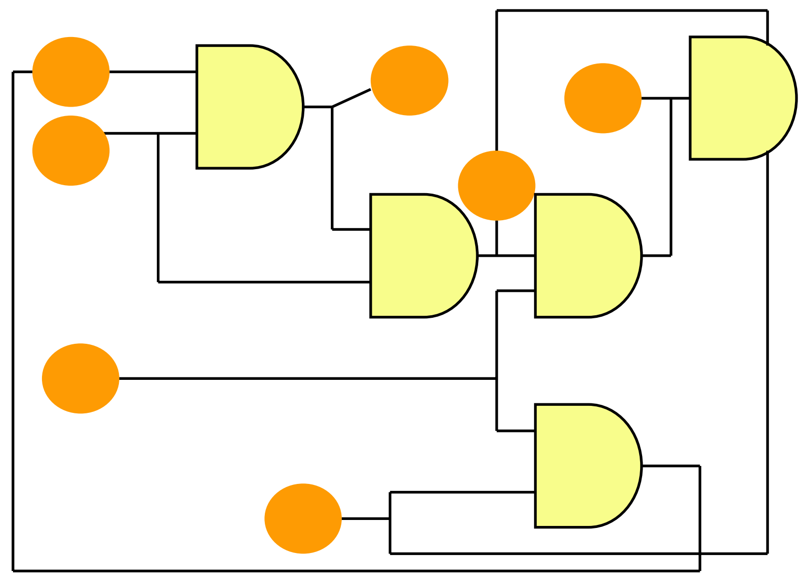
Cost: number of NAND2/NOR2 gates
Delay: 1 time unit/gate

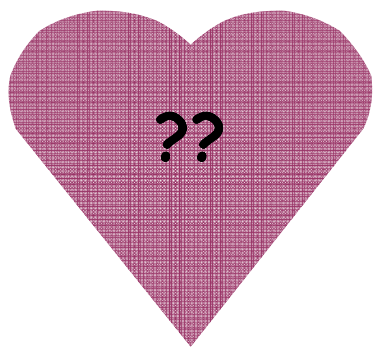
	Berkeley SIS		Caltech CYCLIFY			
benchmark	Area	Delay	Area	Improvement	Delay	Improvement
p82	175	19	167	4.57%	15	21.05%
t1	343	17	327	4.66%	14	17.65%
in3	599	40	593	1.00%	33	17.50%
in2	590	34	558	5.42%	29	14.71%
5xp1	210	23	180	14.29%	22	4.35%
bw	280	28	254	9.29%	20	28.57%
s510	452	28	444	1.77%	24	14.29%
s1	566	36	542	4.24%	31	13.89%
duke2	742	38	716	3.50%	34	10.53%
s1488	1016	43	995	2.07%	34	20.93%
s1494	1090	46	1079	1.01%	39	15.22%

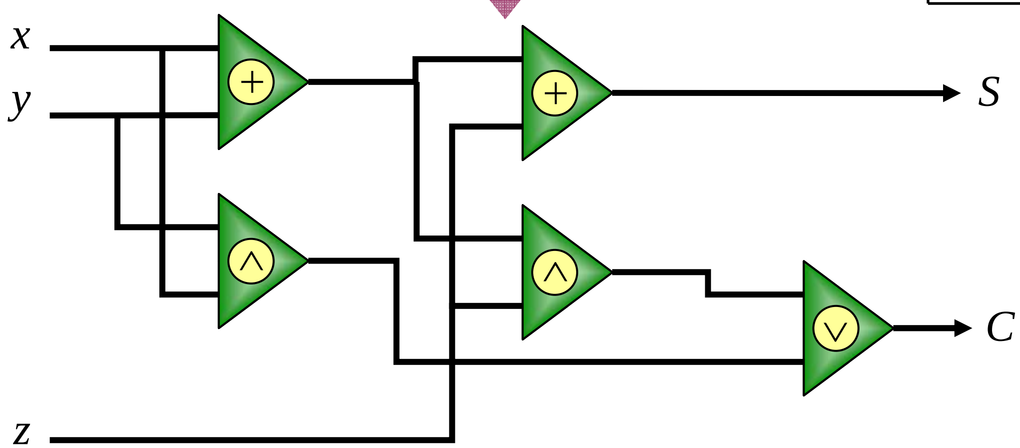
Bio Circuits vs. Combinational Logic Circuits

Joint work with Cook, Soloveichik and Winfree

- Cyclic vs. acyclic (feedback)
- Stochastic vs. deterministic



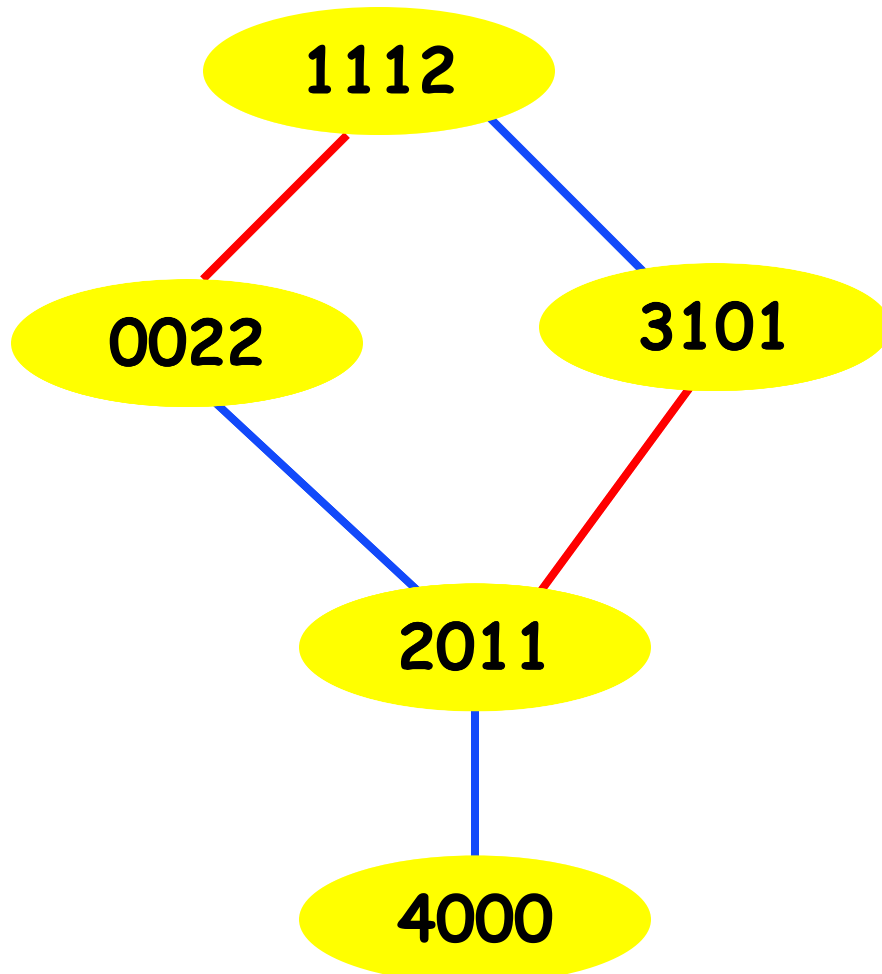




Computing with Systems of Chemical Reactions

$$A + B \longrightarrow C$$

$$C + D \longrightarrow 2A$$



The Reachability Question

Given a system of chemical reactions, and an initial state A (1112). Also given is a state B (4000).

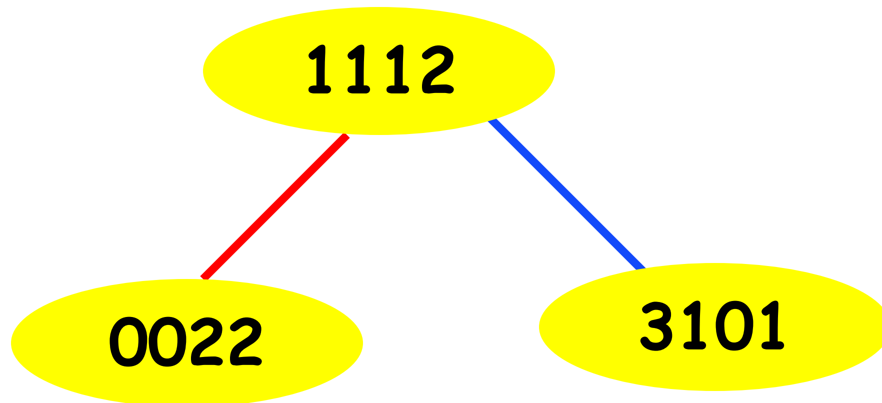
Starting at A, can the system of chemical reactions reach B?

- This question is decidable
- The state space is finite!!!
- Originally proved by Karp and Miller 1969 in the context of Vector Addition Systems (VAS)

Stochastic Chemical Reactions

$A + B \longrightarrow C$

$C + D \longrightarrow 2A$



The probability for a reaction to happen
is a monotonic function in the number of molecules
(#A x #B) or (#C x #D)

Stochastic Reachability

Given a system of chemical reactions, and an initial state A. Also given is a state B.

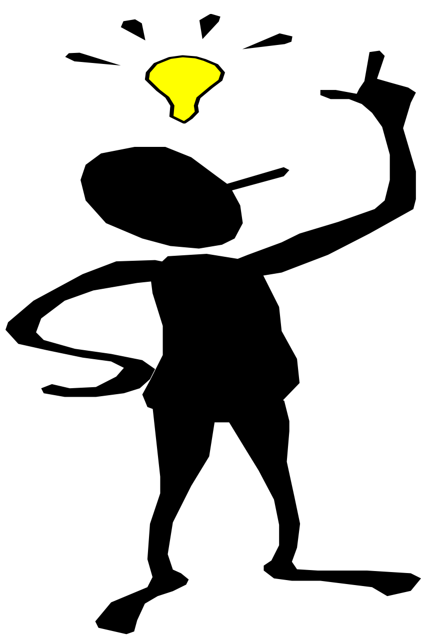
Starting at A, is the probability to reach B bigger than 1-$_{small}$?

Stochastic chemical reactions are Turing universal – with high probability

- This question is undecidable

Stochastic Behavior is a Feature

Probability enables general ('precise') computation in biochemical systems!!



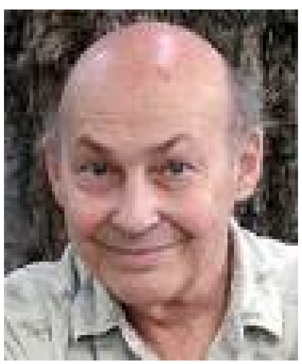
Idea: Simulate Register Machines with Chemical Reaction Networks

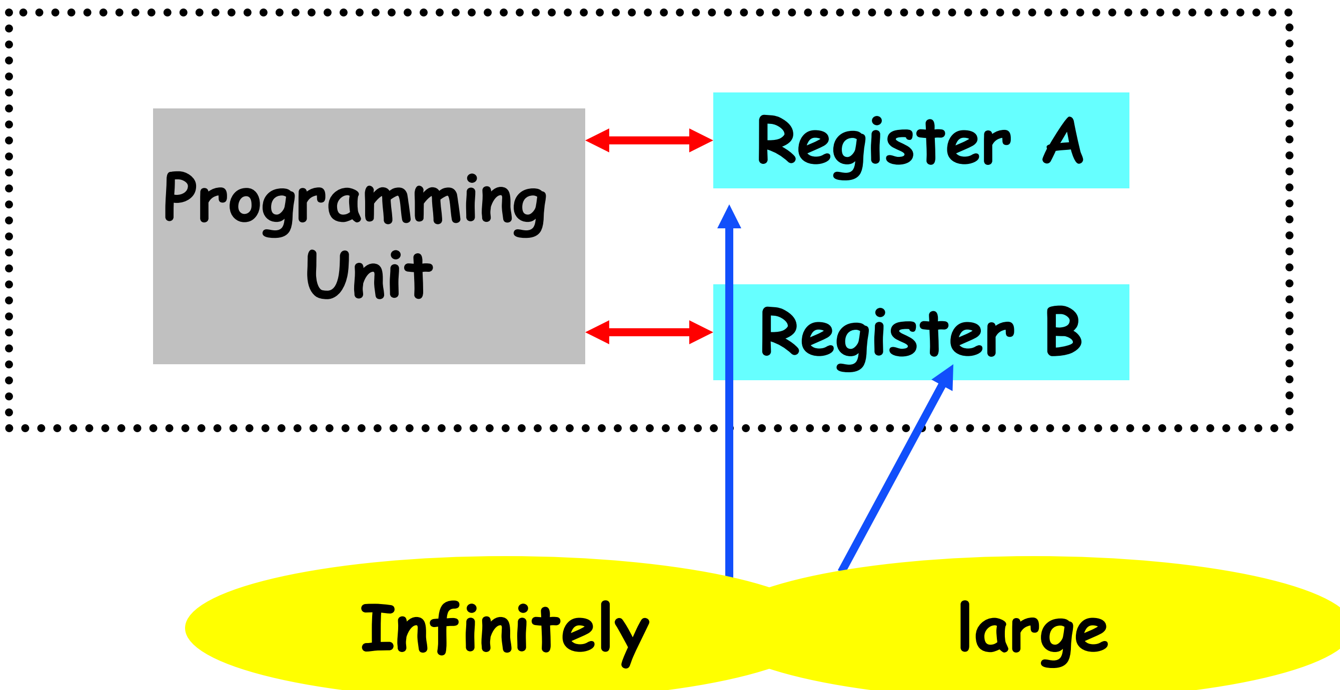
Register Machines are universal!! general computing

Probability enables general ('precise') computation in biochemical systems, Proof?

Register Machines (Minsky 1967)

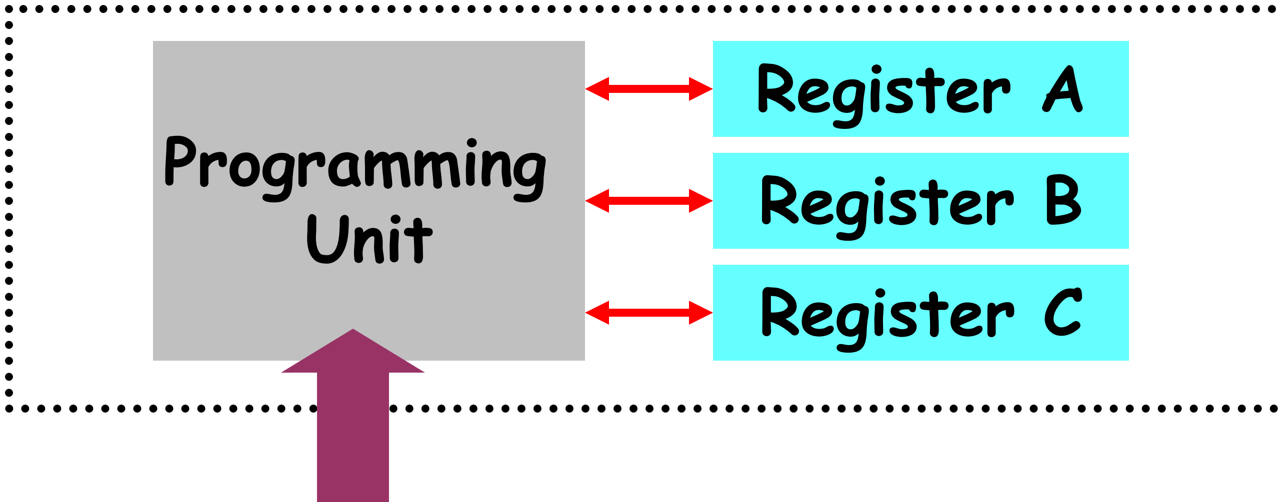
Marvin Minsky
1927 -





Register Machines

Register Machines (Minsky 1967)



Inc(A) – increment A and go to the next instruction

Dec(A,k) – if A is not 0, decrement A and go to next instruction
otherwise, if A is 0, go to instruction k

Register Machines - Example

1: Dec(A,4)
2: Dec(B,5)
3: Dec(C,1)
4: Inc(C)
5: Inc(C)

- Three registers
- a and b are nonnegative integers
- Let A=a , B=b and c = 0
- What is the program computing?

Inc(A) – increment A and go to the next instruction

Dec(A,k) – if A is not 0, decrement A and go to next instruction
otherwise, go to instruction k

Register Machines - Example

1: Dec(A,4)
2: Dec(B,5)
3: Dec(C,1)
4: Inc(C)
5: Inc(C)

- Three registers
- a and b are nonnegative integers
- Let A=a , B=b and c = 0
- What is the program computing?

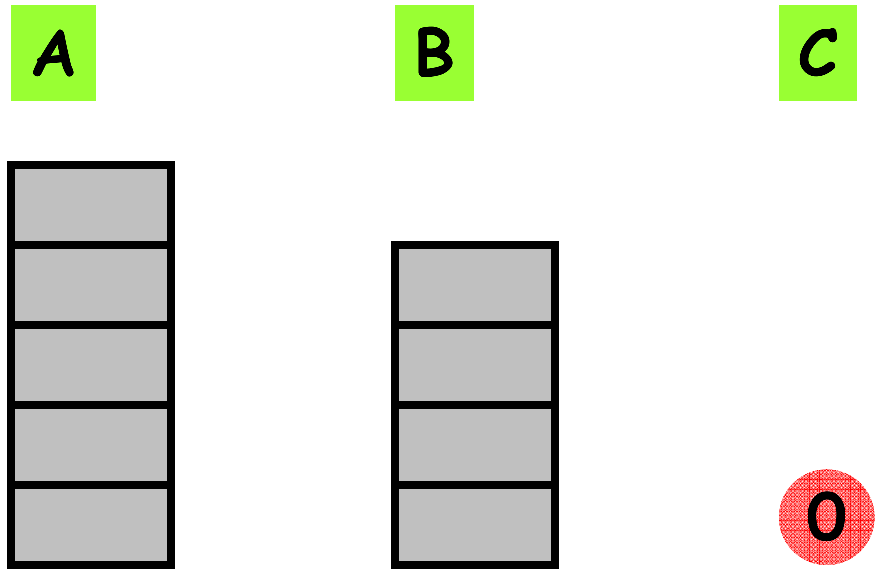
Output of program is in C

C=1 a is bigger than b

C=2 a is smaller or equal to b

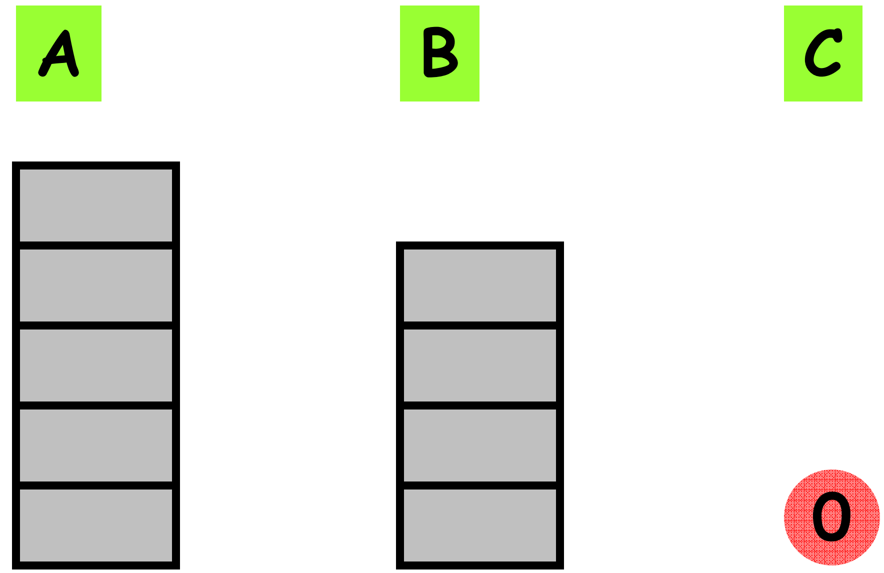
Register Machines - Example

1: Dec(A,4)
2: Dec(B,5)
3: Dec(C,1)
4: Inc(C)
5: Inc(C)



Register Machines - Example

1: Dec(A,4)
2: Dec(B,5)
3: Dec(C,1)
4: Inc(C)
5: Inc(C)



Register Machines - Example

1: Dec(A,4)
2: Dec(B,5)
3: Dec(C,1)
4: Inc(C)
5: Inc(C)

A B C

0 0 0

Output of program is in C

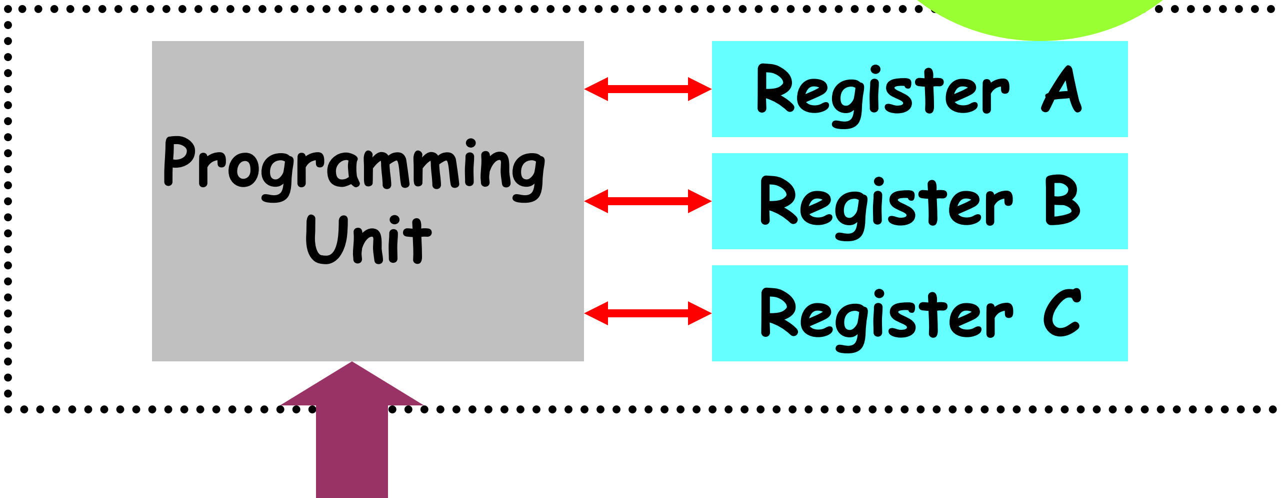
C=1 a is bigger than b

C=2 a is smaller or equal to b

Idea: Simulate Register Machines with Chemical Reaction Networks

Register Machines (Minsky 1967)

Are Universal



Inc(A) – increment A and go to the next instruction

Dec(A,k) – if A is not 0, decrement A and go to next instruction
otherwise, go to instruction k

Simulation of Register Machines with CRNs

Inc(A) – increment A and go to the next instruction

Dec(A,k) – if A is not 0, decrement A and go to next instruction
otherwise, go to instruction k

Compiler
RM to CRN

i: Inc(R) : $S_i \rightarrow R + S_{i+1}$

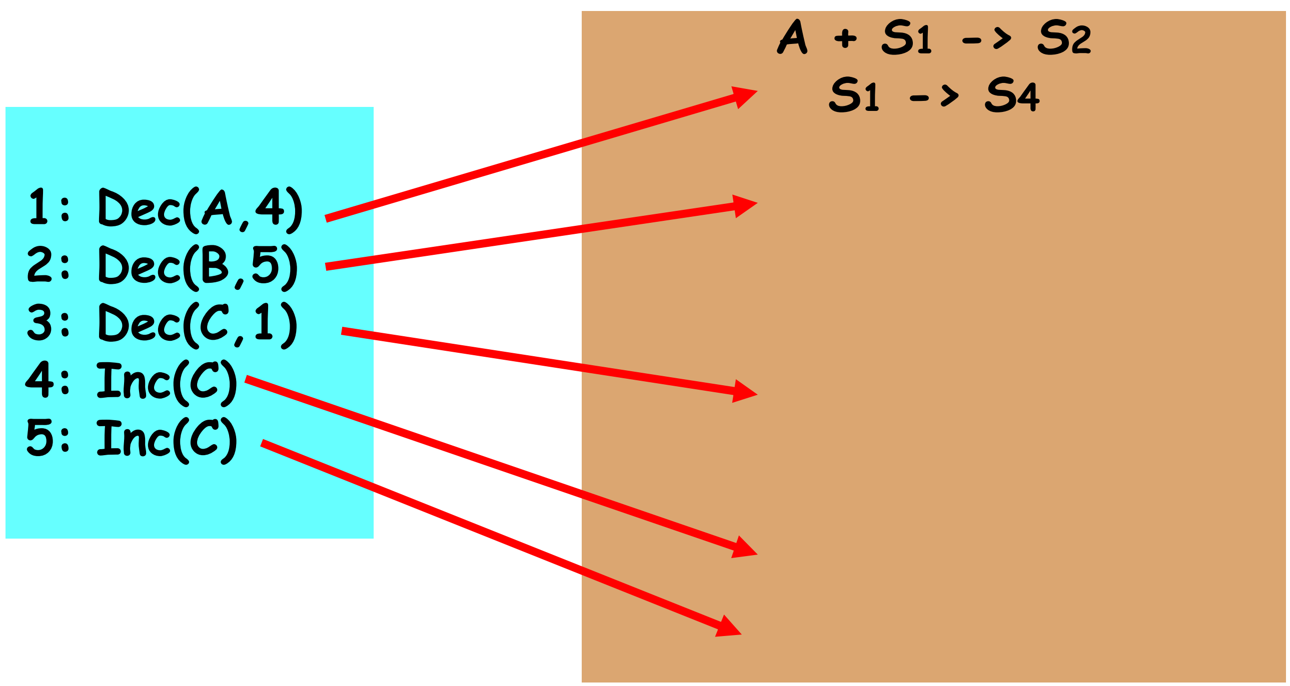
i: Dec(R,k): $R + S_i \rightarrow S_{i+1}$
If R=0 then $S_i \rightarrow S_k$

Simulation of Register Machines with CRNs

i: Inc(R) : $S_i \to R + S_{i+1}$

i: Dec(R,k): $R + S_i \to S_{i+1}$

If R=0 then $S_i \to S_k$



Simulation of Register Machines with CRNs

i: Inc(R) : $S_i \rightarrow R + S_{i+1}$

A Problem:
This reaction can happen even if R is not zero.....

i: Dec(R,k): $R + S_i \rightarrow S_{i+1}$
If R=0 then $S_i \rightarrow S_k$

1: Dec(A,4)
2: Dec(B,5)
3: Dec(C,1)
4: Inc(C)
5: Inc(C)

$A + S_1 \rightarrow S_2$
$S_1 \rightarrow S_4$

$B + S_2 \rightarrow S_3$
$S_2 \rightarrow S_5$

$C + S_3 \rightarrow S_4$
$S_3 \rightarrow S_1$

$S_4 \rightarrow C + S_5$

$S_5 \rightarrow C + S_6$

Simulation of Register Machines with CRNs

i: Inc(R) : $S_i \rightarrow R + S_{i+1}$

The solution:
Delay this reaction using a "stochastic clock"

i: Dec(R,k): $R + S_i \rightarrow S_{i+1}$
If R=0 then $S_i \rightarrow S_k$

1: Dec(A,4)
2: Dec(B,5)
3: Dec(C,1)
4: Inc(C)
5: Inc(C)

$A + S_1 \rightarrow S_2$
$S_1 \rightarrow S_4$

$B + S_2 \rightarrow S_3$
$S_2 \rightarrow S_5$

$C + S_3 \rightarrow S_4$
$S_3 \rightarrow S_1$

$S_4 \rightarrow C + S_5$

$S_5 \rightarrow C + S_6$

Simulation of Register Machines with CRNs

i: Inc(R) : Si -> R + Si+1

Delay this reaction using a "stochastic clock"

When R>0, it is less likely to happen.

i: Dec(R,k): R + Si -> Si+1
If R=0 then Si -> Sk

DEC (R,i)

Case 1: R=0

Si -> Sk with probability 1

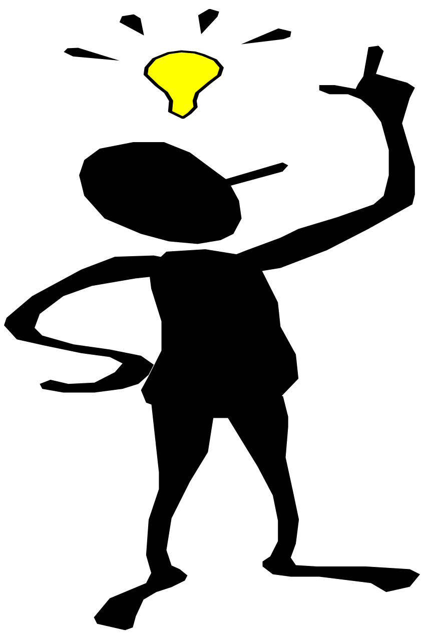
Case 2: R>0

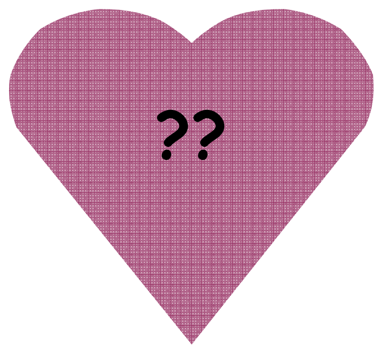
R + Si -> Si+1
with probability close to 1

Si -> Sk with small probability

Stochastic Behavior is a Feature

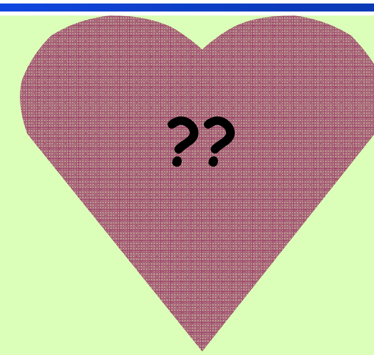
Probability enables general ('precise') computation in biochemical systems!!





Is it a Feature or a Bug?

- Stochastic vs. deterministic
  Probability enables universal computation in chemical reaction networks
  (Cook, Soloveichik, Winfree, Bruck, 2005)



- Cyclic vs. acyclic
  Cycles enable cost savings in real combinational circuits
  (Riedel & Bruck 2003)

Current / future work:

- Relations vs. functions?
- The logic of computing probability distributions?

"The further back you look, the further forward you can see"

Winston Churchill

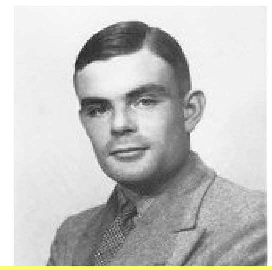
Calculus for Biology??

We need to learn / teach about abstract systems for reasoning about information

Emil Post
1897-1954

Compositions of Boolean functions
Universal Algebra

Turing
1912-1954



Defined Computing via universal machines
Computer Science

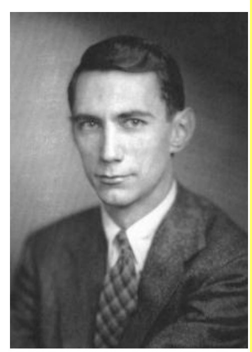
Leibniz
1646-1716

- Logic and Binary system
- Calculus

Boole
1815-1864

Connected Logic with Algebra
Boolean Algebra
Logical Calculation

Shannon
1916-2001



- Connected Boolean Algebra to Electrical Circuits
  Logic Design
- Connected probability to Communications
  Information Theory