MGS: a declarative spatial computing programming language

Jean-Louis Giavitto
Antoine Spicher

a IRCAM – CNRS
b LACL – Université de Paris Est

http://mgs.spa,al-compu,ng.org

We sketch the rationals of the MGS programming language. MGS is an experimental programming language developed to study the application of several spatial computing concepts to the specification and simulation of dynamical systems with a dynamical structure. MGS extends the notion of rewriting by considering more general structure than terms. The basic computation step in MGS replaces in a topological collection \(A\), some subcollection \(B\), by another topological collection \(C\). A topological collection is a set of element structured by a neighborhood relationships describing an underlying space representing a data structure or constraints that must be fulfilled by the computation. This process proposes a unified view on several computational mechanisms initially inspired by biological or chemical processes (Gamma and the CHAM, Lindenmayer systems, Paun systems and cellular automata).
DS
Gamma P systems L systems cellular automata
Spatial generalization
MGS
Algorithmic examples
Biological modeling
Dynamical systems
and
Dynamical Structures
Specifying a dynamical system for simulation

- Structure of state
- Structure of time
- Evolution function

\[ \text{state}_{t-1} \xrightarrow{H} \text{state}_t \xrightarrow{H} \text{state}_{t+1} \]

\[ \text{state}_{t-dt} \xrightarrow{H} \text{state}_t \xrightarrow{H} \text{state}_{t+dt} \]

\[ H^* \]

\[ \int H \, dt \]

\[ \text{N} \rightarrow \text{R} \]
Formalism for Dynamical System

- State often structured by space e.g. fields
- Time
- Evolution function

<table>
<thead>
<tr>
<th></th>
<th>PDE</th>
<th>Coupled ODE</th>
<th>Iteration of functions</th>
<th>Cellular automata</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>D</td>
<td>...</td>
</tr>
<tr>
<td>Time</td>
<td>C</td>
<td>C</td>
<td>D</td>
<td>D</td>
<td>...</td>
</tr>
<tr>
<td>Space</td>
<td>C</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>...</td>
</tr>
</tbody>
</table>
The medium process problem

THE CHEMICAL BASIS OF MORPHOGENESIS

By A. M. TURING, F.R.S. University of Manchester

(Received 9 November 1951—Revised 15 March 1952)

A dynamical system DS
THE CHEMICAL BASIS OF MORPHOGENESIS

By A. M. TURING, F.R.S. University of Manchester

(Received 9 November 1951—Revised 15 March 1952)

A dynamical system (DS) with a dynamical structure $(DS)^2$
Bio-inspired models of (DS)^2
Cellular Automata

Von Neumann
Avoids the dynamic structure problems
Predefined underlying unbounded space
Replace a cell in an NEWS grid by another one with a new state
Lindenmayer systems

The structure of a tree can be coded by a string of parenthetised symbols

A symbol is an elementary part of the plant

The symbol between [ ] and ] represents a sub tree

Additional conventions are used to represent main axis orientation depth etc

A rule

$$s \rightarrow s \ s \ s$$

represents the evolution of s
Replace a substring in a string by another one

e / (C(e) & (e.x >= lm) & (e.p == L))
=> {type="C", a=e.a, h=e.h, x=e.x*shorter, p=L},
   {type="C", a=e.a, h=e.h, x=e.x*longer}
Replace a sub multiset in a multiset by another one
A general device
A general device

A general rewriting mechanism

In a collection of elements

With a collection Y computed from X and its

<table>
<thead>
<tr>
<th>Collection</th>
<th>Neighborhood</th>
<th>data structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term</td>
<td>father, son</td>
<td>free</td>
</tr>
<tr>
<td>Set</td>
<td>all</td>
<td>A, C, Idempotent</td>
</tr>
<tr>
<td>Multiset</td>
<td>all</td>
<td>A, Commutative</td>
</tr>
<tr>
<td>Sequence</td>
<td>left and right</td>
<td>with jump for trees</td>
</tr>
<tr>
<td>Commutative</td>
<td></td>
<td>Associative</td>
</tr>
<tr>
<td>Grid</td>
<td>NEWS</td>
<td></td>
</tr>
</tbody>
</table>
Rewriting systems and abstract transition systems

Rewriting system

- Used to formalize equationnal reasoning
- A generative device: grammar
- Replace a sub part of an entity by another
- Set of rewriting rules $\alpha \rightarrow \beta$
  - $\alpha$: pattern specifying a sub part
  - $\beta$: expression evaluating a new sub part

Example: arithmetic expressions simplification
Rewriting on terms property

→ arithmetic term rewriting

a b → string rewriting L systems

H O → H O multiset rewriting chemistry

string concatenation is a formal associative operation

multiset concatenation the chemical soup is AC
A general device

A general rewriting mechanism

In a collection of elements

\[ a \quad X \]

With a collection \( Y \) computed from \( X \) and its

<table>
<thead>
<tr>
<th>Collection</th>
<th>Neighborhood</th>
<th>data structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term</td>
<td>father, son</td>
<td>free</td>
</tr>
<tr>
<td>Set</td>
<td>all</td>
<td>A, C, Idempotent</td>
</tr>
<tr>
<td>Multiset</td>
<td>all</td>
<td>A, Commutative</td>
</tr>
<tr>
<td>Sequence</td>
<td>left and right, with jump for trees</td>
<td>Associative</td>
</tr>
<tr>
<td>Commutative</td>
<td>NEWS</td>
<td></td>
</tr>
</tbody>
</table>
The Spatial Approach

Use space topology to unify the various collection structures
- space as a resource
- space as a constraint
- space as an input output

Neighborhood relationships
- the structure of the collection
- the structure of the subcollection
- the computation dependencies

Substitution replacement
topological surgery
Properties

mandatory when you cannot express a global function relation because the domain of the function relation is changing in time

interaction based approach
the l h s of a rule specifies a set of elements in the r h s the result of the interaction

a generative process enumerates the elements but membership test can be very hard

for the same set of rules

on the rules or on the derivation e.g. growth function in L system
MGS Proposition

Topological collections

Structure

A collection of topological cells

An \textit{incidence relationship}
MGS Proposition

Topological collections

Structure
  A collection of topological cells
  An incidence relationship

Data  association of a value with each cell
Abstract Simplicial Complex and Simplicial Chains

boundary $f$  
$\begin{align*}
  v & v & v & e & e & e \\
  \end{align*}$

faces $f$  
$\begin{align*}
  e & e & e & e \\
  \end{align*}$

cofaces $v$  
$\begin{align*}
  e & e & e \\
  \end{align*}$

coordinates with vertices

lengths with edges

area with $f$

$$
\begin{pmatrix}
0 \\
4 \\
\end{pmatrix}v_1 + \begin{pmatrix}
3 \\
0 \\
\end{pmatrix}v_2 + \begin{pmatrix}
-3 \\
0 \\
\end{pmatrix}v_3 + 5e_1 + 6e_2 + 5e_3 + 12f
$$
Topology ↔ simplicial complex ↔ lattice

Topology  open and closed sets

Simplicial Complex  set of sets closed by inclusion intersection

Lattice  order relation ∧, ∨

• Concise reformulation of classical approaches
• Extension
Transformations
Functions defined by case on collections
Each case pattern matches a sub collection
Defining a rewriting relationship topological rewriting

\[
\text{trans } T = \{ \negmedspace \begin{aligned}
\text{pattern}_1 & \Rightarrow \text{expression}_1 \\
\text{...} & \\
\text{pattern}_n & \Rightarrow \text{expression}_n \\
\end{aligned} \}
\]
MGS Proposition

Transformations

\[
\text{trans } T = \{ \text{pattern}_1 \Rightarrow \text{expression}_1, \ldots, \text{pattern}_n \Rightarrow \text{expression}_n \}\]
Topological rewriting transformation

- Arithmetic term rewriting
  - Arithmetic operation

- String rewriting
  - String concatenation
  - Is a formal associative operation

- Multiset rewriting
  - Multiset concatenation
  - The chemical soup
  - Is AC

- Topological rewriting
  - Gluing cell in a cell complex
  - AC and algebraic machinery
Pattern matching: specifying a sub-collection of elements in interaction

- **path**: sequence of neighbor elements
- **patch**: arbitrary shape

Concise but limited expressiveness

Longer but higher expressiveness
**Example**  
Diffusion Limited Aggregation (DLA)

Diffusion: some particles are randomly diffusing; others are fixed.

Aggregation: if a mobile particle meets a fixed one, it stays fixed.

```plaintext
trans dla = {
   `mobile , `fixed => `fixed, `fixed ;
   `mobile , <undef> => <undef>, `mobile
}
```

NEIGHBOR OF

![Diagram showing diffusion and aggregation processes](image-url)
Example Diffusion Limited Aggregation DLA

Diffusion some particles are randomly diffusing others are fixed
Aggregation if a mobile particle meets a fixed one it stays fixed

\[
\text{trans dla} = \{
\text{mobile, fixed} \Rightarrow \text{fixed, fixed};
\text{mobile, undef} \Rightarrow \text{undef}, \text{mobile}
\}
\]

this transformation is an abstract process that can be applied to any kind of space
Polytypisme
Algorithmic examples
Bead Sort
Bead Sort
Gbf NEWS = < North, South, East, West; 
   North+South=0, East+West=0>

trans dla = {
   `bead | south> `empty  =>  `empty, `bead ;
}

Eratosthene’s Sieve

\[
\begin{align*}
\text{trans } \text{Generate} &= \{x, \text{true}\} \Rightarrow x, \{x + 1, \text{true}\}; \\
\text{trans } \text{Succed} &= \{x, \text{true}\} \Rightarrow x; \\
\text{trans } \text{Eliminate} &= (x, y \mid y \mod x = 0) \Rightarrow x;
\end{align*}
\]

\[
\text{Eliminate}[^{\text{fixrule}}] \left( \text{Succed} \left( \text{Generate}^N \left( \{2, \text{true}\}, \text{set : (}) \right) \right) \right)
\]
Eratosthenes’s Sieve
Eratosthene’s Sieve

\[
\text{trans } Eratos = \{
\begin{align*}
\text{Genere1} & = \quad n : \text{integer} \sslash \text{right} n \Rightarrow n, \{\text{prime} = n\}; \\
\text{Genere2} & = \quad n : \text{integer}, \{\text{prime as } x, \text{~candidate, ~ok}\} \\
& \quad \Rightarrow n + 1, \{\text{prime} = x, \text{candidate} = n\}; \\
\text{Test1} & = \quad \{\text{prime as } x, \text{candidate as } y, \text{~ok}\} / y \mod x = 0 \Rightarrow \{\text{prime} = x\}; \\
\text{Test2} & = \quad \{\text{prime as } x, \text{candidate as } y, \text{~ok}\} / y \mod x <> 0 \\
& \quad \Rightarrow \{\text{prime} = x, \text{ok} = y\}; \\
\text{Next} & = \quad \{\text{prime as } x1, \text{ok as } y\}, \{\text{prime as } x2, \text{~ok, ~candidate}\} \\
& \quad \Rightarrow \{\text{prime} = x1\}, \{\text{prime} = x2, \text{candidate} = y\}; \\
\text{NextCreate} & = \quad \{\text{prime as } x, \text{ok as } y\} \text{ as } s / \text{right} \ s \\
& \quad \Rightarrow \{\text{prime} = x\}, \{\text{prime} = y\};
\end{align*}
\]
Hamiltonian path

\[
\text{trans } h\_path = \{ \\text{`start}, x^* \text{ as } p, \text{`stop} \\
/ \text{size}(p) = n-2 \Rightarrow \text{return } p \}
\]

\[
\text{trans } maze = \{ \text{`input}, c^* \text{ as } p, \text{`output} \Rightarrow \text{return } p \}
\]
Fractal construction by carving

Menger sponge steps

Sierpinsky sponge steps
Modeling of (bio)physical systems

Various models of Phage $\lambda$
Sperm crawling
Neurulation
Prototyping a synthetic multicellular bacteria
Higher dimensional objects for complex simulations

Example of electrostatic Gauss law \textbf{Tonti}

Electric charge content $\rho$ \hspace{1em} \textit{dimension}

Electric flux $\Phi$ \hspace{1em} \textit{dimension}

Law available on a arbitrary complex domain

$$ \phi = \iiint w \cdot dS = \frac{Q^c}{\varepsilon_0} = \iiint_{(V)} \frac{\rho}{\varepsilon_0} d\tau $$

\textit{electric field in space:}
- $V$: electric potential (dim 0)
- $E$: voltage (dim 1)
- $w$: electric flux (dim 2)
- $Q_c$: electric charge (dim 3)

\textit{A Direct Discrete Formulation of Field Laws: The Cell Method}
A Synthetic « Multicellular Bacterium »

Synthetic Biology is
A) the design and construction of new biological parts, devices, and systems, and
B) the re-design of existing, natural biological systems for useful purposes.

David Bikard, Thomas Landrain, David Puyraimond, Eimad Shotar, Gilles Vieira, Aurélien Rizk, David Guegan, Nicolas Chiaruttini, Thomas Clozel, Thomas Landrain
The Paris iGEM project a multicellular bacteria to decouple growth and transgene expression.
Implementation using BioBricks

DAP feeding → Differentiation → Germinal cell

DAP starvation → RECOMBINAISON → Differentiation

Somatic cell

cre differential control

 irreversable recombination

ftsK needed for cellular division
Proof of Concept  Simulation to answer  questions

How does differentiation induces feeding? proof of concept cellular automaton (in MGS)

diffusion of DAP

somatic and germ cell
Proof of Concept Simulation to answer questions

How does differentiation induces feeding? proof of concept cellular automaton in MGS

How do spatial organization and distribution evolve? agents based system (in MGS)
Proof of Concept  Simulation to answer  questions

How does differentiation induces feeding?  proof of concept  
cellular automaton  in MGS  

How do spatial organization and distribution evolve?  
agents based system  in MGS  

**How robust and tunable is the model?**  
ODE kinetics (matlab)
Proof of Concept Simulation to answer questions

How does differentiation induces feeding? proof of concept cellular automaton in MGS

How do spatial organization and distribution evolve? agents based system in MGS

How robust and tunable is the model? ODE kinetics

How sensitive is the system to noise? Gillespie based simulation (in MGS)
Conclusions and Perspectives
MGS drawbacks and successes

Success

Polytypisme is good
Rule application strategies are good
Patterns rules are expressive and usually concise
Clean semantics

Shortcomings

Rules may be heavy e.g. variables for the fractal sponge

Efficiency

Implicit methods solvers are hairy
Perspectives

An intrinsic complexity theory
e.g. w.r.t. interactions

A logic of spatial interactions

Relationships to physics
a discrete differential calculus cf. PhysicaD

Internalizing time

Implementation
pattern matching compilation and optimization specific
abstract combinatorial complex parallelism

Non standard applications
e.g. in knowledge representation
or in music analysis Louis Bigo talk
A topological manifesto

Spatial computing proposes to celebrate corporeality of data rather than trying to deny it

MIT medialab

The logical approach in computer science
computation  deduction
the Curry Howard isomorphism

Other paradigms can be fruitfull topology
computation = moving in a space

Try to perceive space and time in programs rather than logical operations purposes
Call for Papers: Spatial Computing Workshop 2011

3rd ANTS International Conference on Self-Adaptive and Self-Organizing Systems
Ann Arbor, Michigan, USA, October 3, 2011

Organizers: Jacob Beal (BKN Technologies, USA) Stefan Dullman (Delft University, the Netherlands) Olivier Michel (University Paris Est Créteil, France) Antoine Spiteri (University Paris Est Créteil, France)

Submission Deadline: July 4th, 2011

Many self-organizing or self-adaptive systems are “spatial computers” — collections of local computational devices distributed through physical space, in which

- the difficulty of moving information between any two devices is strongly dependent on the distance between them, and
- the “functional goals” of the system are generally defined in terms of the system’s spatial structure.

Such systems that can be viewed as spatial computers are abundant, both natural and man-made. For example, in wireless sensor networks and large collections of robots, synergetic communication of network nodes is controlled by the “signals” conveyed through

- the “collectives” as a whole solve spatially-defined problems like “analyze and react to spatial temperature variance” or “surround and destroy an enemy.”

Similarly, in reconfigurable microchip platforms, moving data between adjacent logic blocks is much faster than moving it across a chip, which in turn favors problems with spatial structure like stream processing. In biological embryos, each developing cell’s behavior is controlled not by its local chemical and physical environment, but the eventual structure of the organism is a global property...
Thanks

Antoine Spicher
Olivier Michel

PhD and other students

Louis Bigo
  J Cohen  P Barbier de Reuille
  E Delsinnew V Larue F Letierce B Calvez
  F Thonerieux D Boussié

Past and presents Collaborations

  A Lesne  IHES stochastic simulation
  P Prusinkiewicz  UoC declarative modeling
  P Barbier de Reuille  meristeme model
  C Godin  CIRAD biological modeling
  H Berry  INRIA stochastic simulation
  G Malcolm  Liverpool rewriting
  J P Banâtre  IRISA programming
  P Fradet  Inria Alpes programming
  F Delaplace  IBISC synthetic biology
  P Dittrich  Jena chemical organization
  F Gruau  LRI language and hardware
  P Liehnard  Poitier CAD Gmap and quasi manifold

http://mgs.spatial-computing.org