Molecular Programming

Luca Cardelli
Microsoft Research

INRIA Scientific Board, Paris, 2011-11-18
http://lucacardelli.name
Smaller and Smaller

First working transistor

First integrated circuit

50 years later

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

Single molecule transistor

Molecules on a chip

~10 Moore’s Law cycles left!
Building The Smallest Things

• How do we build structures that are by definition smaller than your tools?
• Basic answer: you can’t. Structures (and tools) should build themselves!
• By programmed self-assembly.

www.youtube.com/watch?v=Ey7Emmddf7Y
Molecular IKEA

• Nature can self-assemble. Can we?

• “Dear IKEA, please send me a chest of drawers that assembles itself.”

• We need a magical material where the pieces are pre-programmed to fit into each other.

• At the molecular scale many such materials exist…

Programmed Self-Assembly

Proteins

DNA/RNA

Membranes
Molecular Languages
- modeling languages -
Chemistry

• Chemical reactions
  - \( A + B \rightarrow_r C + D \) (a program)

• Ordinary Differential Equations
  - \( \frac{d[A]}{dt} = -r[A][B] \ldots \) (a semantics)

• Rich analytical techniques based on Calculus

• But prone to combinatorial explosion
  - E.g., due to the peculiarities of protein interactions
High(er)-Level Languages

• Gene Networks
  o Synchronous Boolean networks
    • Stewart Kauffman, etc.
  o Asynchronous Boolean networks
    • René Thomas, etc.

• Protein Networks
  o Process Algebra (stochastic π-calculus etc.)
    • Priami, Regev–Shapiro, etc.
  o Graph Rewriting (kappa, BioNetGen etc.)
    • Danos–Laneve, Fontana & al., etc.

• Membrane Networks
  o Membrane Computing
    • Gheorghe Păun, etc.
  o Brane Calculi
    • Luca Cardelli, etc.
Molecular Languages

- **Reaction-Based** \((A + B \rightarrow C + D)\) (Chemistry)
  - Limited to finite set of species (no polymerization)
  - Practically limited to small number of species (no run-away complexation)

- **Interaction-Based** \((A = !r; C)\) (Process Algebra)
  - Reduces combinatorial complexity of models by combining independent submodels connected by interactions.

- **Rule-Based** \((A{-}\cdot{B}^{p} \rightarrow A^{p}\cdot{B}^{-})\) (Logic, Graph Rewriting)
  - Further reduces model complexity by describing molecular state, and by allowing one to ‘ignore the context’: a rule is a reaction in an unspecified (complexation/phosphorylation) context.
  - Similar to informal descriptions of biochemical events (“narratives”).

- **Syntactic connections**
  - The latter two can be translated (to each other and) to the first, but doing so may introduce an infinite, or anyway extremely large, number of species.
Semantic Connections

These diagrams commute via appropriate maps.

L. Cardelli: “On Process Rate Semantics” (TCS)
L. Cardelli: “A Process Algebra Master Equation” (QEST’07)
But what about Execution?

• Chemistry is not easily executable
  o Please Mr Chemist, execute me these reactions that I just made up.

• Similarly, the molecular languages seen so far are descriptive (modeling) languages

• How can we actually execute molecular languages? With real molecules?
Molecular Languages
- executable languages -
DNA

GC Base Pair
Guanine–Cytosine

TA Base Pair
Thymine–Adenine

Interactive DNA Tutorial
(http://www.biosciences.bham.ac.uk/labs/minchin/tutorials/dna.html)

Sequence of Base Pairs (GACT alphabet)
Robust, and *Long*

- **DNA in each human cell:**
  - 3 billion base pairs
  - 2 meters long, 2nm thick
  - folded into a 6μm ball
  - 750 MegaBytes

- **A huge amount for a cell**
  - Every time a cell replicates it has to copy *2 meters of DNA* reliably.
  - To get a feeling for the scale disparity, compute:

- **DNA in human body**
  - 10 trillion cells
  - 133 Astronomical Units long
  - 7.5 OctaBytes

- **DNA in human population**
  - 20 million light years long

DNA wrapping into chromosomes

Andromeda Galaxy 2.5 million light years away

wehi.edu.au
DNA can support structural and computational complexity.

DNA replication in *real time*

- In Humans: 50 nucleotides/second
- Whole genome in a few hours (with parallel processing)
- In Bacteria: 1000 nucleotides/second (higher error rate)

DNA transcription in *real time*

- RNA polymerase II: 15–30 bases/second

Drew Berry
http://www.wehi.edu.au/wehi-tv
Unnatural DNA Operation

- **Sensing**
  - Reacting to forces
  - Binding to molecules
- **Actuating**
  - Releasing molecules
  - Producing forces
- **Constructing**
  - Chassis
  - Growth
- **Computing**
  - Signal Processing
  - Decision Making

Nucleic Acids can do all this. And interface to biology.
Aptamers: natural or artificially evolved DNA molecules that stick to other molecules (highly selectively).

Adenine riboswitch aptamer
Constructing

Crosslinking

Chengde Mao, Purdue

Andrew Turberfield, Oxford

Folding DNA into Twisted and Curved Nanoscale Shapes
Hendrik Dietz, Shawn M. Douglas, & William M. Shih
Actuating

DNA tweezers

DNA walkers

Bernard Yurke, Boise State
Computing

- Sensors and Actuators at the 'edge' of the system
  - They can use disparate technologies and phenomena

- Computation in the 'kernel' of the system

- Compositionality in the kernel
  - The components should use uniform inputs and outputs
  - The components should be ‘computationally complete’
“Embedded” Computing (Synthetic Biology)

- Using bacterial machinery (e.g.) as the hardware. Using embedded gene networks as the software.

- MIT Registry of Standard Biological Parts

- GenoCAD
  - Meaningful sequences [Cai et al.]

- GEC
  - [Pedersen & Phillips]
“Autonomous” Computing
(Nano-engineering)

• Mix & go
  o All (or most) parts are synthesized
  o No manual cycling (cf. early DNA computing)
  o In some cases, all parts are made of DNA (no enzyme/proteins)

• Self-assembled and self-powered
  o Can run on its own (e.g. environmental sensing)
  o Or be embedded into organisms, but running ‘separately’
Curing

A doctor in each cell

Fig. 1 *Medicine in 2050: “Doctor in a Cell”*
RNA operation in (dead) cells

- Using RNA Hybridization Chain Reaction for imaging of mRNA expression.
  - The programmability of orthogonal RNA reactions enables spatial imaging with 5 simultaneous targets.
Molecular Computation
DNA Computing

• Non–goals
  o Not to solve NP–complete problems.
  o Not to replace electronics.
  o Not necessarily using genes or producing proteins.

• For general ‘molecular programming’
  o To precisely control the organization and dynamics of matter and information at the molecular level.
  o To interact algorithmically with biological entities.
  o The use of DNA is “accidental”: no genes involved.
  o In fact, no material of biological origin.
Domains

• Subsequences on a DNA strand are called **domains**. *Provided* they are “independent” of each other.

• I.e., differently named domains must not hybridize:
  o With each other
  o With each other’s complement
  o With subsequences of each other
  o With concatenations of other domains (or their complements)
  o Etc.

• Choosing domains (subsequences) that are suitably independent is a tricky issue that is still somewhat of an open problem (with a vast literature). But it can work in practice.
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” (or equivalently 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).
Transducer $x \rightarrow y$
Transducer $x \rightarrow y$

$t a$ is a *private* signal (a different ‘a’ for each $xy$ pair)
Transducer $x \rightarrow y$
Transducer $x \rightarrow y$

Active waste

$\text{Active waste}$
Transducer $x \rightarrow y$
Transducer $x \rightarrow y$

So far, a $tx$ signal has produced an $at$ cosignal. But we want signals as output, not cosignals.
Transducer $x \rightarrow y$
Transducer $x \rightarrow y$
Transducer $x \rightarrow y$
Transducer \( x \rightarrow y \)

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 $x \rightarrow y$
Transducer $x \rightarrow y$
Transducer $x \rightarrow y$
Transducer $x \rightarrow y$
Transducer $x \rightarrow y$

Input: $t\ x\ t\ a\ t\ a$

Output: $t\ y$

$x\ t\ y\ t\ a\ t$
Transducer $x \rightarrow y$

Done.

N.B. the gate is consumed: it is the energy source.
Transducer $x \rightarrow y$
Join x+y
General $n \times m$ Join–Fork

• Easily generalized to 2+ inputs (with 1+ collectors).
• Easily generalized to 2+ outputs.

Figure 9: 3-Join $J_{wxyz} \ | \ tw \ | \ tx \ | \ ty \rightarrow tz$: initial state plus inputs $tw, tx, ty$. 
DNA Programming
Experiments

Two-domain gate for $X+Y \rightarrow Y+B$

$X+Y \rightarrow Y+B$

35°C

$1x = 50\text{nM}$

Yuan-Jyue Chen and Georg Seelig
U.Washington.

Y

<table>
<thead>
<tr>
<th>$X+Y \rightarrow Y+B$</th>
<th>Concentration</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG1</td>
<td>1.5x</td>
</tr>
<tr>
<td>LG2</td>
<td>1.5x</td>
</tr>
<tr>
<td>input</td>
<td>1x</td>
</tr>
<tr>
<td>Catalyst</td>
<td>0x, 0.05x, 0.1x, 0.2x, 0.3x, 1x</td>
</tr>
<tr>
<td>$\sim B$</td>
<td>2x</td>
</tr>
<tr>
<td>R1</td>
<td>2x</td>
</tr>
<tr>
<td>B readout</td>
<td>3x</td>
</tr>
</tbody>
</table>

Output (nM) vs. hours

A graph showing the reaction kinetics with different concentrations of $Y$.
Verification
Verification Issues

• Environment
  o The nano-environment is messy (stochastic noise, failures, etc.)
  o But we should at least ensure our designs are *logically correct*  

• Verifying Components
  o Reversible reactions (infinite traces)
  o Interferences (deadlocks etc.) between copies of the same gate
  o Interferences (deadlocks etc.) between copies of different gates
  o Removal of active byproducts (garbage collection) is tricky

• Verifying Populations
  o Gates come in (large) populations
  o Each population *shares private domains* (technologically unavoidable)
  o Correctness of populations means proofs with large state spaces
Correctness

• The spec of a transducer:

\[ x . y \mid x \rightarrow y \]

- Is it true at all?
- Is it true possibly, necessarily, or probabilistically?
- Is it true in the context of a population of identical transducers?
- Is it true in all possible contexts?
- If false, does it become true for infinite populations?
Interfering Transducers

- Let a be the private transducer domain, but let’s share it between x.y and y.x

- Interference: \( x.a \, y \mid y.a \, x \mid x \not\rightarrow \forall \, x \)

- But still: \( x.a \, y \mid y.a \, x \mid x \mid y \rightarrow \forall \, x \mid y \)

- A large population of such gates in practice does not deadlock easily.

- The wisdom of crowds: individuals can be wrong, but the population is all right.
Modelchecking DNA Systems

- Using the PRISM stochastic modelchecker
  - Termination probability of interfering transducers $x \mid x.a\ y \mid y.a\ z$

Design and Analysis of DNA Circuits using Probabilistic Model Checking.
http://qav.comlab.ox.ac.uk/papers/dna-pmc.pdf. September 2010
Conclusions
A Brief History of DNA

Turing Machine, 1936

Transistor, 1947

Digital Computers

Computer programming

Software
systematic manipulation of information
20th century

DNA, ~3,800,000,000

Matterware

Matterware??

DNA Computers

Molecular programming

DNA Algorithm, 1994

Structural DNA, 1982
Acknowledgments

• Microsoft Research
  o Andrew Phillips

• Caltech
  o Winfree Lab

• U.Washington
  o Seelig Lab