Natural Computing

Lecture 17

Michael Herrmann
mherrman@inf.ed.ac.uk
phone: 0131 6 517177
Informatics Forum 1.42

15/11/2011

Membrane Computing

Perspectives

- Nanotechnology: Self-assembly, micromanipulation
- Synthetic biology
- Swarm intelligence in nano-robots
- Computing using designed molecules to carry out elementary computations
- Computation on surfaces, gels, networks
- Membrane computing
- Quantum computing: first universal 2-qubit quantum computer in 2009 (79% accuracy), 3 qubit (2010)

Natural Computing: A Formal Perspective

- Studying of models of computation inspired by biological systems
- Some approaches in Natural Computing use the methods of formal language theory
  - L systems: Development of multicellular organisms (plants), (Aristid Lindenmayer, 1968)
  - Cellular Automata (Stephen Wolfram, 1983; based on work by v. Neumann, Hedlund, Conway et al.)
  - H systems: DNA (Tom Head, 1987)
  - P systems: Membranes (Gheorghe Păun, 1998)

L Systems

- An L system can be defined as

$$G = (V; \omega; P)$$

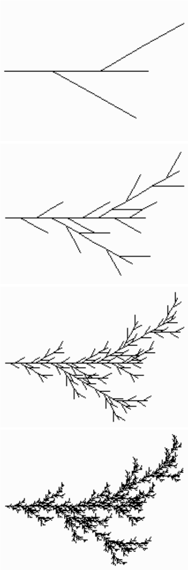
  - V is an alphabet
  - $\omega \in V^*$ is the initial state of the system
  - P is a finite set of rules $a \rightarrow v$ with $a \in V$ and $v \in V^*$(rules are applied simultaneously)

- Total parallelism: All symbols of a string processed at the same time
- Example $G = (\{a\}, a, \{a \rightarrow aa\})$ generates the language $\{a^{2^n} | n \geq 1\}$ which is not context-free (due to parallelism!)
- Languages generated by L Systems are recognized by Systolic Automata (not context-free)

L Systems: Fractal plant

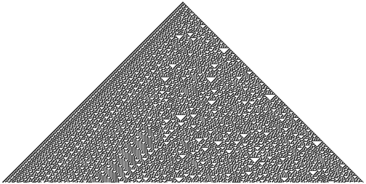
- Variables : X (meta symbol) F (draw forward)
- Constants : $+$ $-$ (turn left/right with angle 30^o)
- Start : F
- Rules : $(F \rightarrow F[-FF]F[+FF]F)$
- [save current position and angle
-] restore values saved at corresponding [

http://en.wikipedia.org/wiki/L-system
pics: http://www.jcu.edu/math/vignettes/lsystems.htm



Cellular Automata

- Grid of cells: 1D or 2D ...
- Each cell assumes a state (*n*-ary, $n \geq 2$)
- State changes in discrete time in dependence of a finite number of neighbors
- Every cell has the same rule for updating
- Examples:
  - Sierpinski triangle (Rule 60)
  - Conway's Game of Life



Rule 30

111	110	101	100	011	010	001	000
0	0	0	1	1	1	1	0

http://en.wikipedia.org/wiki/File:CA_rule30s.png

H Systems

- "Splicing systems" (existed already before Adleman 1994)
- Inspired by DNA reproduction (crossover action of restriction proteins)
- Crossing over operations assume the place of rewrite rules
- H systems can be defined as

$$H = (V; A; R)$$

  - V: alphabet
  - $A \subseteq V^*$ initial language
  - R: splicing rules $u_i,\ x_i,\ y_i\ \in V^*$

$$u_1 \# u_2 \$ u_3 \# u_4 :\ (x_1 u_1 u_2 x_2,\ y_1 u_3 u_4 y_2) \rightarrow\ x_1 u_1 u_4 y_2$$

- Turing complete (even without mutation)

Membrane Computing

- Cells have a usually a large number of compartments hosting a huge variety of biochemical reactions
- Membrane Computing is a generalization of DNA computing: Within different regions of space different but not unrelated computations can be performed.
- Functions of membranes in the cell
  - Separators between compartments
  - Channels for communication between compartments
- Biologically inspired, but a computational rather than a biological model

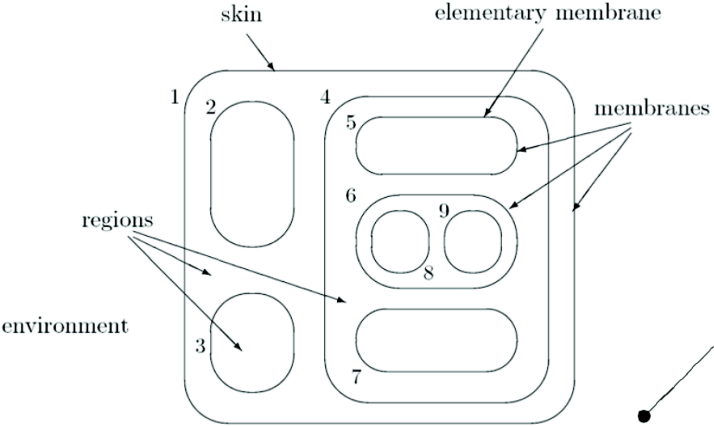
Gheorghe Păun: Computing with Membranes, (1998)

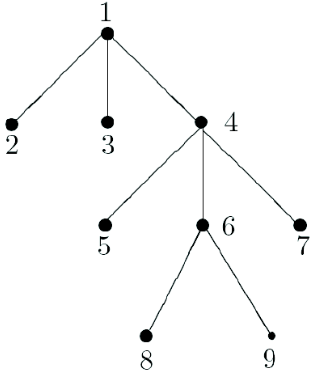
Membrane Computing

- An area that seeks to discover new computational models from the study of the cellular membranes.
- It not so much the task of creating a cellular model but to derive a computational mechanism from processes that are know to proceed in a cell.
- Deals with distributed and parallel computing models, processing multisets of symbol objects
- The various types of membrane systems have been formalized as P systems.

Gheorghe Păun: Introduction to Membrane Computing

(citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.76.8425&rep=rep1&type=pdf)
Păun showed also that splicing systems (or H systems) are computationally universal, i.e. proved an important fact in DNA computing.





$$[_1\ [_2\ ]_2\ [_3\ ]_3\ [_4\ [_5\ ]_5\ [_6\ [_8\ ]_8\ [_9\ ]_9\ ]_6\ [_7\ ]_7\ ]_4\ ]_1$$

$$=$$

$$[_1\ [_3\ ]_3\ [_4\ [_6\ [_8\ ]_8\ [_9\ ]_9\ ]_6\ [_7\ ]_7\ [_5\ ]_5\ ]_4\ [_2\ ]_2\ ]_1$$

P Systems

- A membrane structure formed by several membranes embedded in a unique main membrane (skin)
- Multisets of objects placed inside the regions delimited by the membranes (one per each region)
- The objects are represented as symbols of a given alphabet (each symbol denotes a different object)
- Sets of evolution rules associated with the regions (one per each region), which allow the system
  - to produce new objects starting from existing ones
  - to move objects from one region to another

P Systems: Formal Definition

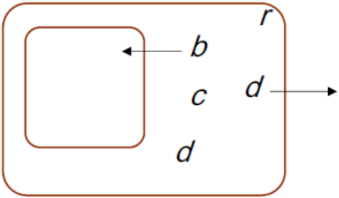
- A P System Π is given by

$$\Pi = (V, C, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, i_o)$$

- V: alphabet (elements are called objects)
- $C \in V$: catalysts
- $\mu \subset \mathbb{N} \times \mathbb{N}$: membrane structure, such that $(i, j) \in \mu$ denotes that membrane j is contained in membrane i
- $w_i \in V^*$ $(1 \leq i \leq n)$: multiset of objects inside membrane i
- R_i $(1 \leq i \leq n)$: evolution rule inside membrane i
- i_o: output region

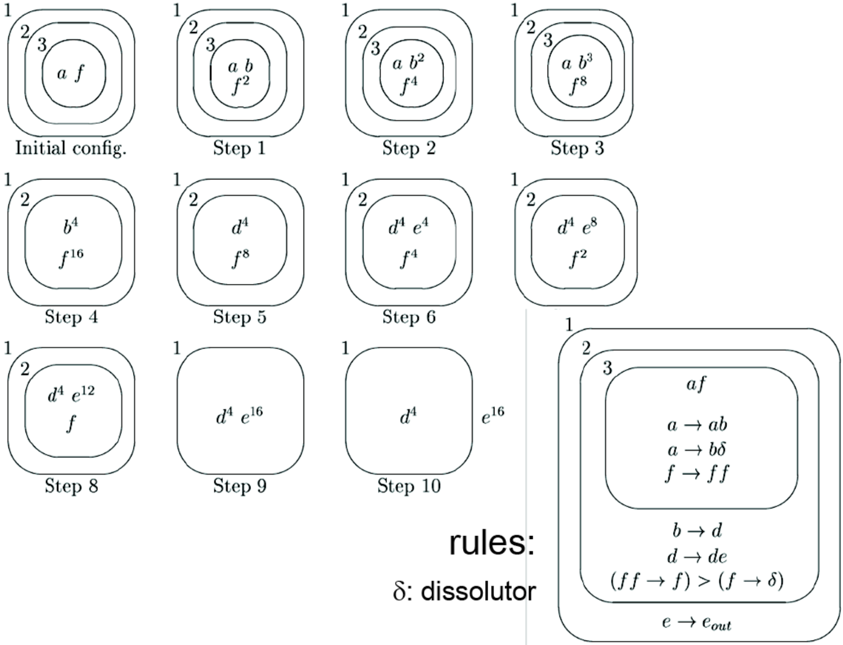
P systems

- Evolution rule of region
  $r : ca \to cb_{in} d_{out} d_{here}$
- "a copy of object a in the presence of a copy of the catalyst c is replaced by a copy of the object b and 2 copies of the object d" and
- "b enters the inner membrane of region r" and
- "one copy of d leaves region r" and
- "one copy of d remains in r."



Computation in a P-system

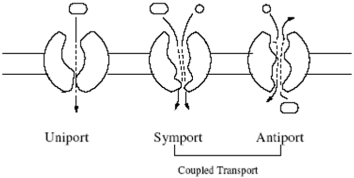
- Initial configuration: Construct a membrane structure and place an initial multiset of objects inside the regions of the system.
- Apply the rules in a nondeterministic parallel manner: in each step, in each region, each object can be evolved according to some rule
- Halting if a configuration is reached where no rules can be applied
- The result is the multisets formed by the objects contained in a specific output membrane
- A non-halting computation yields no result
- Example: Assume the environment is reduced to some specific input objects. Now if the system halts in a final configuration, the system has "recognized"this input.



Initial config.
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Step 8
Step 9
Step 10
$a\ f$
$a\ b\ f^2$
$a\ b^2\ f^4$
$a\ b^3\ f^8$
$b^4\ f^{16}$
$d^4\ f^8$
$d^4\ e^4\ f^4$
$d^4\ e^8\ f^2$
$d^4\ e^{12}\ f$
$d^4\ e^{16}$
d^4
e^{16}
rules:
δ: dissolutor
af
$a \to ab$
$a \to b\delta$
$f \to ff$
$b \to d$
$d \to de$
$(ff \to f) > (f \to \delta)$
$e \to e_{out}$

More operators

- Membrane dissolution
- Priorities, reaction rates
- Catalysts
- Bi-stable catalysts
- Membrane permeability
- Active membranes: creation, deletion, duplication



- *P*-systems with catalysts are computationally universal
- *P*-systems with symport/antiport (communicative *P*-systems) are computationally universal – even without chemical reactions)

Maximal Parallelism

Evolution rules are applied with maximal parallelism:

- More than one rule can be applied (on different objects) in the same step
- Each rule can be applied more than once in the same step (on different objects)
- Maximality means that:
  - A multiset of instances of evolution rules is chosen nondeterministically such that no other rule can be applied to the system obtained by removing all the objects necessary to apply the chosen instances of rules.

From a course of Andrea Maggiolo Schettini http://www.di.unipi.it/~maggiolo/Corso_Macao/

Types of "Membrane" Systems

1. Cell-like P systems: membranes hierarchically arranged
2. Tissue-like P systems: nodes are associated with the cells
3. Neural-like P systems
4. Population P-systems: Networks of evolutionary processes

Spiking Neural P Systems

Neural networks can be expressed as P systems:

- Only one object: the symbol denoting a spike
- One-membrane cells (called neurons) which can hold any number of spikes
- Each neuron fires in specified conditions (after collecting a specified number of spikes): sends one spike along its axon
- The spike passes to all neurons connected by a synapse to the spiking neuron (replicated into as many copies as many target neurons exist);
- One of the neurons is considered the output one, and its spikes provide the output of the computation.

Spiking neural P systems are universal

M. Ionescu et al.: Spiking neural P systems. Journal Fundamenta Informaticae archive 71:2,3, 2006.

Natural computing with P systems

- Complex dynamics of a biophysical system
- Parallel by nature
- Largely nondeterministic
- Encoding/readout problems must be solved in applications to practical problems
- Computation works well in many problems but may be ineffective or inefficient on some problems
- In natural computing, good solutions emerge or are discovered rather than being designed, although capabilities of design are presently improving in a very impressive way.

Use of X-systems ($X \in \{P, L, H\}$)

- Understanding computing in nature
- Formulating behavioural equivalence
- Planning experiments

"Computing is a Natural Science"

- Information processes abundant in nature
- Wiener (1958) "Cybernetics is the science of communication and control, whether in machines or living organisms."
- Ken Wilson: Computing as a third leg of science (joining theory and experiment)
  - tools (beginning in the 1940s)
  - methods (beginning in the 1980s)
  - fundamental processes (beginning in the 2000s)
- Computation is a sequence of representations, in which each transition is controlled by a representation (Peter J. Denning)
- Information and computation are being discovered as fundamental processes in many fields. Computing is no longer a science of just the artificial. It is the study of information processes, natural and artificial ($\rightarrow$ informatics).

Peter J. Denning, Communications of the ACM, July 2007, Vol. 50, No. 7

Natural Computing

- Inspired or realized by the complex dynamics of a biophysical system
- Parallel by nature
- Nondeterministic
- Encoding problems are crucial in applications to practical problems
- Computation works well in many problems but may be ineffective or inefficient on some problems
- In natural computing, good solutions emerge rather than being designed

Conclusion

- Nature is the major source of inspiration
  - Natural phenomena can be translated into computing paradigms
  - Natural processes can be (and are) used as carriers of computational operations
- In addition to electronic phenomena other physical effects are prospectively useful for computation
- Specific problems deserve specific solutions, less specified problems require more general approaches

Final lecture will be on: Quantum computing (brief introduction)