How Does Nature Compute?

Lila Kari

Dept. of Computer Science

University of Western Ontario

London, ON, Canada

http://www.csd.uwo.ca/~lila/

lila@csd.uwo.ca

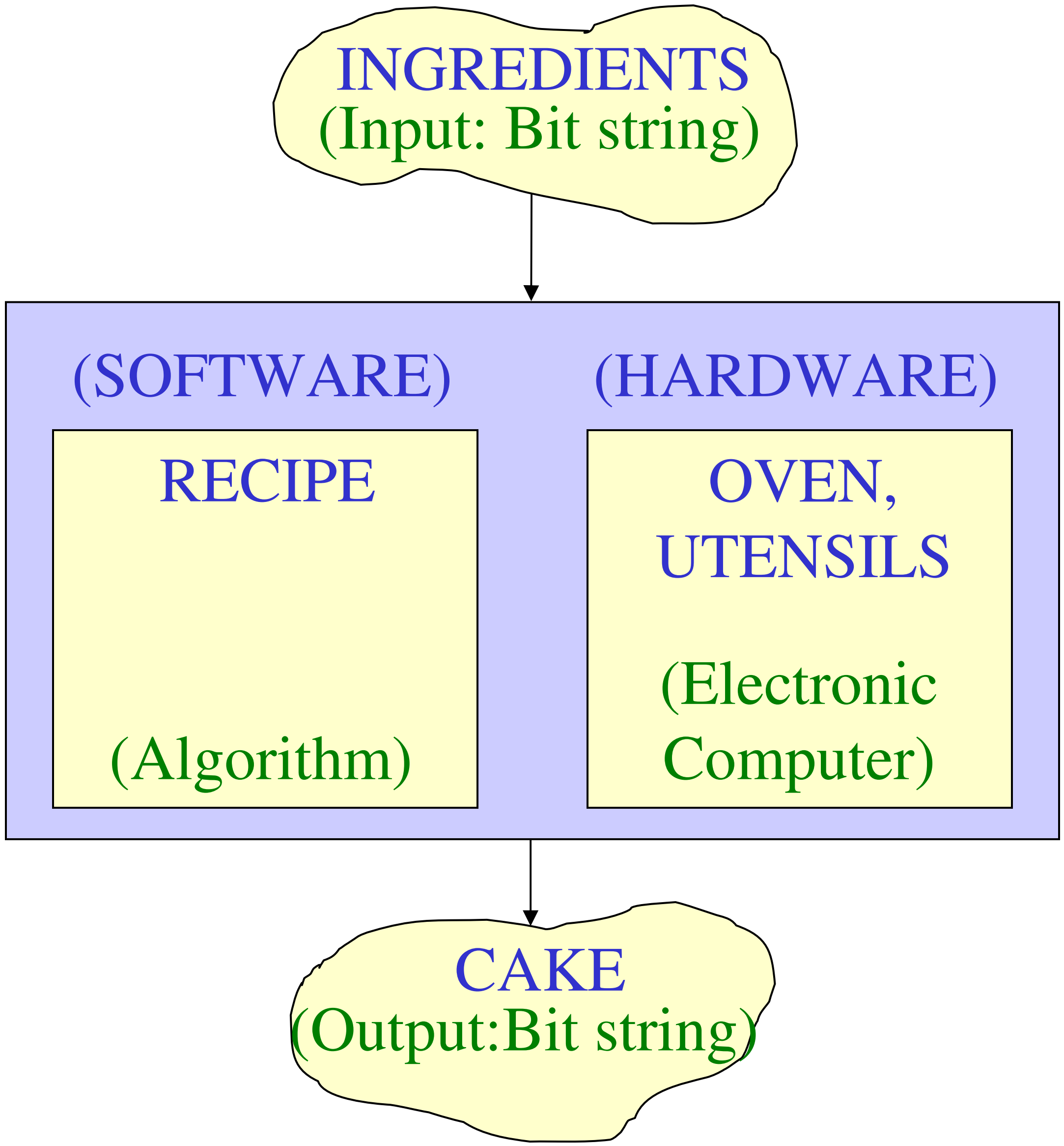
Computers: What can they accomplish?

- Fly spaceships to Mars
- Control aircraft
- Robot aided manufacturing
- Computer games
- Expedite journal submissions
- Email

Computers:
What do they actually do?

- Computers = a collection of switches (bits) that are on (1) or off (0).
- Can execute only simple operations
  - § Flipping a bit's value
  - § Zeroing a bit
  - § Testing a bit

How do they do it?


INGREDIENTS
(Input: Bit string)
(SOFTWARE)
(HARDWARE)
RECIPE
(Algorithm)
OVEN,
UTENSILS
(Electronic
Computer)
CAKE
(Output:Bit string)

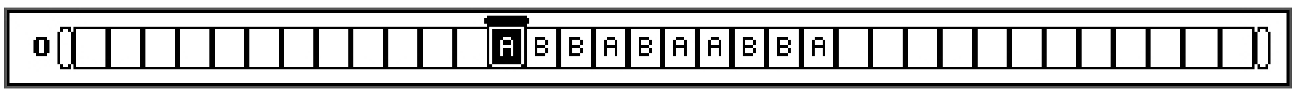
Formal Models of Computing: Turing Machines

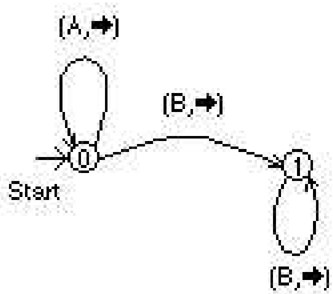
- Data
  - § String of symbols written on a tape
- Operations
  - § Read a square
  - § Overwrite the symbol with another
  - § Move left or right

Turing Machine

Computation = Finite list of instructions

"If you are in state S and read input symbol X then write Y and move Left/Right"





TO RUN: enter a sequence of A's and B's, position read head on leftmost symbol, and start.

	State	Significance
Start	0	Has not yet seen a B
	1	Has seen at least one B

Lila Kari, University of Western Ontario

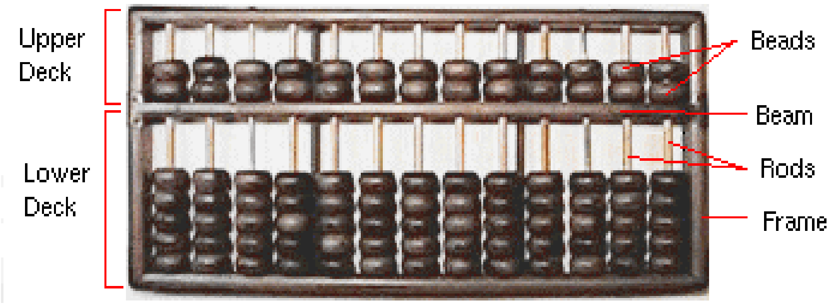
Turing Machine

- Turing machines are capable of universal computation (everything that can be computed can be computed by a TM)
- The abstract notion of computation (Turing machine, algorithm, program) is hardware independent

History of Hardware

- Abacus
- Pascal
- Jacquard
- Babbage
- Hollerith
- ENIAC
- Chip

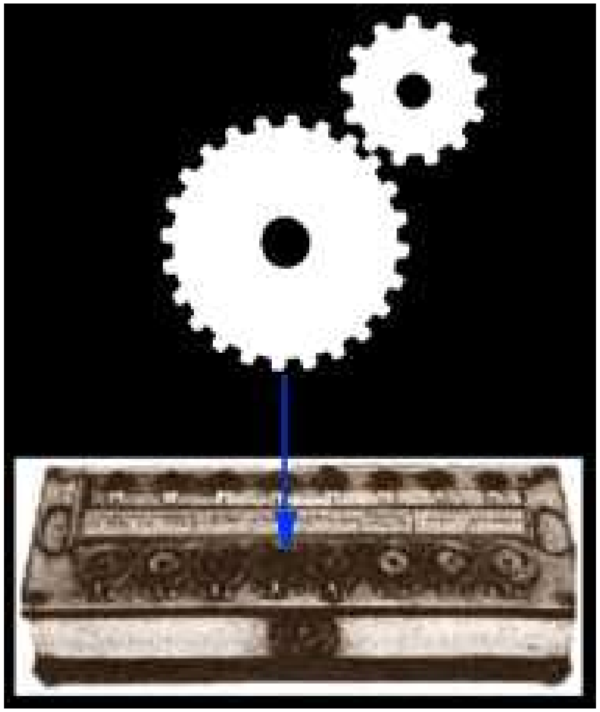
Abacus



History of Hardware

- Abacus
- Pascal
- Jacquard
- Babbage
- Hollerith
- ENIAC
- Chip

Mechanical adding machine (1642)



History of Hardware

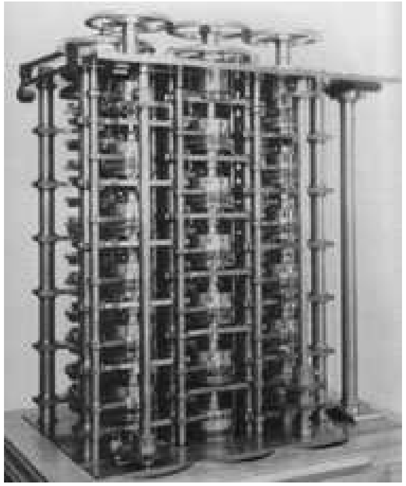
- Abacus
- Pascal
- Jacquard
- Babbage
- Hollerith
- ENIAC
- Chip

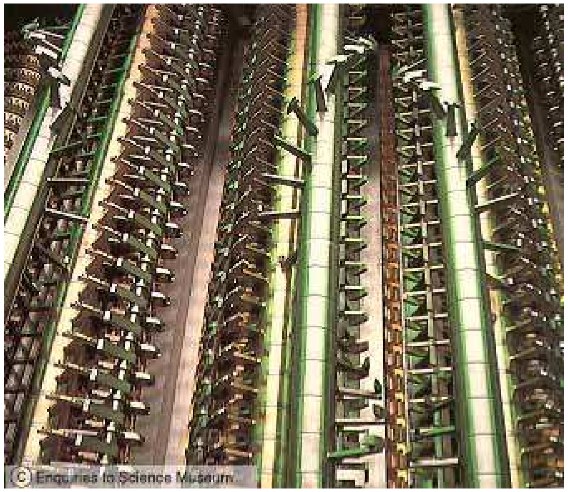
Jacquard's punch card loom (1801)

History of Hardware

- Abacus
- Pascal
- Jacquard
- Babbage
- Hollerith
- ENIAC
- Chip

Babbage's difference engine (1833)

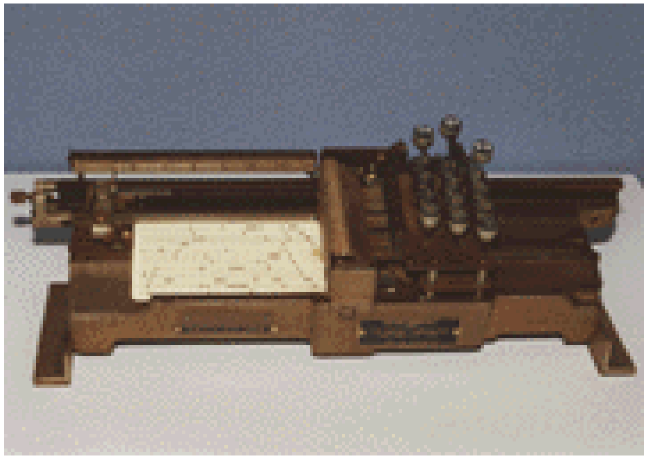




History of Hardware

- Abacus
- Pascal
- Jacquard
- Babbage
- Hollerith
- ENIAC
- Chip

Hollerith punch card system (1890)



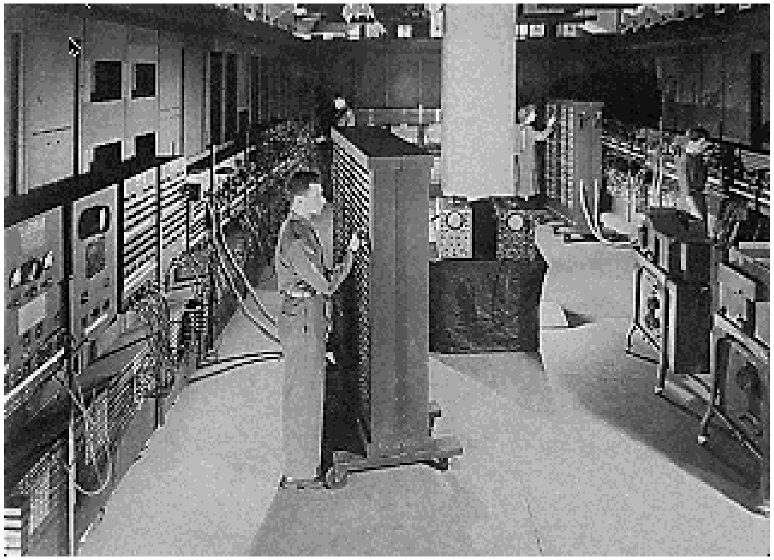
History of Hardware

- Abacus
- Pascal
- Jacquard
- Babbage
- Hollerith
- ENIAC
- Chip

ENIAC (1939-45) -167 sq.m.

- 18,000 vacuum tubes

- not programmable



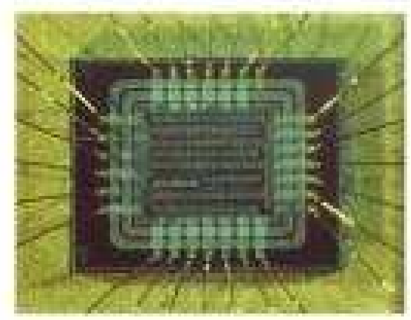
History of Hardware

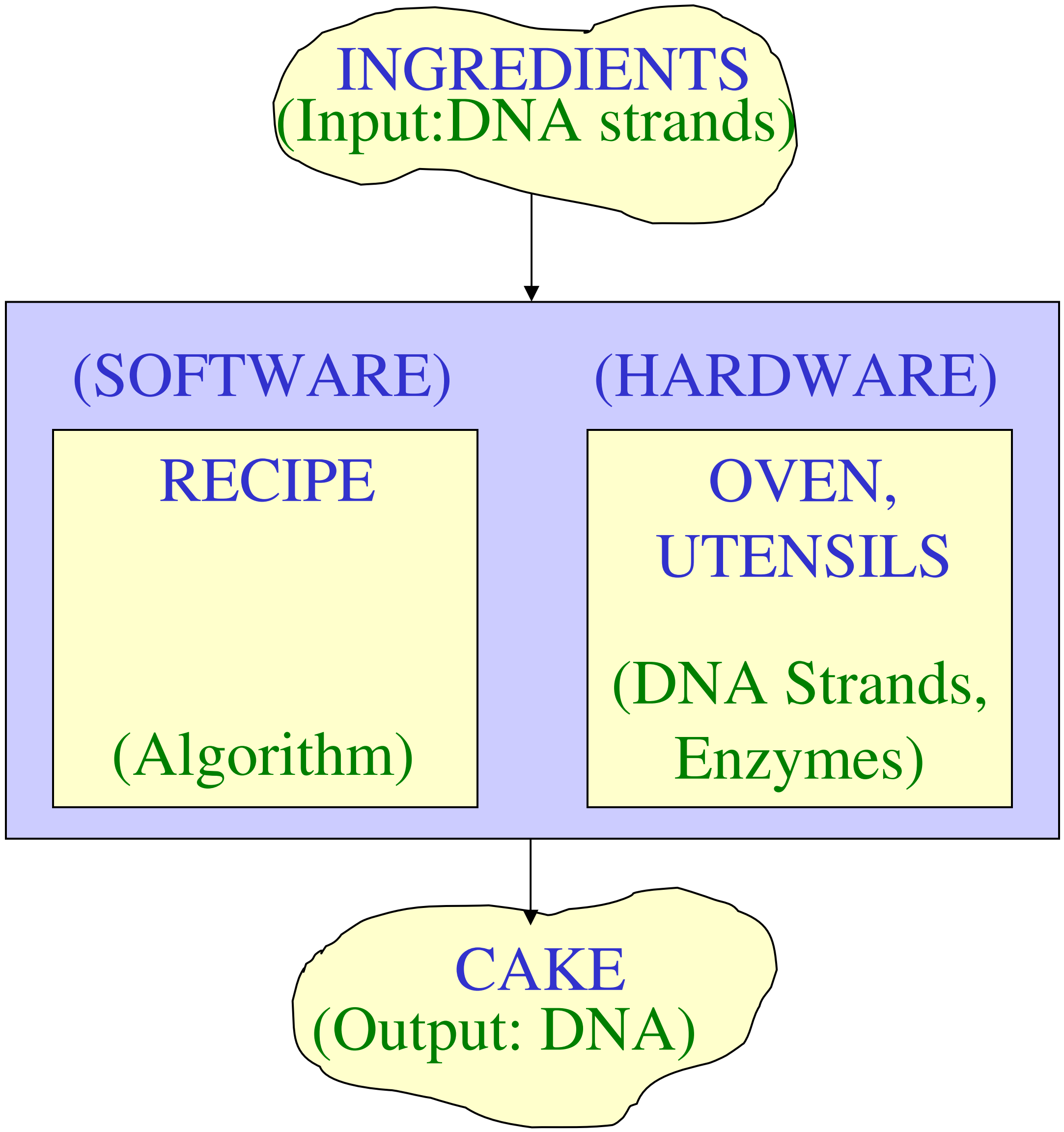
- Abacus
- Pascal
- Jacquard
- Babbage
- Hollerith
- ENIAC
- Chip

Modern computer chip

- Transistors

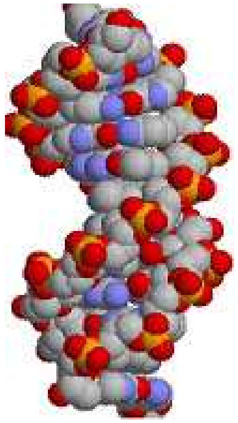
- Integrated circuit




INGREDIENTS
(Input:DNA strands)
(SOFTWARE)
(HARDWARE)
RECIPE
(Algorithm)
OVEN,
UTENSILS
(DNA Strands,
Enzymes)
CAKE
(Output: DNA)

DNA Computer

- Input / Output (DNA)
  - § Data encoded using the DNA alphabet {A, C, G, T} and synthesized as DNA strands
- Bio-operations
  - § Cut
  - § Paste
  - § Copy
  - § Anneal
  - § Recombination



Biomolecular (DNA) Computing

- Hamiltonian Path Problem [Adleman, Science, 1994]
- DNA-based addition [Guarnieri et al, Science, 1996]
- Maximal Clique Problem [Ouyang et al, Science, 1997]
- DNA computing by self-assembly [Winfree et al, Nature 1998]
- Computations by circular insertions, deletions [Daley et al,1999]
- DNA computing on surfaces [Liu et al, Nature, 2000]
- Molecular computation by DNA hairpin formation [Sakamoto et al, Science, 2000]
- Programmable and autonomous computing machines made of biomolecules [Benenson et al, Nature, 2001]
- 20-variable Satisfiability [Braich et al., Science 2002]

How Does Nature Compute?

- Technical difficulties encountered in experimental DNA computations (error-detection, error-correction) are routinely solved by biological systems in nature
- Idea: study and utilize the computational abilities of unicellular organisms

Ciliates: Unicellular Protozoa

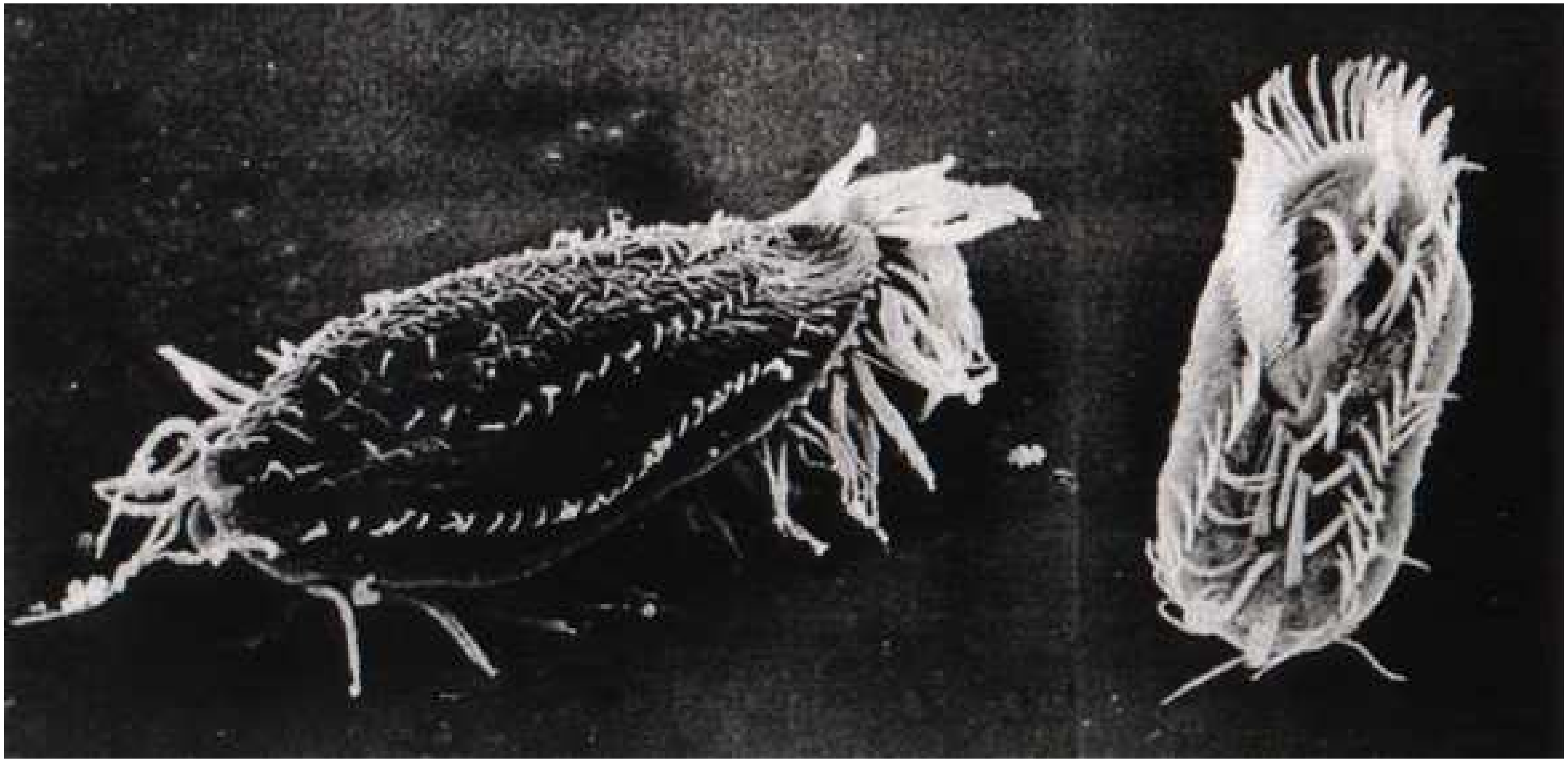


Photo courtesy of L.F. Landweber

Ciliates: Genetic Info Exchange

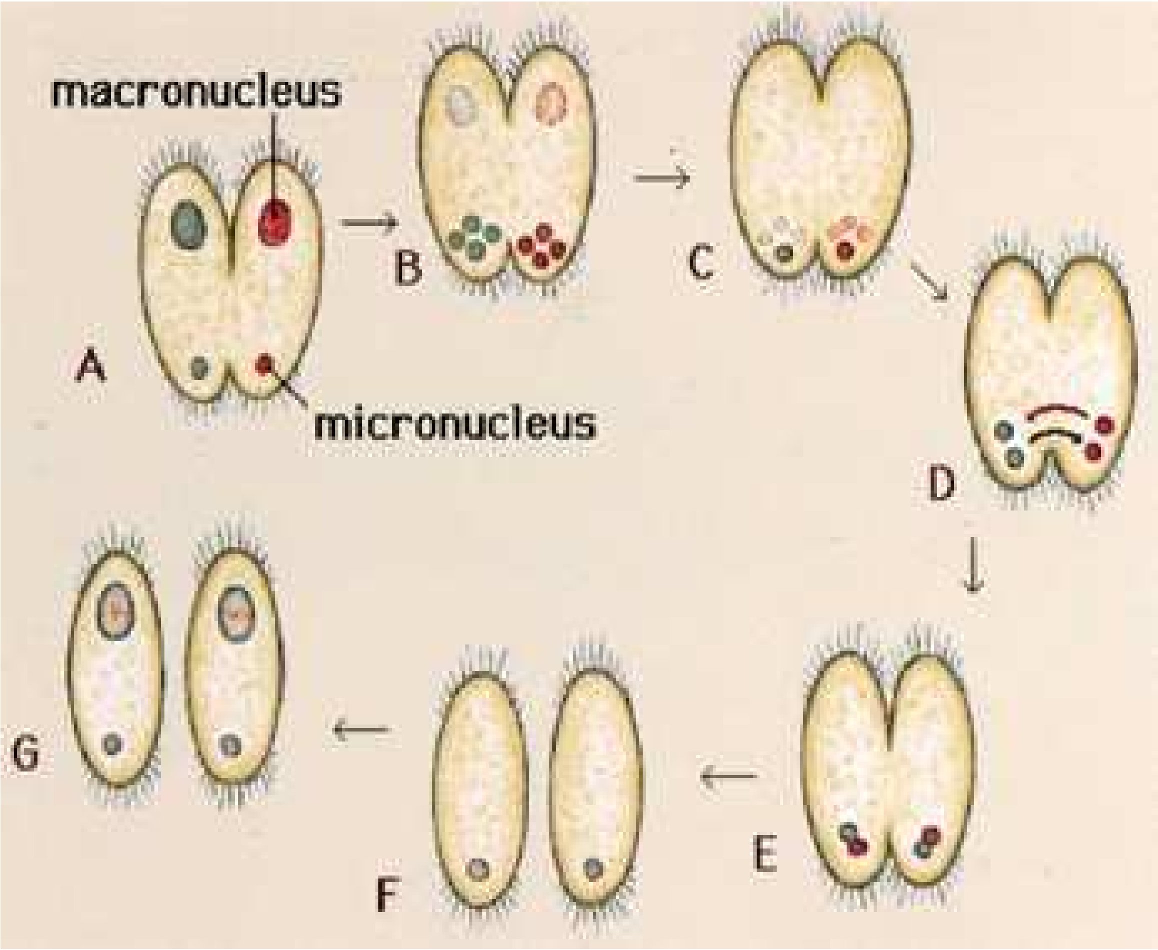


Photo courtesy of L.F. Landweber

Ciliates: Gene Rearrangement

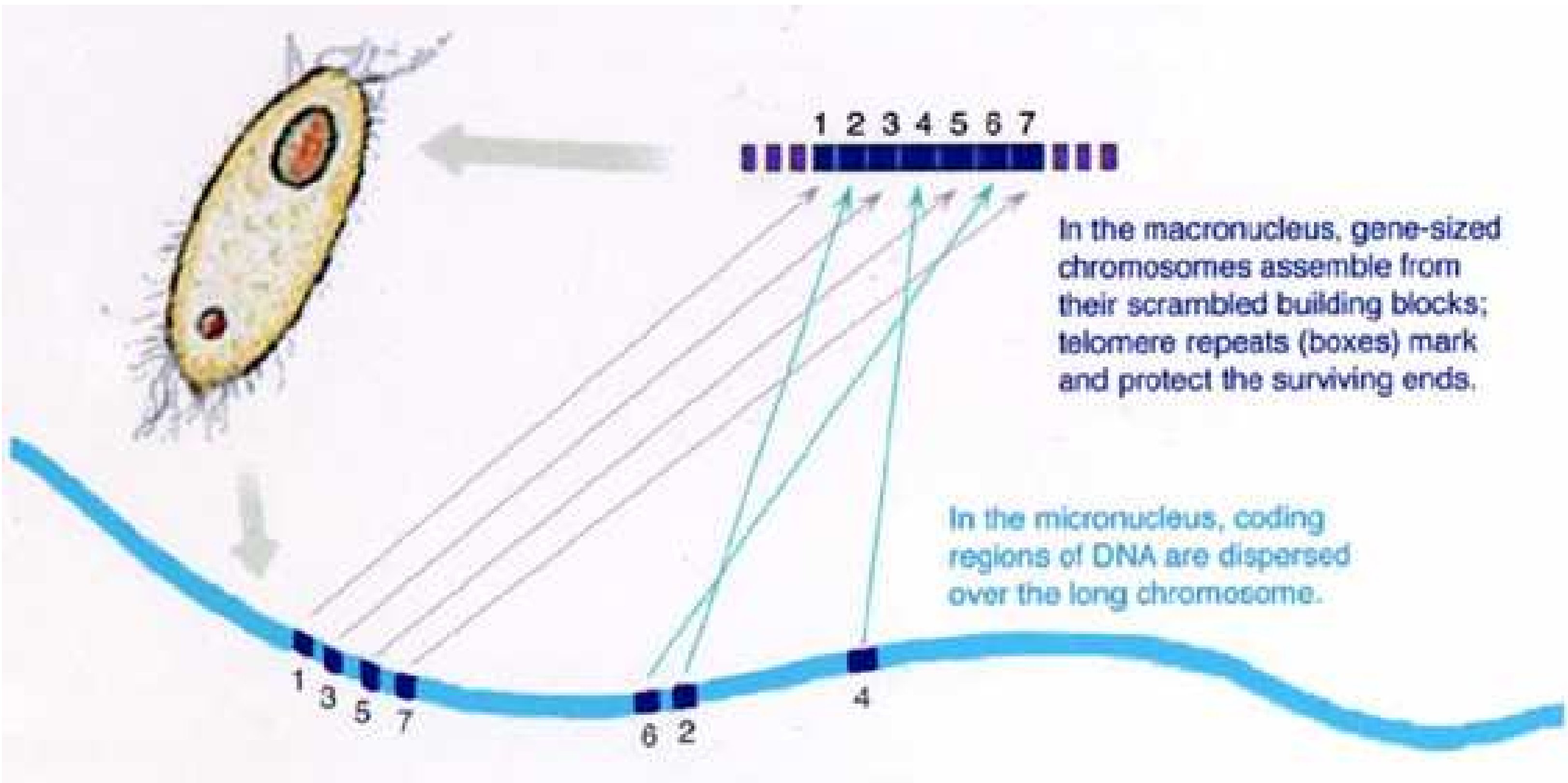
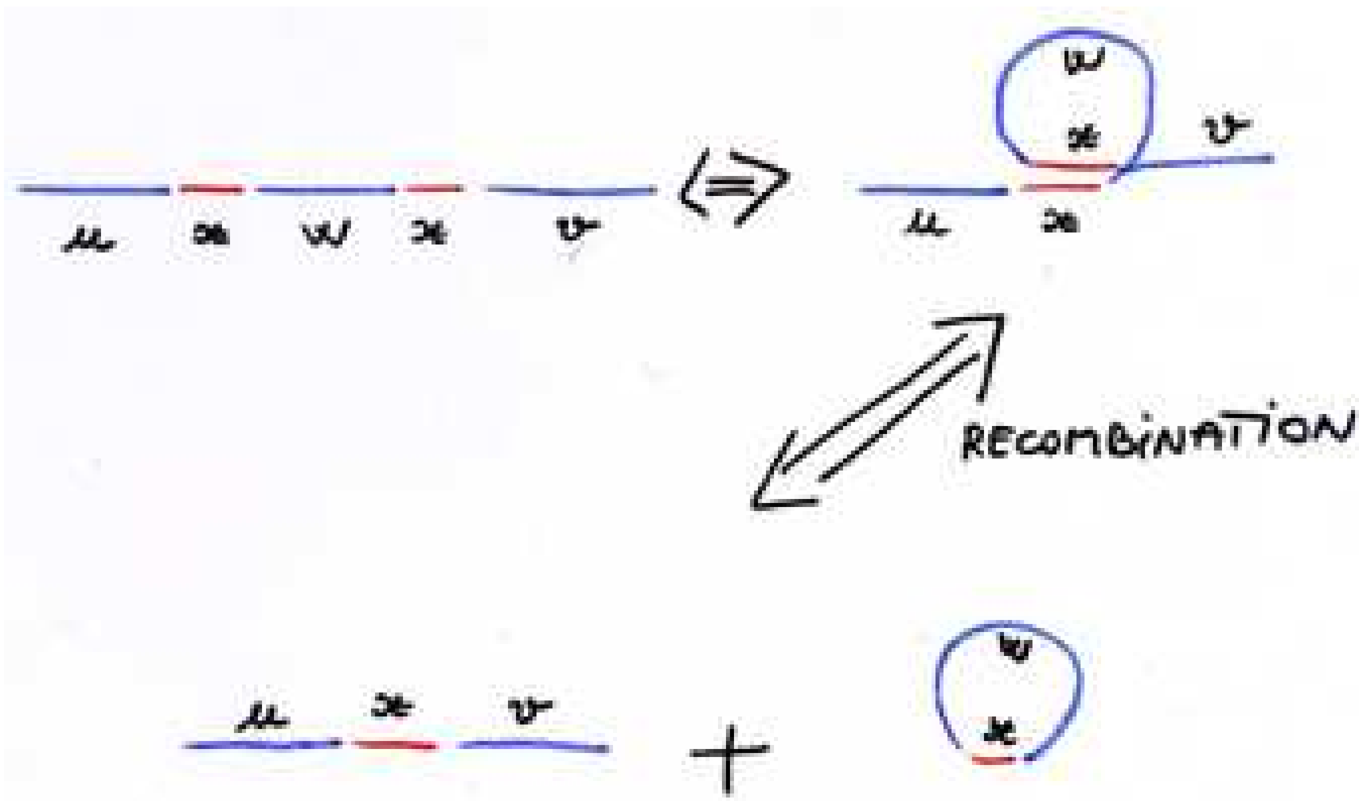


Photo courtesy of L.F. Landweber

Lila Kari, University of Western Ontario

Ciliates: Bio-operations



Ciliates: Results

- Guided Recombination System = A formal computational model based on contextual circular insertions and deletions
- Such systems have the computational power of Turing Machines (Landweber, Kari, ’99)
- The model is consistent with the limited knowledge of this biological process

Essential Feature of Biocomputing: Self-Assembly

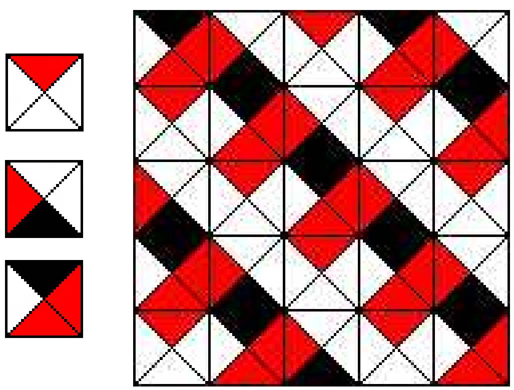
- Use knowledge of how simple components (DNA molecules, enzymes) interact
- Design a setup such that the computation happens essentially by itself
- Useful in nano-technology where components are too small for existing tools

Model of Self-Assembly

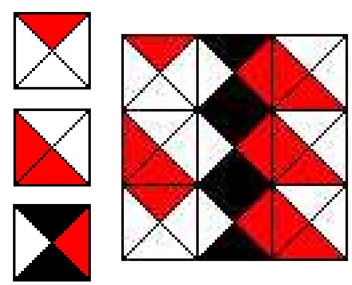
- Tile
  - A 1 x 1 square
  - Each side is "painted" with a certain kind of glue
  - Tiles cannot be rotated
  - Two "adjacent" tiles will "stick" only if they have matching glues at the touching edges
- Tile system
  - T = A finite number of tile types (as above)
  - Unlimited supply of each tile type available

(Classic) Tiling Problem

- Can any square, of any size, be tiled using only the available tile types, without violating the glue-matching rule?



Yes



No

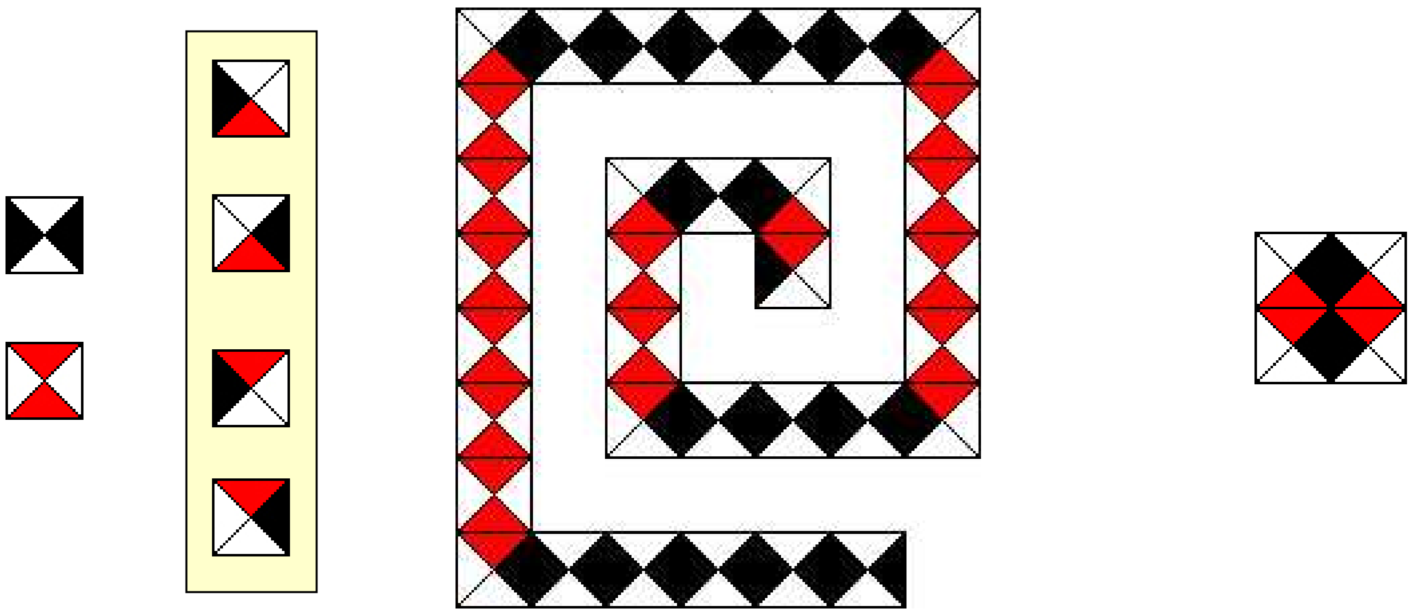
Self-Assembly Problems

- What is the minimal number of tile types that can self-assemble into a given shape and nothing else?
- What is the optimal initial concentration of tile types that ensures fastest self-assembly?
- What happens if “bonds” have different strengths?

The Ribbon Problem

- Given a tile system, can we determine whether or not it can produce shapes that grow indefinitely?
- Can we decide whether or not a set of given tiles can produce unlimited-size ribbons?

Generating Ribbons



Answer

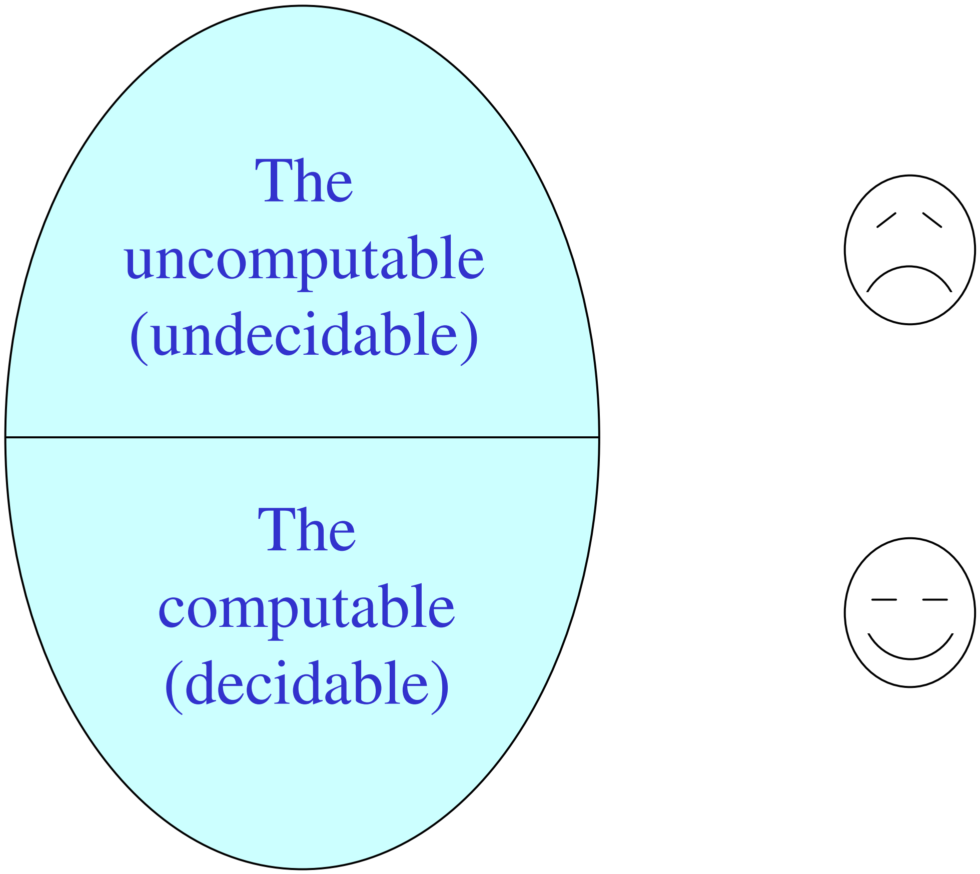
- There is no algorithm, and there never will be, for solving the Ribbon Problem!
  [Adleman, Kari, Kari, Reishus, 2002]

- You can devise a program that might work quite well, on some of the inputs. But there always will be inputs upon which your algorithm will misbehave; it will either run forever, or produce the wrong output.

Computable vs. Uncomputable!

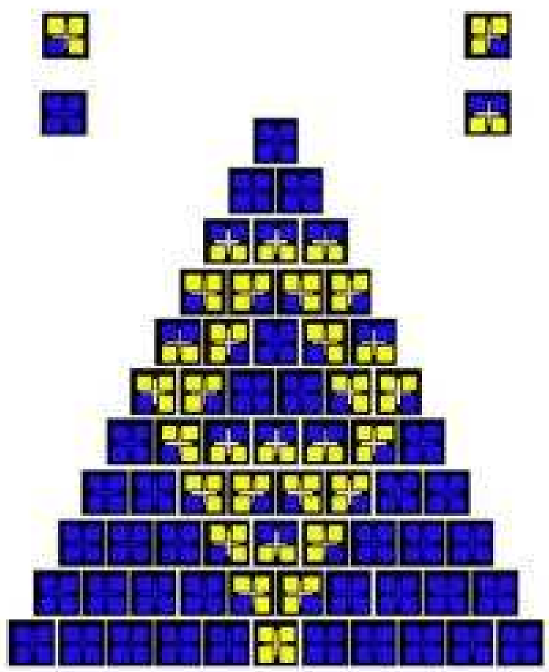
- An algorithmic problem that admits no solution is termed uncomputable
- If it is a Yes/No problem, it is termed undecidable
- The Ribbon Problem is undecidable

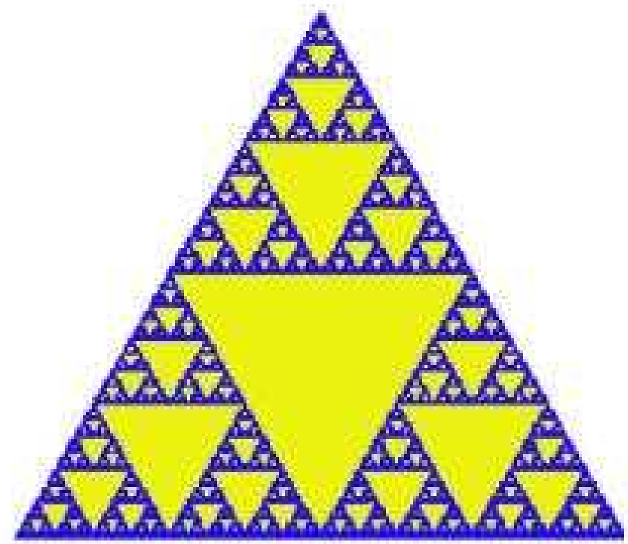
Sometimes we cannot do it!



Experimental Self-Assembly

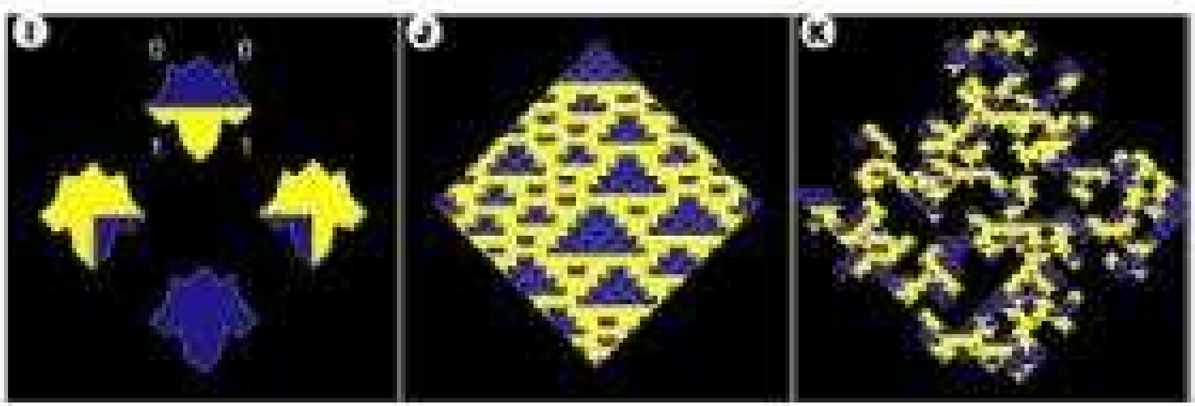
- Self-assembly using capillary forces (Rothemund)
- DNA computing by self-assembly (Winfree, Seeman)





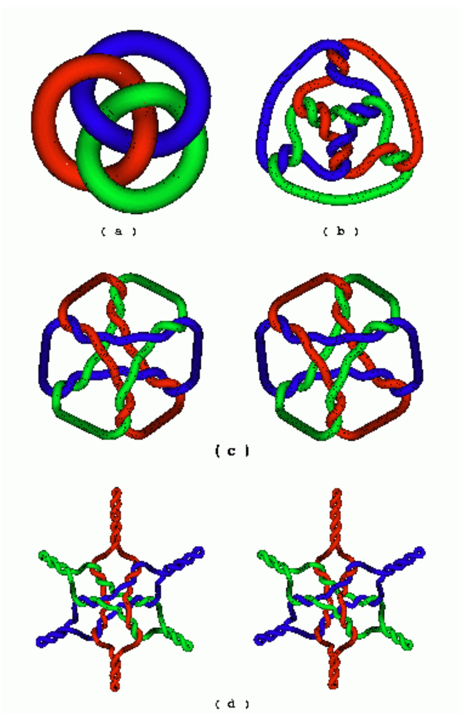
Experimental Self-Assembly

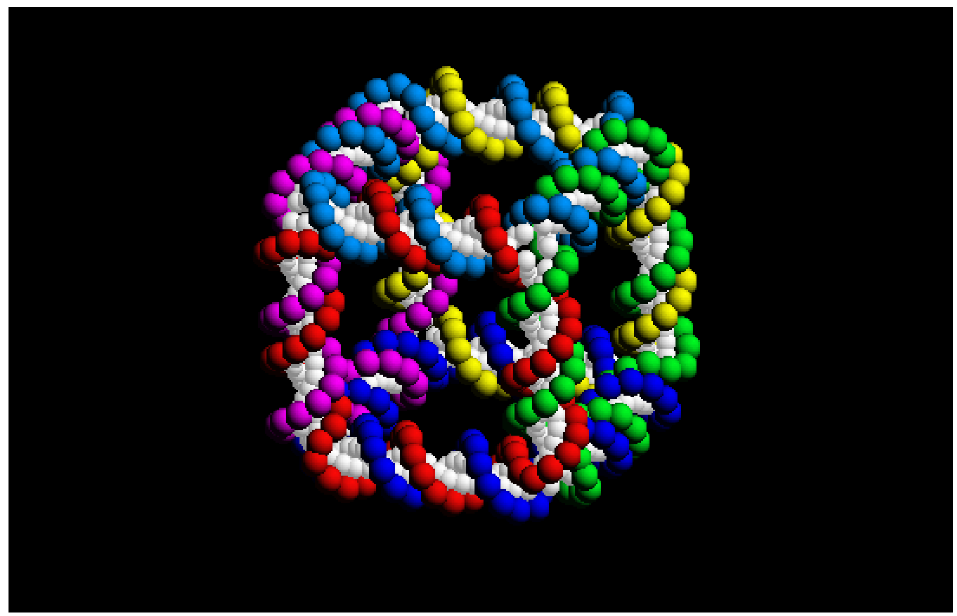
- Self-assembly using capillary forces (Rothemund)
- DNA computing by self-assembly (Winfree, Seeman)



Experimental Self-Assembly

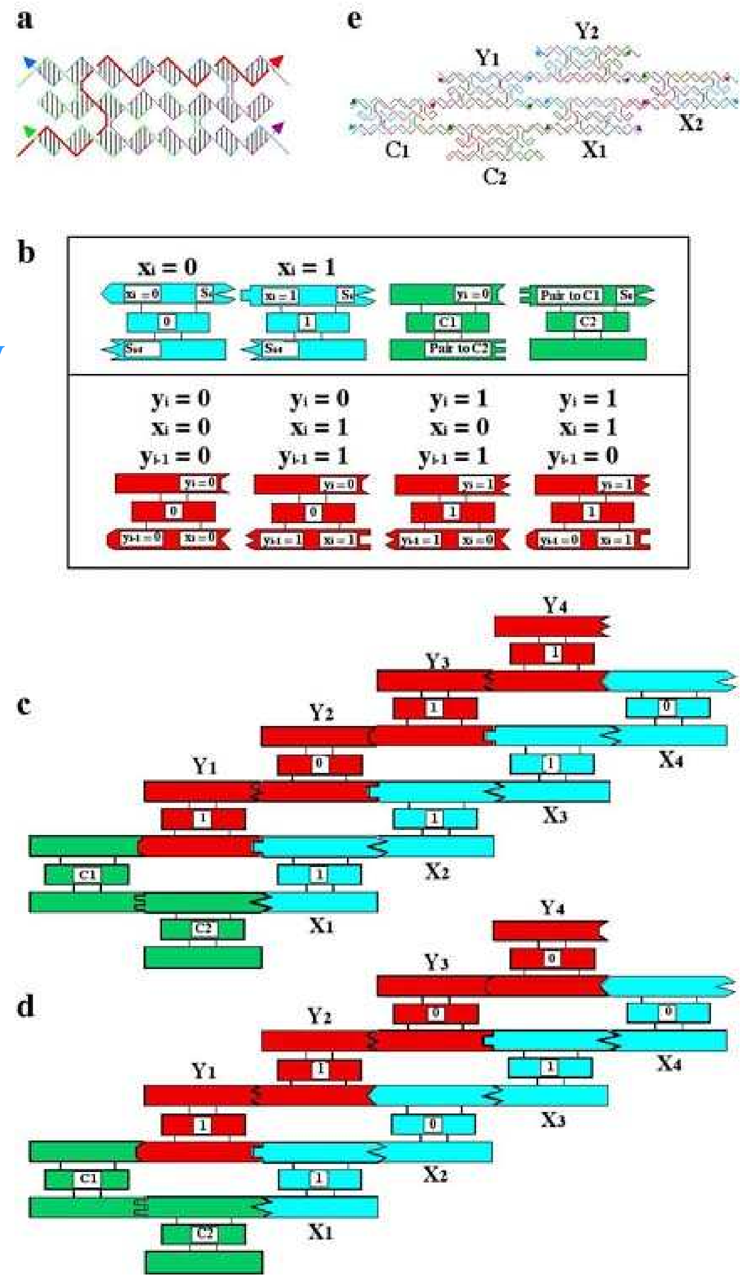
- Self-assembly using capillary forces (Rothemund)
- DNA computing by self-assembly (Winfree, Seeman)





Lila Kari, University of Western Ontario

DNA computing by self-assembly (Winfree, Seeman, Mao, LaBean, Reif)



Lila Kari, University of Western Ontario

Potential Advantages of Biomolecular (DNA) Computing

- Information density

  1 gram of DNA (1 cm^3 when dry) = 1 trillion CDs

  1 lb DNA – more memory then all computers together.

- Speed

  Thousand to million times faster than an electronic computer due to massive parallelism

- Energy consumption

  thousand times more energy efficient

IMPACT OF DNA COMPUTING

- Sheds new light into the nature of computation
- Opens prospects of radically different computers
- Could lend new insights into the information processing abilities of cells
- “Biology and Computer Science – life and computation – are related” (Adleman)

"If we knew what it was that we were doing, it wouldn't be called research, would it?" (Einstein)