BASICS OF MEMBRANE COMPUTING

Nestine Hope S. Hernandez

Algorithms and Complexity Laboratory
Department of Computer Science
University of the Philippines, Diliman
nshernandez@dcs.upd.edu.ph

Cell-inspired Computing: An Introduction and Overview of Membrane Computing
A Seminar on Membrane Computing
Palawan State University, Puerto Princesa, Palawan
15 March 2016
Outline

Overview of Algorithms and the Limits of Computation

Membrane Computing
  How it started
  The Basics
  Types of P Systems by membrane structure
“Computer Science is no more about computers than astronomy is about telescopes”

- Edsger Dijkstra
“[Computer science] is not really about computers – and it’s not about computers in the same sense that physics is not really about particle accelerators, and biology is not about microscopes and Petri dishes...and geometry isn’t really about using surveying instruments. Now the reason that we think computer science is about computers is pretty much the same reason that the Egyptians thought geometry was about surveying instruments: when some field is just getting started and you don’t really understand it very well, it’s very easy to confuse the essence of what you’re doing with the tools that you use.”

- Hal Abelson (1986) Introduction of video of lectures on the Structure and Interpretation of Computer Programs
Then...
Overview of Algorithms and the Limits of Computation

Then...

Now...
What Computers Can’t Do

---

What Computers Can’t Do\textsuperscript{1}

\textsuperscript{1}Reference: Lecture Slides on Explorations in Computing: An Introduction to Computer Science, by John S. Conery.
What Computers Can’t Do

What Computers Can’t Do\textsuperscript{1}

- there are an estimated $10^{43}$ possible games
- a supercomputer checking $10^{12}$ boards/sec would need $10^{21}$ years to look at them all

\textsuperscript{1}Reference: Lecture Slides on Explorations in Computing: An Introduction to Computer Science, by John S. Conery.
What Computers Can’t Do

Computation

A computation is a sequence of well-defined operations that lead from an initial starting point to a desired final outcome.
Computation

A computation is a sequence of well-defined operations that lead from an initial starting point to a desired final outcome.

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sanchez, Maria</td>
<td>Feb 14, 1988</td>
<td>20</td>
</tr>
<tr>
<td>Sanders, Eric</td>
<td>Mar 24, 1978</td>
<td>30</td>
</tr>
<tr>
<td>Sato, Noriko</td>
<td>Oct 14, 1989</td>
<td>18</td>
</tr>
<tr>
<td>Singer, Fred</td>
<td>Apr 30, 1983</td>
<td>25</td>
</tr>
<tr>
<td>Smith, John</td>
<td>Feb 26, 1990</td>
<td>18</td>
</tr>
</tbody>
</table>

Average (Mean) Age: \( \frac{20 + 30 + 18 + 25 + 18}{5} = 22.2 \)
A computation is a sequence of well-defined operations that lead from an initial starting point to a desired final outcome.

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Age</th>
<th>Average (Mean) Age:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sanchez, Maria</td>
<td>Feb 14, 1988</td>
<td>20</td>
<td>(\frac{20 + 30 + 18 + 25 + 18}{5} = 22.2)</td>
</tr>
<tr>
<td>Sanders, Eric</td>
<td>Mar 24, 1978</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Sato, Noriko</td>
<td>Oct 14, 1989</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Singer, Fred</td>
<td>Apr 30, 1983</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Smith, John</td>
<td>Feb 26, 1990</td>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

The sequence of steps carried out during a computation are defined by an algorithm.
Algorithms

- The earliest known algorithms were defined by Greek mathematicians
  - e.g. Euclid’s method for the greatest common divisor of two integers, ca. 300 BC

- The modern word “algorithm” comes from the name of the Persian scholar Muḥammad ibn Mūsā al-Ḵwārizmī (ca. 780 -- ca. 850)
  - when his work was published in Latin his name was spelled Algoritmi
  - he was the author of several influential works on mathematics and natural science
  - his book on the systematic solution of linear equations contained several algorithms
  - the title of this book is also the source of our word “algebra”
Algorithms

Informally: step-by-step, simple, mechanical procedure
Turing Machine

- Any problem with an algorithm can be solved using a Turing machine.
Limits of Turing Machine

- Some Unsolvable Problems
  - Halting Problem: Given a computer program, does it always halt?
  - Hilbert’s 10th Problem: Given an equation, does it have a solution?
  - Post Correspondence Problem
- Some Intractable Problems
  - Traveling Salesman Problem
  - Boolean Satisfiability Problem
  - Graph Coloring Problem

\[\text{Reference: An Introduction to Membrane Computing (Y4IT2012), R.A.Juayong.}\]
Traveling Salesman Problem

Find a route that passes through all cities exactly once and returns to the starting city.
Traveling Salesman Problem

Find a route that passes through all cities exactly once and returns to the starting city.
## On Complexity

<table>
<thead>
<tr>
<th>n</th>
<th>f(n)</th>
<th>lg n</th>
<th>n</th>
<th>n lg n</th>
<th>n²</th>
<th>2^n</th>
<th>n!</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td>0.003 μs</td>
<td>0.01 μs</td>
<td>0.033 μs</td>
<td>0.1 μs</td>
<td>1 μs</td>
<td>3.63 ms</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0.004 μs</td>
<td>0.02 μs</td>
<td>0.086 μs</td>
<td>0.4 μs</td>
<td>1 ms</td>
<td>77.1 years</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>0.005 μs</td>
<td>0.03 μs</td>
<td>0.147 μs</td>
<td>0.9 μs</td>
<td>1 sec</td>
<td>8.4 × 10^15 yrs</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>0.005 μs</td>
<td>0.04 μs</td>
<td>0.213 μs</td>
<td>1.6 μs</td>
<td>18.3 min</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>0.006 μs</td>
<td>0.05 μs</td>
<td>0.282 μs</td>
<td>2.5 μs</td>
<td>13 days</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>0.007 μs</td>
<td>0.1 μs</td>
<td>0.644 μs</td>
<td>10 μs</td>
<td>4 × 10^13 yrs</td>
<td></td>
</tr>
<tr>
<td>1,000</td>
<td></td>
<td>0.010 μs</td>
<td>1.00 μs</td>
<td>9.966 μs</td>
<td>1 ms</td>
<td>100 ms</td>
<td></td>
</tr>
<tr>
<td>10,000</td>
<td></td>
<td>0.013 μs</td>
<td>10 μs</td>
<td>130 μs</td>
<td>10 sec</td>
<td>16.7 min</td>
<td></td>
</tr>
<tr>
<td>100,000</td>
<td></td>
<td>0.017 μs</td>
<td>0.10 ms</td>
<td>1.67 ms</td>
<td>1.16 days</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,000,000</td>
<td></td>
<td>0.020 μs</td>
<td>1 ms</td>
<td>19.93 ms</td>
<td>115.7 days</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10,000,000</td>
<td></td>
<td>0.023 μs</td>
<td>0.01 sec</td>
<td>0.23 sec</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100,000,000</td>
<td></td>
<td>0.027 μs</td>
<td>0.10 sec</td>
<td>2.66 sec</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,000,000,000</td>
<td></td>
<td>0.030 μs</td>
<td>1 sec</td>
<td>29.90 sec</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• Can we explore a different perspective on computing?
On Natural Computing

- Networking
- OS
- Natural computing
- ...
The natural computing paradigm is a branch of computing that houses biologically-inspired computational ideas, tools, techniques and theoretical models.
Some Bio-inspired Computing Models

Reference: Membrane Computing: Power, Efficiency and Applications by G. Paun
On Membrane Computing

- Introduced by Gh. Păun in 1998 through the paper entitled "Computing with Membranes"
- Computing models from membrane computing are called P systems
- Membrane computing is a branch of the natural computing paradigm which abstract features from living cells in order to devise new ways of computing.
The Inspiration
The Inspiration

WHAT IS A CELL? (for a mathematician)

- membranes, separating “inside” from “outside” (hence protected compartments, “reactors”)

- chemicals in solution (hence multisets)

- biochemistry (hence parallelism, nondeterminism, decentralization)

- enzymatic activity/control

- selective passage of chemicals across membranes

- etc.
Main Ingredients
Main Ingredients\(^3\)
Main Ingredients

1. \( a \rightarrow b \)
2. \( b \rightarrow t \)
3. \( t \rightarrow t' \delta \)
4. \( t \rightarrow t \)
5. \( c \rightarrow b \)
6. \( a \rightarrow a_{in}b_{out} \)

\( ab \rightarrow dd_{out}e_{in} \)
Evolution Rules

- General form: $u \rightarrow v$, where $u$ and $v$ are strings representing multisets of objects
Evolution Rules

• General form: $u \rightarrow v$, where $u$ and $v$ are strings representing multisets of objects

• when $|u| = 1$ we say the rule is noncooperative, while a rule with $|u| > 1$ is cooperative
Evolution Rules

• General form: \( u \rightarrow v \), where \( u \) and \( v \) are strings representing multisets of objects

• when \(|u| = 1\) we say the rule is noncooperative, while a rule with \(|u| > 1\) is cooperative
Communication Rules

- General form: \( u \rightarrow (a, tar) \), where \( u \) and \( a \) are strings representing multisets of objects and \( tar \) is either \textit{here}, \textit{out} or \textit{in}_j
Communication Rules

- General form: $u \rightarrow (a, tar)$, where $u$ and $a$ are strings representing multisets of objects and $tar$ is either here, out or $in_j$
Communication Rules

- General form: $u \rightarrow (a, tar)$, where $u$ and $a$ are strings representing multisets of objects and $tar$ is either here, out or in $j$. 
and more...
Rule Application

- nondeterministic
- maximally parallel
Rule Application

- nondeterministic
- maximally parallel
An Example

1 2
3

\[
\begin{align*}
    cd \\
    a & \rightarrow \delta \\
    a & \rightarrow (a, in_3) \\
    ac & \rightarrow \delta
\end{align*}
\]

4

\[
\begin{align*}
    c & \rightarrow (d, out) \\
    b & \rightarrow b
\end{align*}
\]

\[
\begin{align*}
    c & \rightarrow (c, in_4) \\
    a & \rightarrow (a, in_2)b \\
    dd & \rightarrow (a, in_4)
\end{align*}
\]
An Example

\[
\begin{align*}
3 \quad & a \cd & a \rightarrow \delta \\
& c \rightarrow (a, \text{in}_3) & ac \rightarrow \delta \\
& c \rightarrow (c, \text{in}_4) & a \rightarrow (a, \text{in}_2)b \\
& dd \rightarrow (a, \text{in}_4) & 
\end{align*}
\]
An Example

What if we have this instead?

```
3
  cd
  a → δ
  a → (a, in_3)
  ac
  ac → δ
```

```
4
  c → (d, out)
  b → b
```

```
12
  ac
  c → (c, in_4)
  a → (a, in_2)b
  dd → (a, in_4)
```
• cell-like (a hierarchical arrangement of membranes)
• cell-like (a hierarchical arrangement of membranes)
- cell-like (a hierarchical arrangement of membranes)

- tissue-like (several one-membrane cells evolving in a common environment)
• cell-like (a hierarchical arrangement of membranes)

• tissue-like (several one-membrane cells evolving in a common environment)
• cell-like (a hierarchical arrangement of membranes)

• tissue-like (several one-membrane cells evolving in a common environment)

• neural-like (cells(neurons) are placed in the nodes of a general graph)
- cell-like (a hierarchical arrangement of membranes)

- tissue-like (several one-membrane cells evolving in a common environment)

- neural-like (cells (neurons) are placed in the nodes of a general graph)
TO BE CONTINUED ...
Questions?

Thank you for listening!