Microsoft Research

Microsoft

Molecular Programming

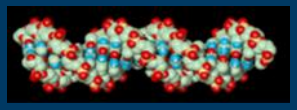
Luca Cardelli

2013-05-15 Hasselt

Objectives

- The promises of Molecular Programming:
  - In Science & Medicine
  - In Engineering
  - In Computing

- The current practice of Molecular Programming
  - DNA technology
  - Molecular languages and tools
  - Example of a molecular algorithm



Microsoft Research

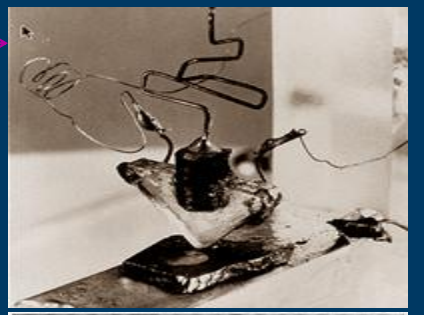
Microsoft

The Hardware Argument

Smaller and smaller things can be built

Smaller and Smaller

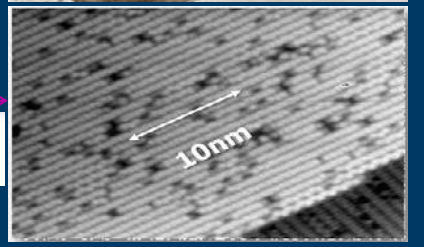
First working transistor
John Bardeen and Walter Brattain , Dec. 23, 1947



First integrated circuit
Jack Kilby, Sep. 1958.

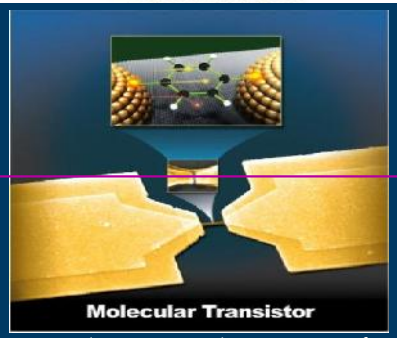
50 years later

25nm NAND flash
Intel&Micron, Jan. 2010. ~50atoms

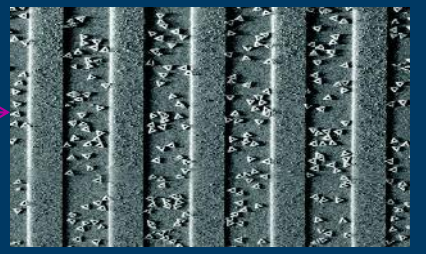


Scanning tunneling microscope image of a silicon surface showing 10nm is ~20 atoms across

Single molecule transistor
Observation of molecular orbital gating
Nature, 2009; 462 (7276): 1039



Molecules on a chip



Placement and orientation of individual DNA shapes on lithographically patterned surfaces. Nature Nanotechnology 4, 557 - 561 (2009).

~10 Moore's Law cycles left!

Microsoft Research

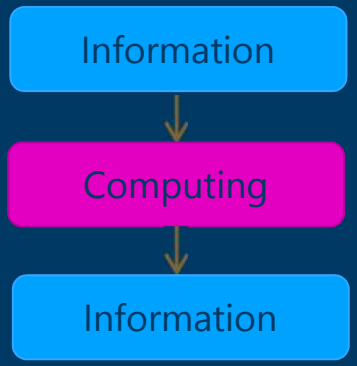
Microsoft

The Software Argument

Smaller and smaller things can be programmed

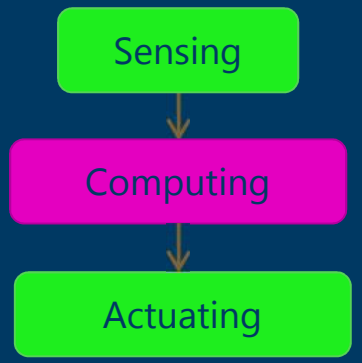
We can program…

- Computers.
  - **Completely!**



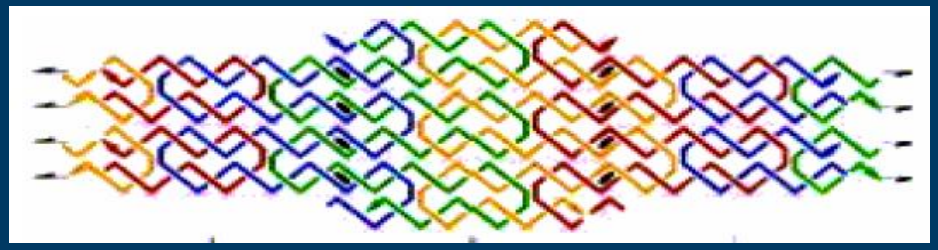
We can program…

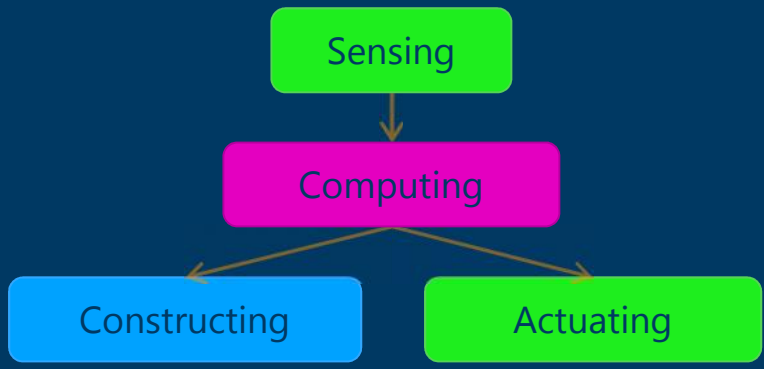
- Physical systems.
  - Completely!
    (Modulo sensors/actuators)



We can program…

- Matter
  - Completely and directly!
  - Currently: only DNA/RNA.

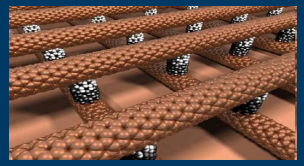


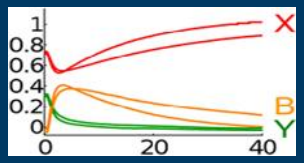


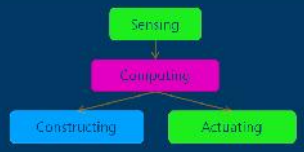
It's like a 3D printer without the printer!
[Andrew Hellington]

What can we do with "just" DNA?

- Organize ANY matter [caveats apply]
- Execute ANY kinetics [caveats: up to time scaling]
- Build Nano-Control Devices
- Interface to Biology





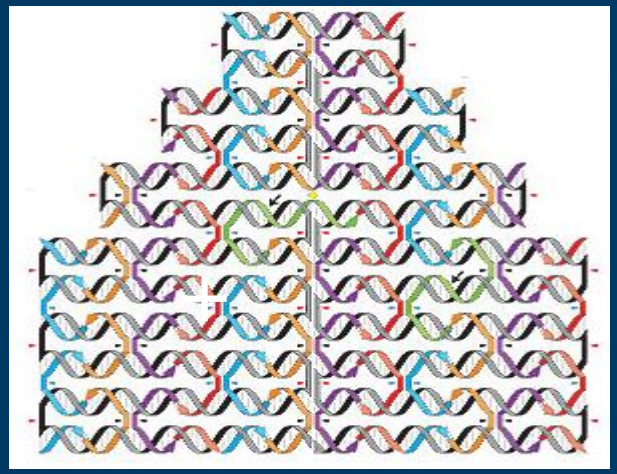


H.Lodish & al. Molecular Cell Biology 4th ed.

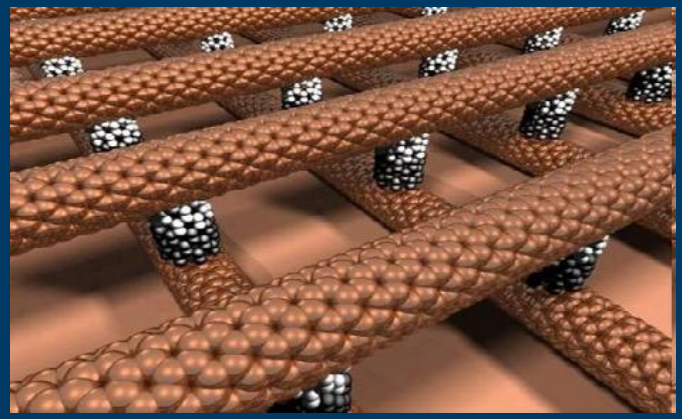
Organizing Any Matter

- Use one kind of programmable matter (e.g. DNA).
- To organize (almost) ANY matter through it.

6 nm grid of individually addressable DNA pixels



PWK Rothemund, *Nature* 440, 297 (2006)



European Nanoelectronics Initiative Advisory Council

"What we are really making are tiny DNA circuit boards that will be used to assemble other components."
Greg Wallraff, IBM

Executing Any Kinetics

- The kinetics of any finite network of chemical reactions, can be implemented (physically) with especially programmed DNA molecules.

- Chemical reactions as an executable programming language for dynamical systems!

DNA as a universal substrate for chemical kinetics

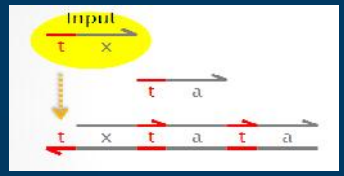
David Soloveichik[a,1], Georg Seelig[a,b,1], and Erik Winfree[c,1]

Building Nano-Control Devices

- All the components of nanocontrollers can already be built entirerly and solely with DNA, and interfaced to the environment

DNA Aptamers

Sensing



DNA Logical Gates

Computing

Constructing

Actuating

Self-assembling DNA Tiles

DNA Walkers & Tweezers

Interfacing to Biology

A doctor in each cell

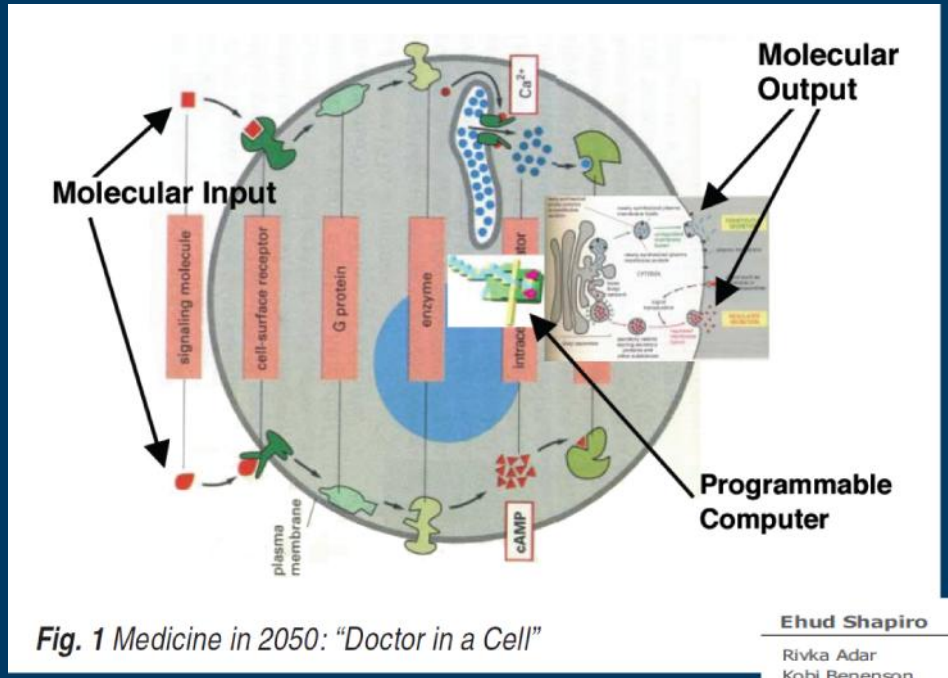


Fig. 1 Medicine in 2050: "Doctor in a Cell"

Ehud Shapiro
Rivka Adar
Kobi Benenson
Gregory Linshitz
Aviv Regev
William Silverman

Molecules and computation

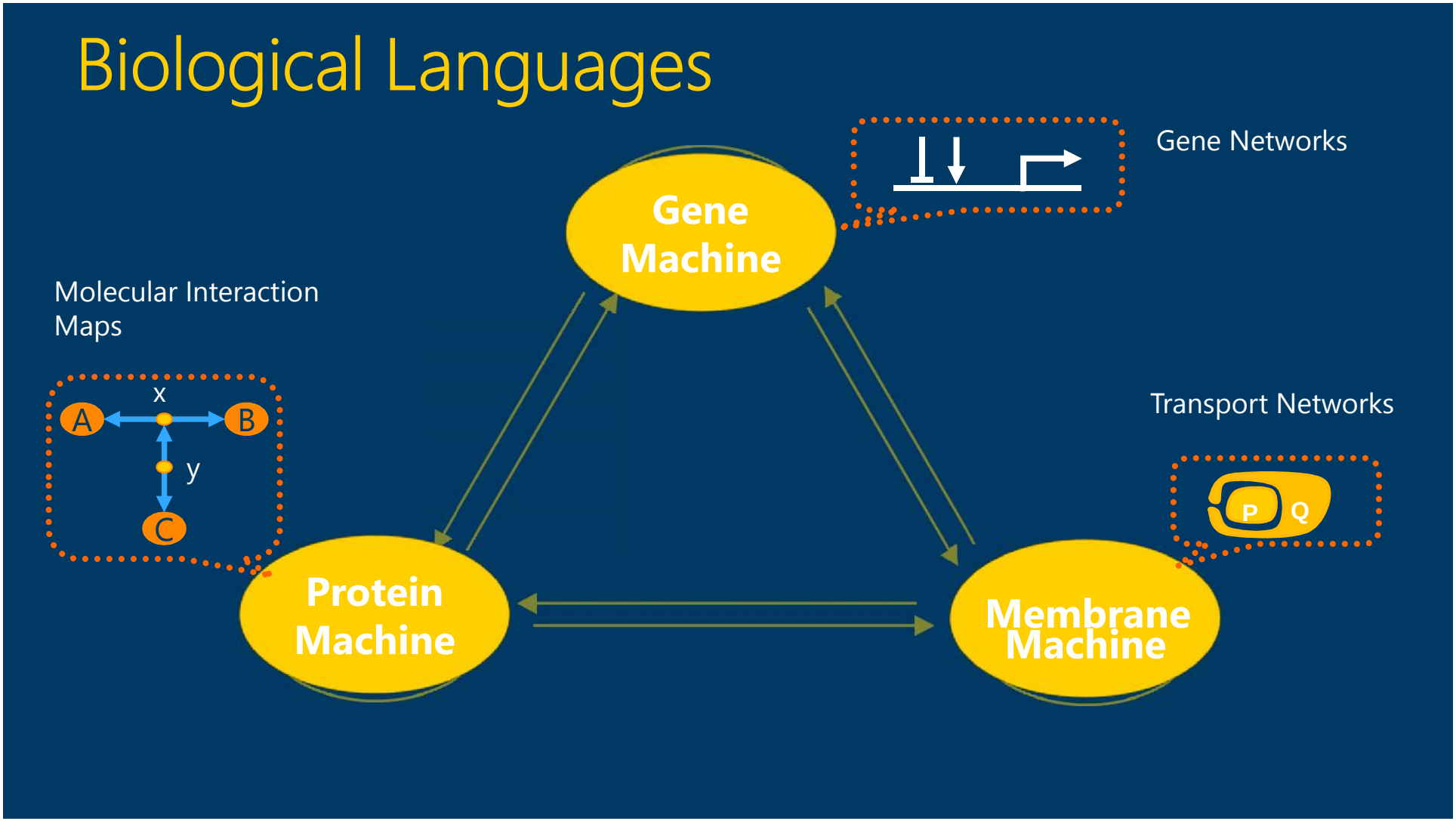
Microsoft Research

Microsoft

The Biological Argument

Biological systems are already ‘molecularly programmed’

Biological Languages



But …

- Biology is programmable, but not by us!
- Still work in progress:
  - Gene networks are being programmed in synthetic biology, but using existing ‘parts’
  - Protein networks are a good candidate, unfortunately we cannot yet effectively design proteins
  - Transport networks are being looked at for programming microfluidic devices manipulating vesicles

Microsoft Research

Microsoft

Molecular Languages

… that we can execute

Action Plan

- Building a full software/hardware pipeline for a new fundamental technology
  - Mathematical Foundations [~ concurrency theory in the 80's]
  - Programming Languages [~ software engineering in the 70's]
  - Analytical Methods and Tools [~ formal methods in the 90's]
  - Device Architecture and Manufacturing [~ electronics in the 60's]
- To realize the potential of Molecular Programming
- This is largely a 'software problem' even when working on device design

The role of DNA Computing

- Non-goals
  - Not to solve NP-complete problems with large vats of DNA
  - Not to replace silicon
- Bootstrapping a carbon-based technology
  - To precisely control the organization and dynamics of matter and information at the molecular level
  - DNA is our engineering material
    - Its biological origin is “accidental” (but convenient)
    - It is an information-bearing programmable material
    - It is possible that other such materials will be developed

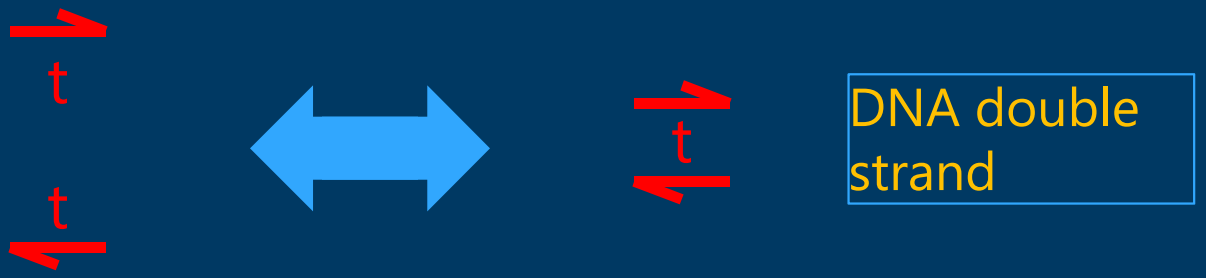
Domains

- Subsequences on a DNA strand are called domains
  - *provided* they are "independent" of each other

oriented DNA single strand

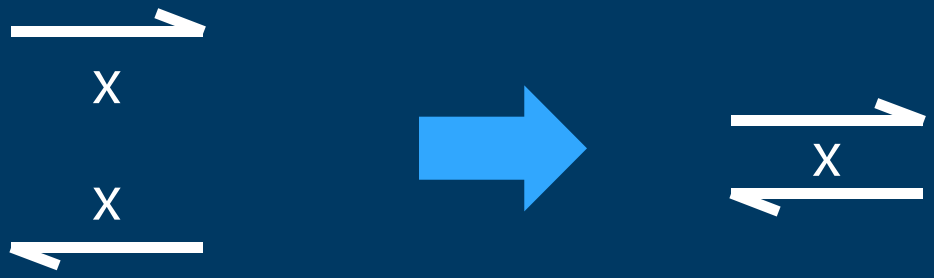
- That is, differently named domains must not hybridize
  - With each other, with each other's complement, with subsequences of each other, with concatenations of other domains (or their complements), etc.

Short Domains



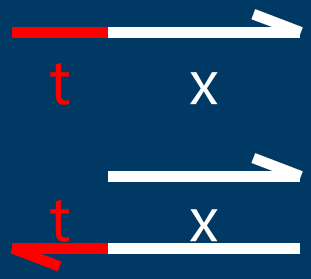
Reversible Hybridization

Long Domains



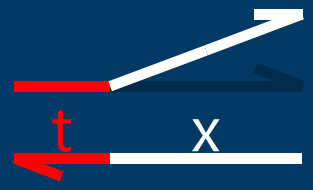
Irreversible Hybridization

Strand Displacement



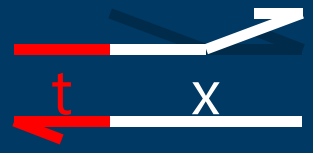
"Toehold Mediated"

Strand Displacement



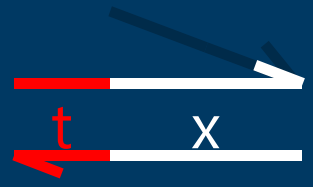
Toehold Binding

Strand Displacement



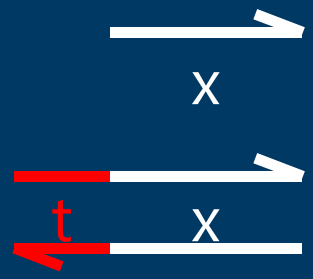
Branch Migration

Strand Displacement



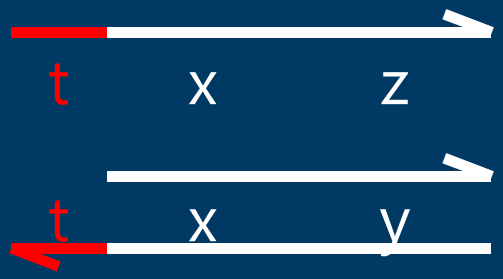
Displacement

Strand Displacement

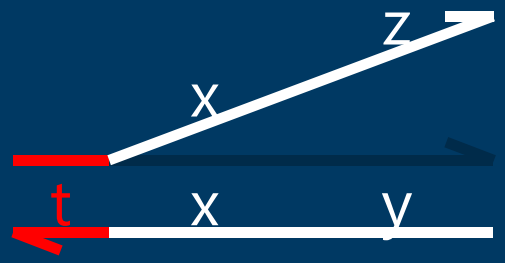


Irreversible release

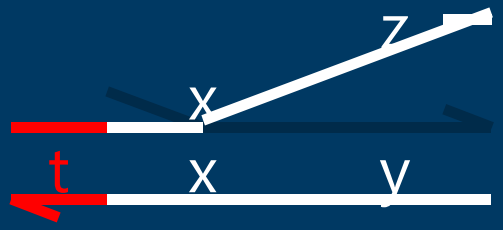
Bad Match



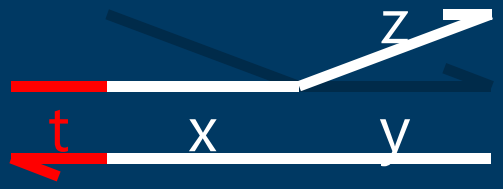
Bad Match



Bad Match



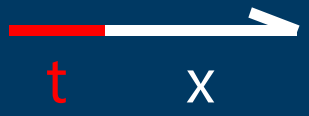
Bad Match



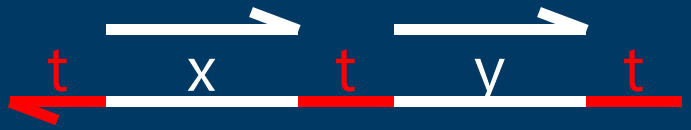
Cannot proceed
Hence will undo

Two-Domain Architecture

- Signals: 1 toehold + 1 recognition region



- Gates: "top-nicked double strands" with open toeholds



Garbage collection
"built into" the gates

Two-Domain DNA Strand Displacement

Luca Cardelli

In S. B. Cooper, E. Kashefi, P. Panangaden (Eds.):
Developments in Computational Models (DCM 2010).
EPTCS 25, 2010, pp. 33-47. May 2010.

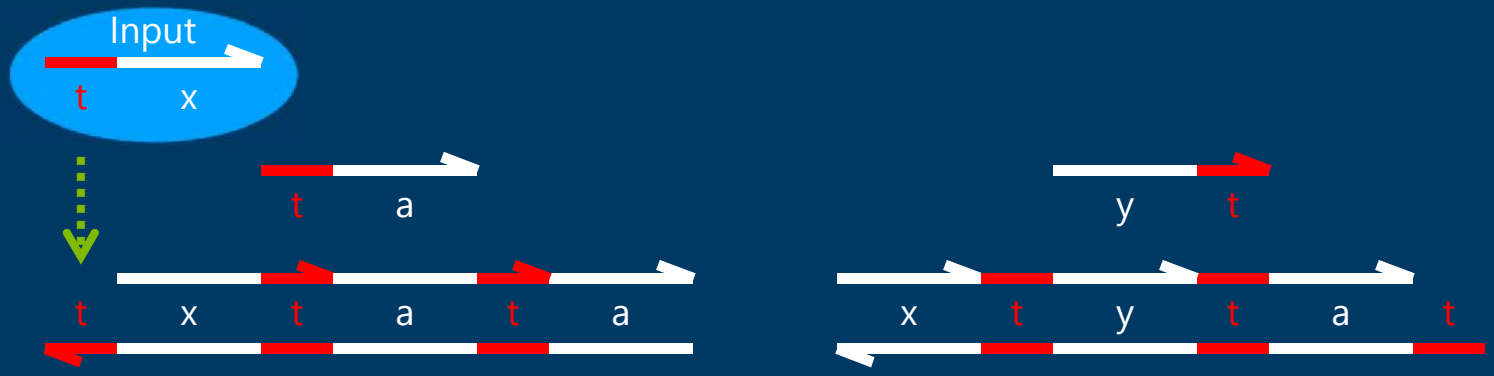
Microsoft Research

Microsoft

Transducer

Transducer x→y

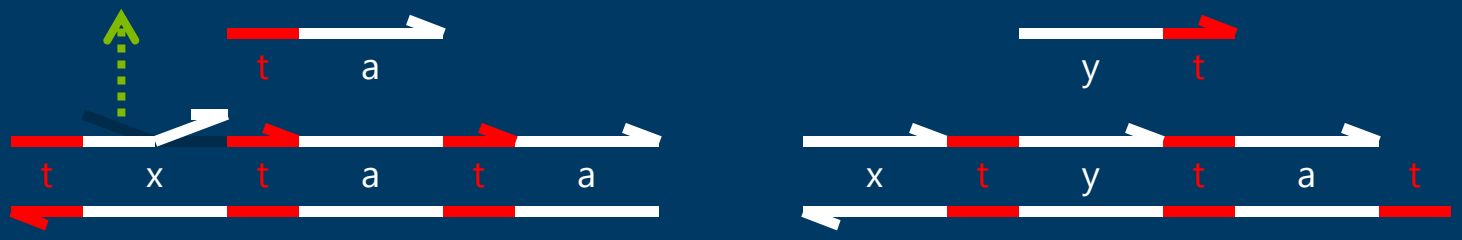
Transducer x→y



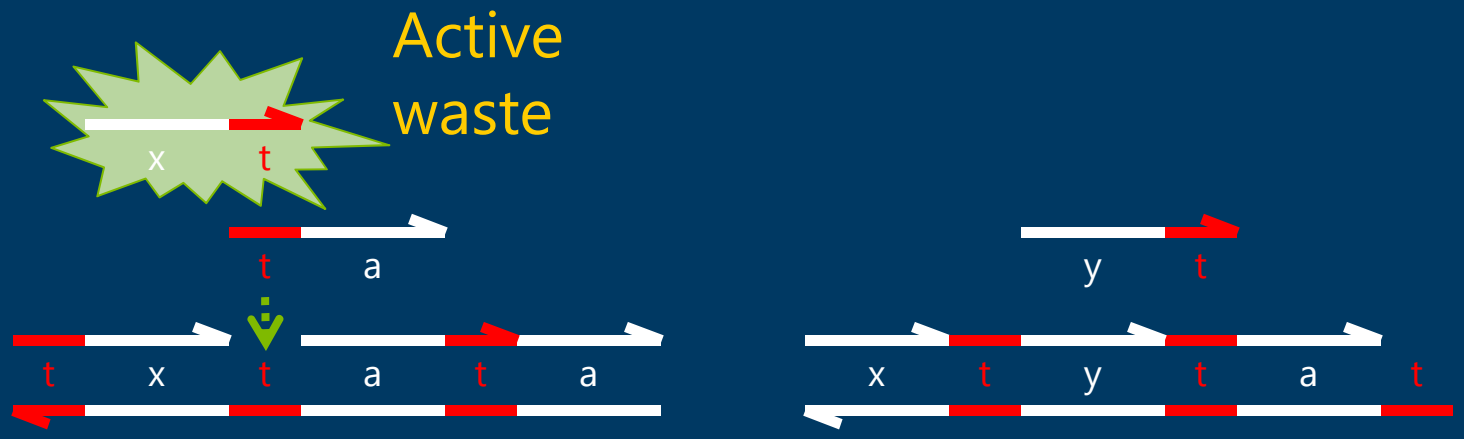
Built by self-assembly!

ta is a *private* signal (a different ‘a’ for each xy pair)

Transducer x→y

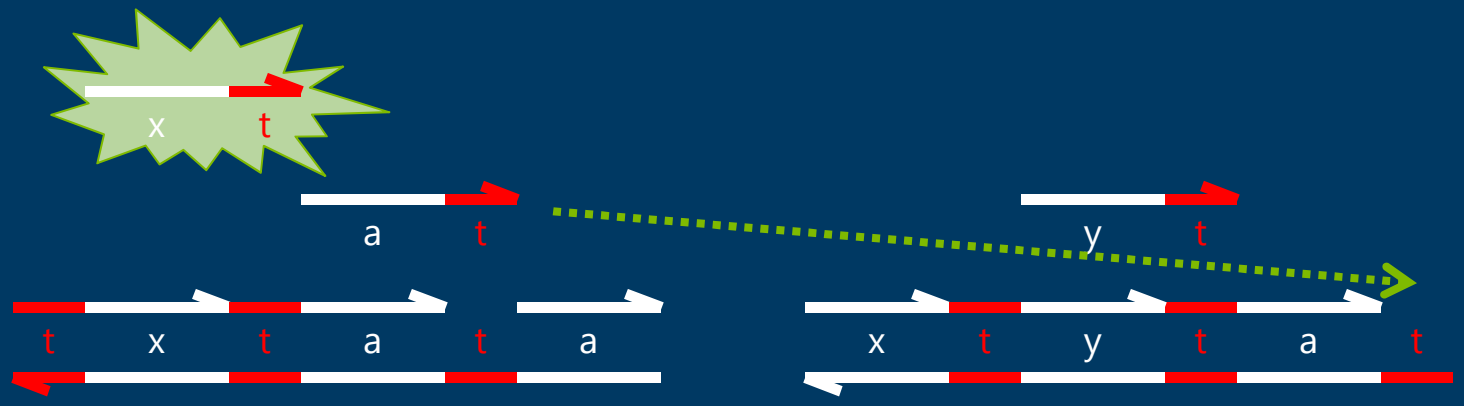


Transducer x→y



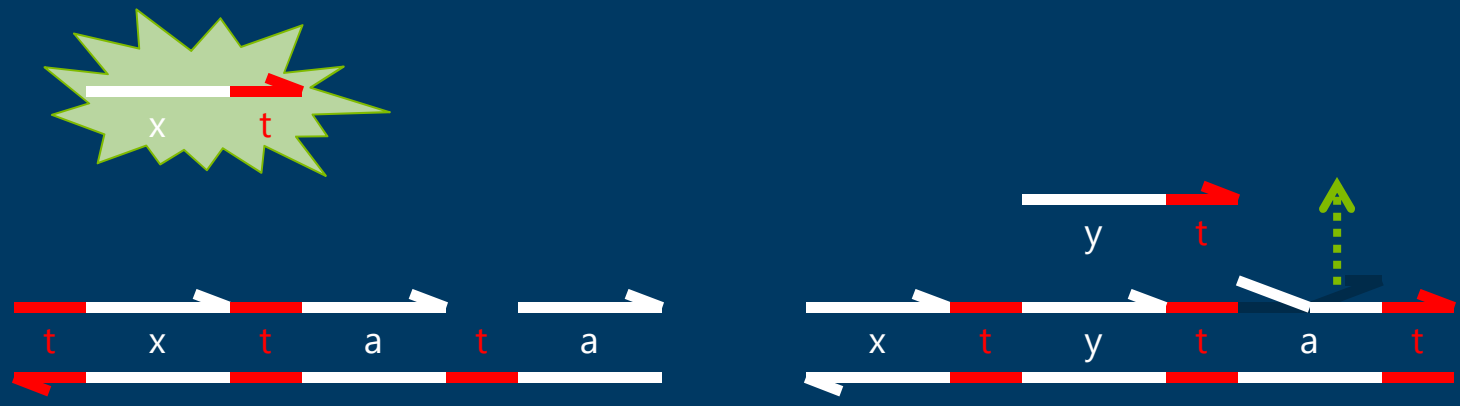
Transducer x→y

Transducer x→y

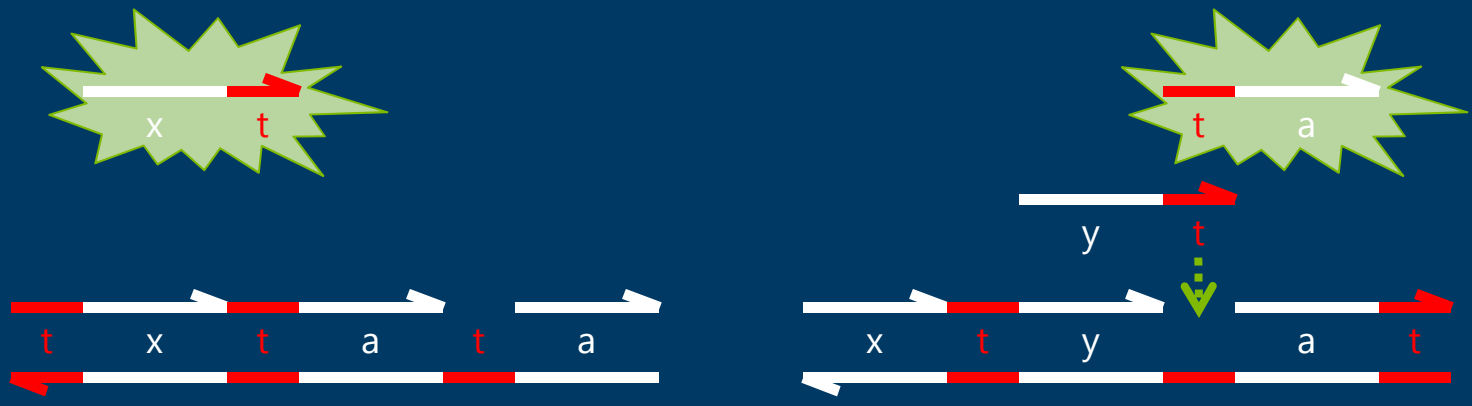


So far, a **tx** *signal* has produced an **at** *cosignal*.
But we want signals as output, not cosignals.

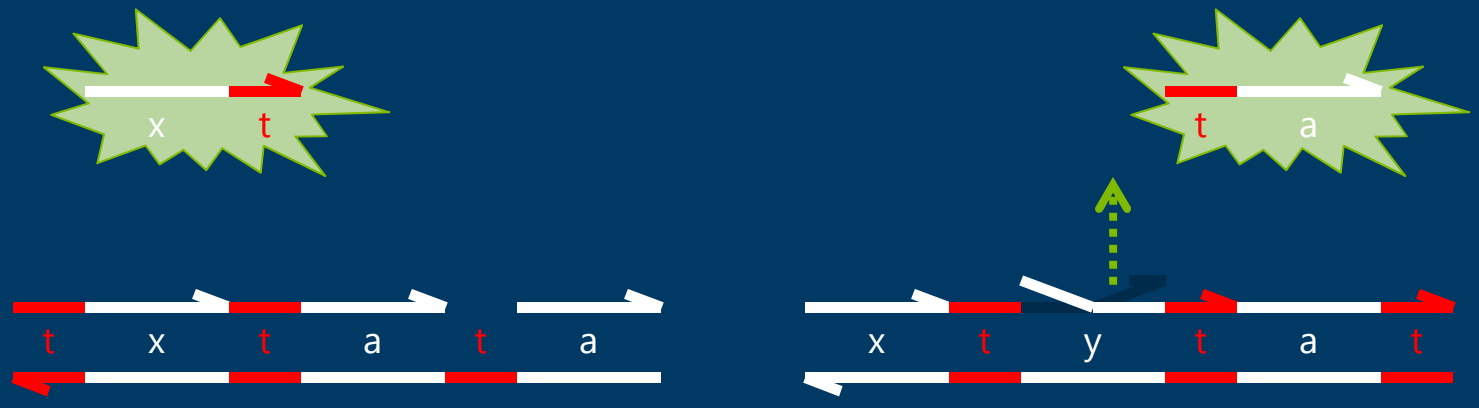
Transducer x→y



Transducer x→y



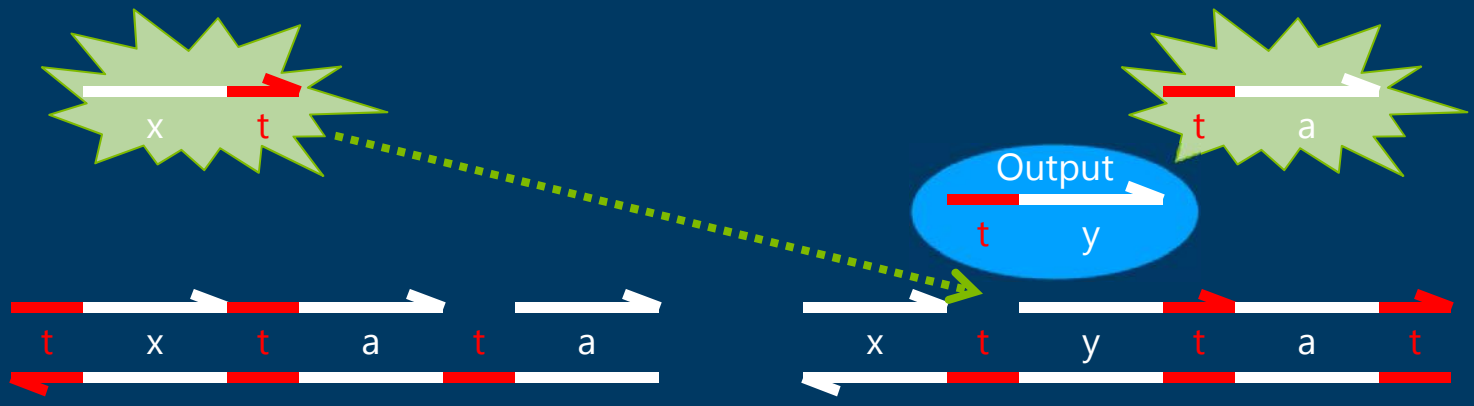
Transducer x→y



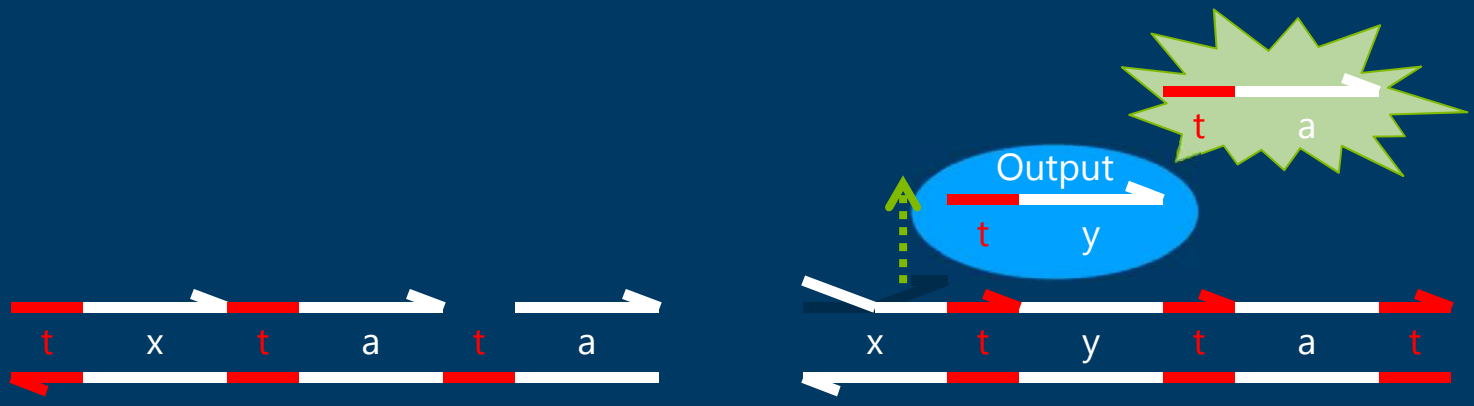
Transducer x→y

Here is our output **ty** *signal*.
But we are not done yet:
1) We need to make the output irreversible.
2) We need to remove the garbage.
We can use (2) to achieve (1).

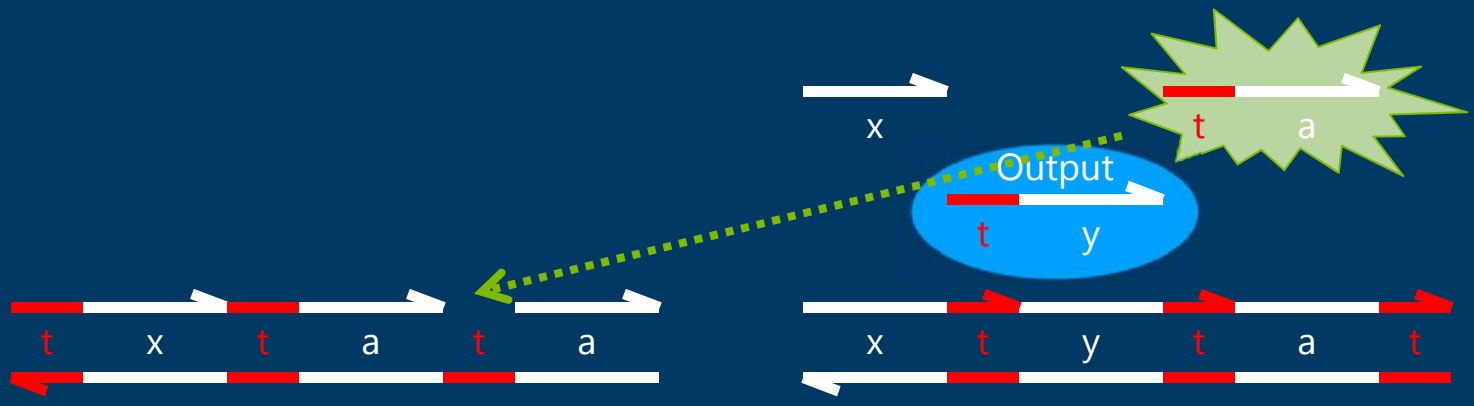
Transducer x→y



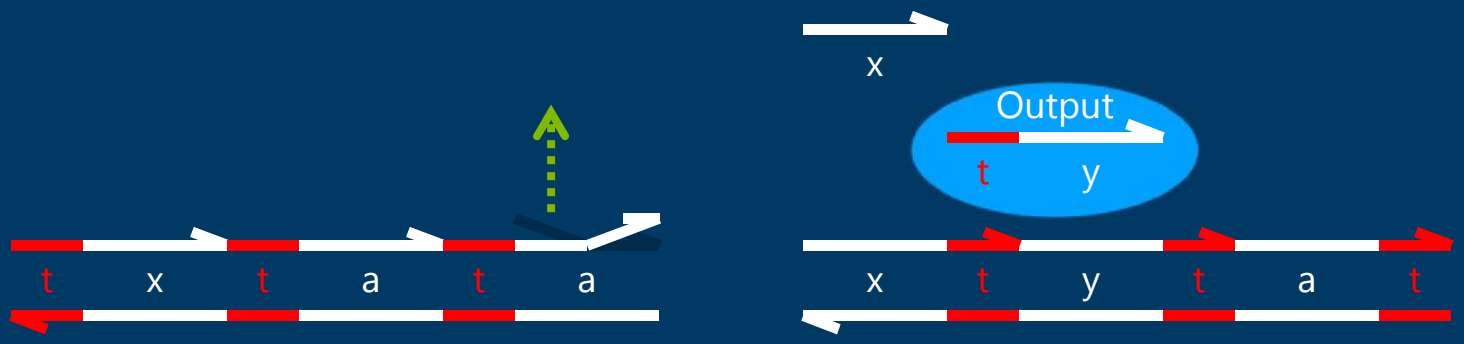
Transducer x→y



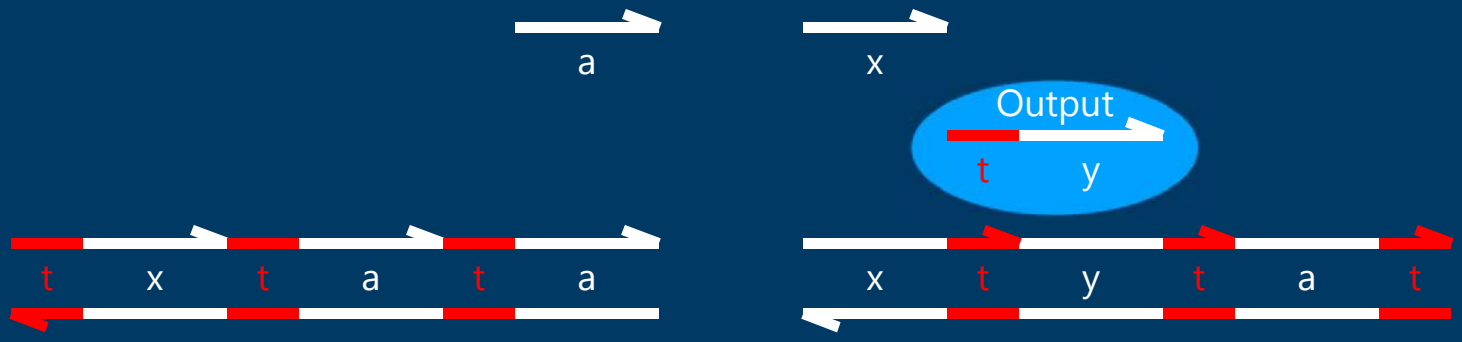
Transducer x→y



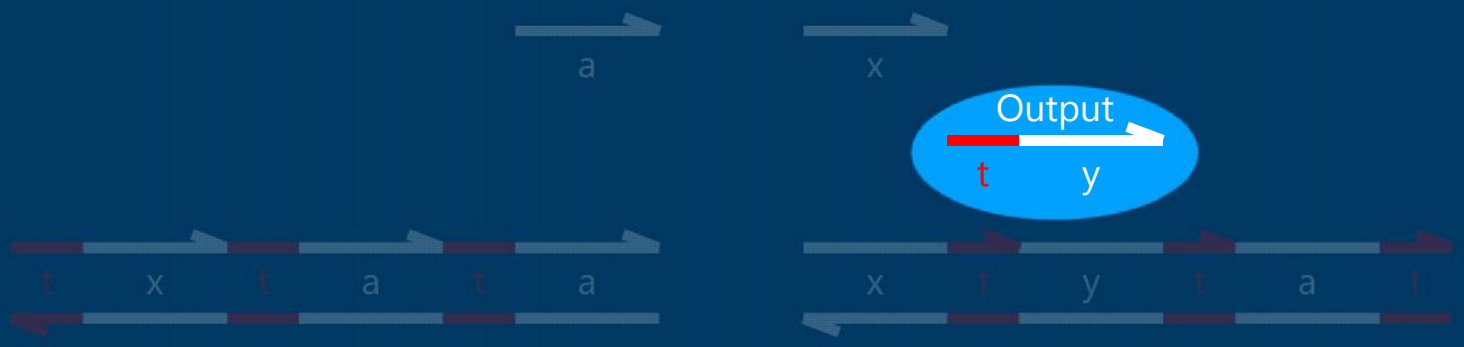
Transducer x→y



Transducer x→y



Transducer x⟶y

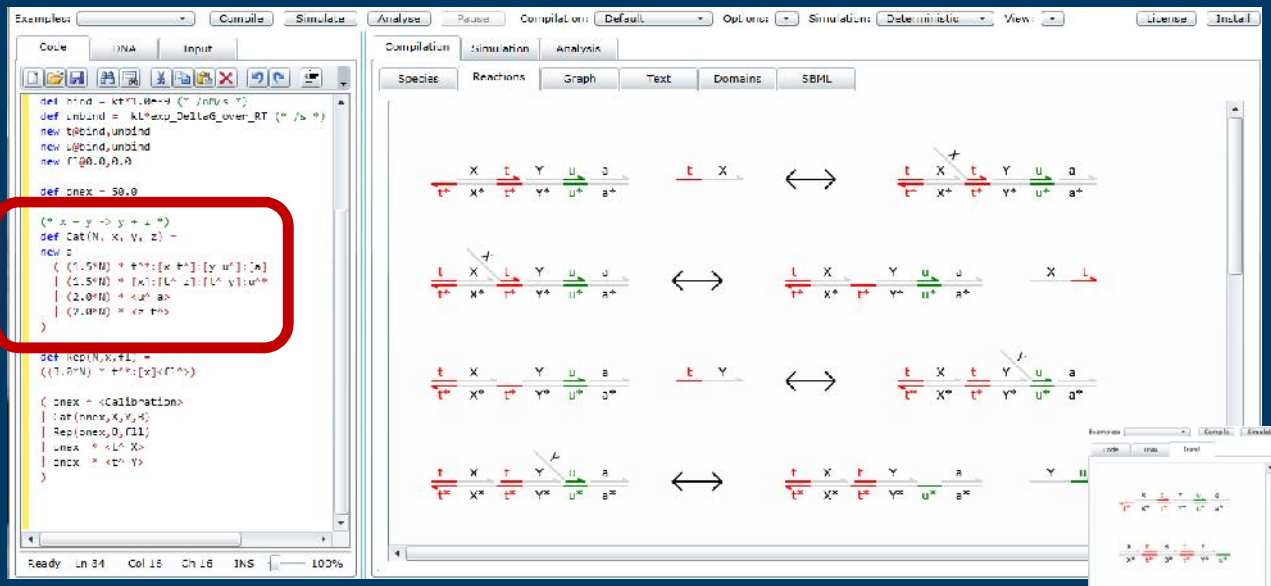


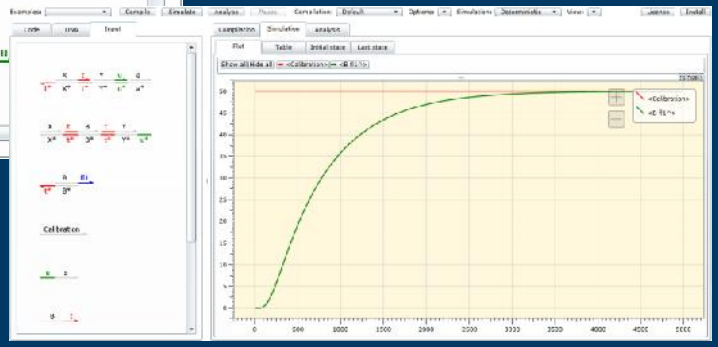
Done.

N.B. the gate is consumed: it is the energy source

Development Tools

MSRC Bio Computation Group





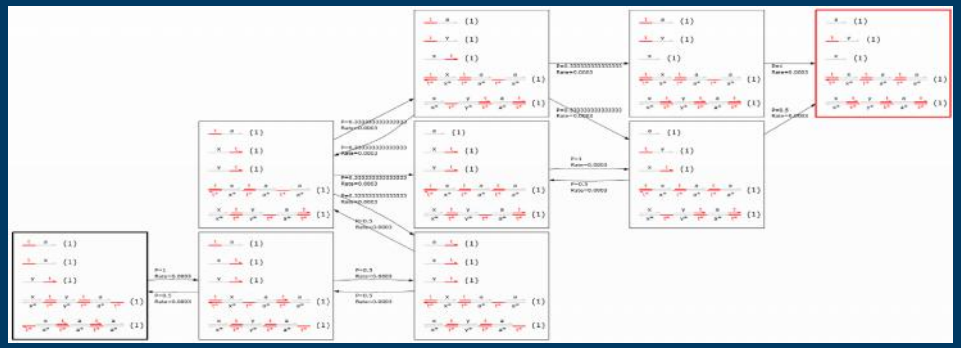
JOURNAL OF THE ROYAL SOCIETY Interface

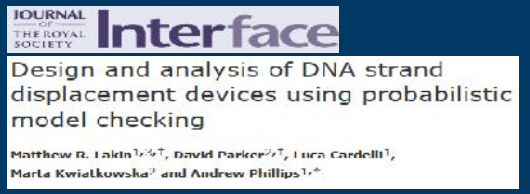
A programming language for composable DNA circuits

Andrew Phillips and Luca Cardelli

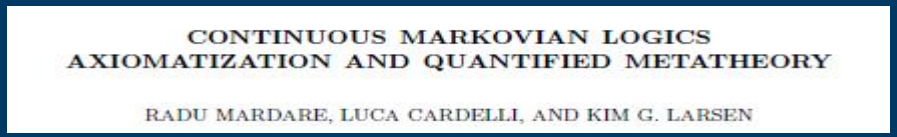
Analytical Methods

- Probabilistic modelchecking (complete state space exploration) to test system correctness.





- Quantitative theories of system equivalence and approximation.

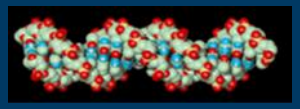


Approximate Majority Algorithm

- Given two populations of agents (or molecules)
  - Randomly communicating by radio (or by collisions)
  - Reach an agreement about which population is in majority
  - By converting all the minority to the majority
    [Angluin et al., Distributed Computing, 2007]
- Could also be used to restore a digital signal to full strength

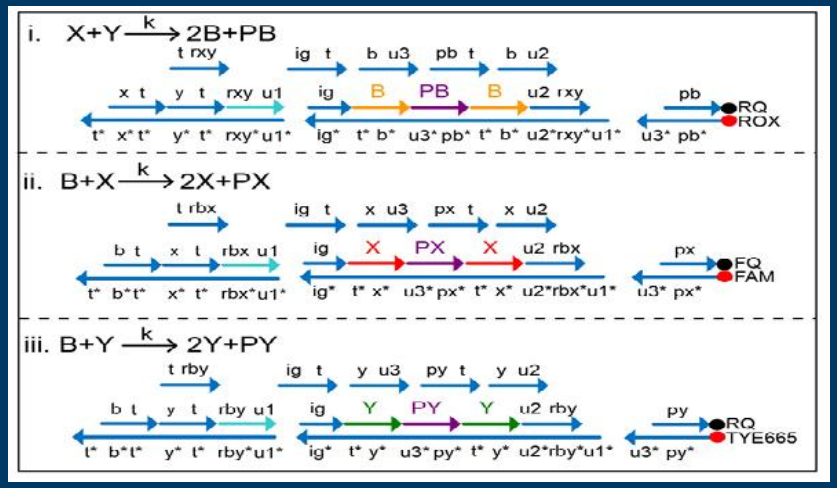
- 3 rules of agent (or molecule) interaction
  - $X + Y \rightarrow B + B$
  - $B + X \rightarrow X + X$
  - $B + Y \rightarrow Y + Y$

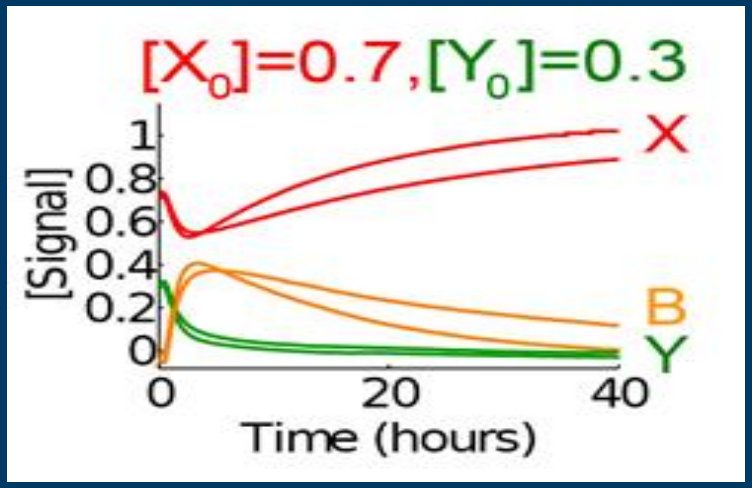
"our program" →



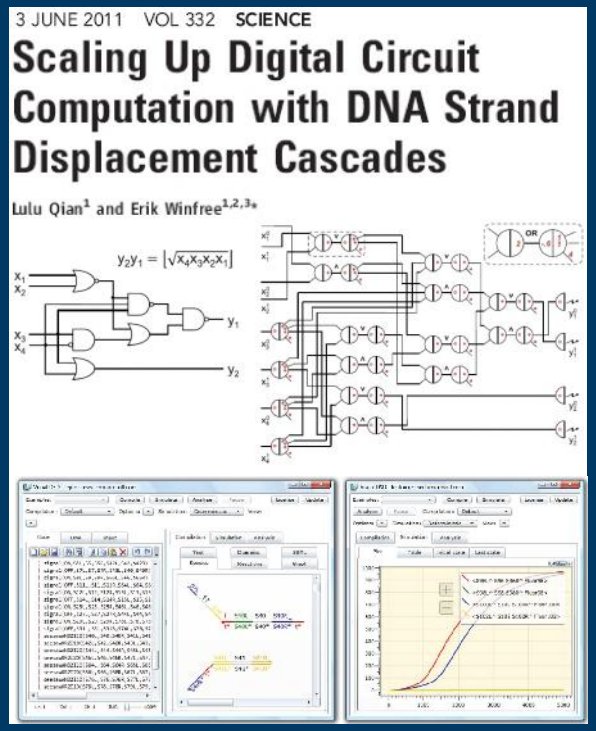
DNA Implementation, at U.W.

- A DNA Realization of Chemical Reaction Networks
  [Yuan-Jyue Chen, Neil Dalchau, Niranjan Srinivas, Andrew Phillips, Luca Cardelli, David Soloveichik and Georg Seelig]

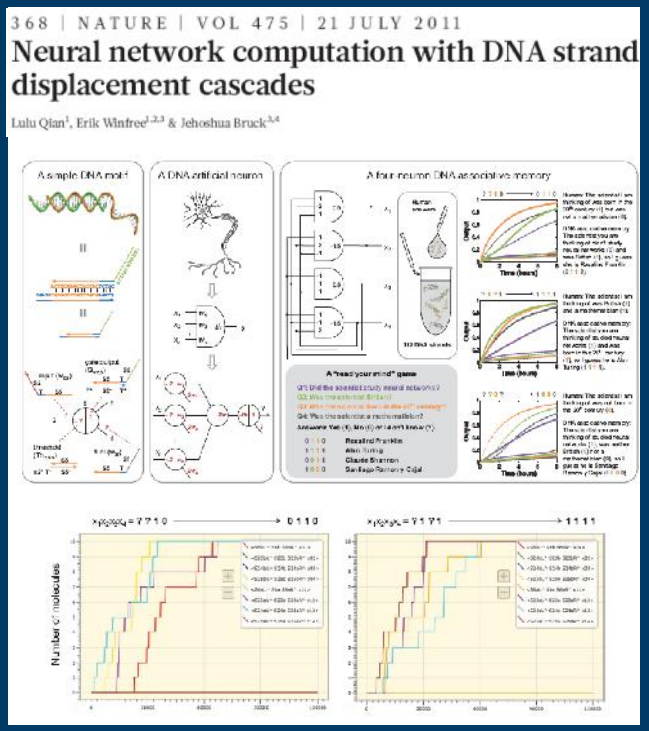




Related Work Supporter by our Tools



Square root of a 4-bit number



Associative memory

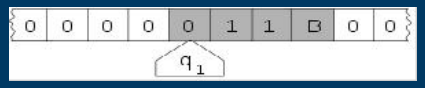
Microsoft Research

Microsoft

Final Remarks

Outlook: A Brief History of DNA

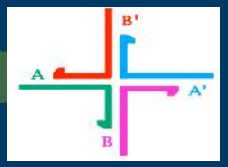
Turing Machine, 1936



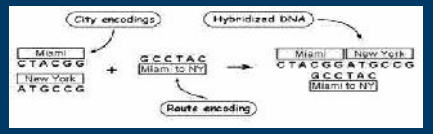
Transistor, 1947

DNA, -3,800,000,000

Structural DNA, 1982



DNA Algorithm, 1994



~~Digital Computers~~

Computer programming

20th century

Software
systematic manipulation of information

<??>
systematic manipulation of matter

~~DNA Computers~~

Molecular programming

21th century

Microsoft