A Biochemical Calculus Based on Strategic Graph Rewriting

Oana Andrei[1] Hélène Kirchner[2]

[1]INRIA Nancy - Grand-Est & LORIA
[2]INRIA Bordeaux - Sud-Ouest
France

Algebraic Biology'08

Short communication

Biological Intuition

Graphs are suitable for describing the structure complex systems and graph transformations for modeling their dynamic evolution.
We are interested in a particular representation of molecular complexes as graphs and of reaction patterns as graph transformations [DL04]:

- the behavior of a protein is given by its functional domains / sites on the surface
- two proteins can interact by binding or changing the states of sites
- bound proteins form complexes that have a graph-like structure
- membranes can also form molecular complexes, called tissues

Biological Intuition

Graphs are suitable for describing the structure complex systems and graph transformations for modeling their dynamic evolution.
We are interested in a particular representation of molecular complexes as graphs and of reaction patterns as graph transformations [DL04]:

- the behavior of a protein is given by its functional domains / sites on the surface
- two proteins can interact by binding or changing the states of sites
- bound proteins form complexes that have a graph-like structure
- membranes can also form molecular complexes, called tissues

Port graphs are graphs with multiple edges and loops [AK08], where

- nodes have explicit connection points, called ports
- the edges attach to ports of nodes.

Biological Intuition

Molecular graph	**Port graph**
protein	node
site	port with maximum degree 1
bond	edge

Biological Intuition

Molecular graph	Port graph
protein	node
site	port with maximum degree 1
bond	edge

transformation of molecular complexes $\rightsquigarrow$
molecular graph rewrite rule $\rightsquigarrow$
port graph rewrite rule $\rightsquigarrow$
port graph

Biological Intuition

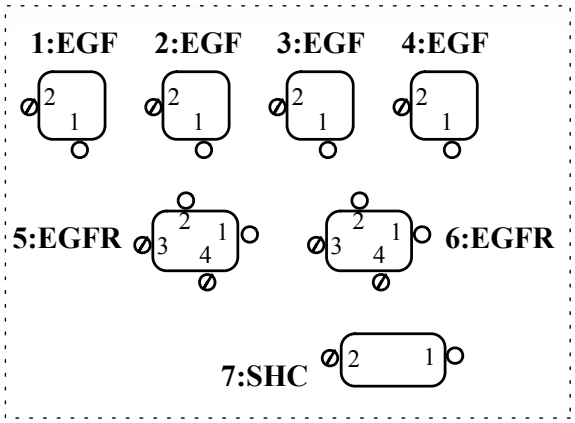
Molecular graph	Port graph
protein	node
site	port with maximum degree 1
bond	edge

transformation of molecular complexes $\rightsquigarrow$
molecular graph rewrite rule $\rightsquigarrow$
port graph rewrite rule $\rightsquigarrow$
port graph

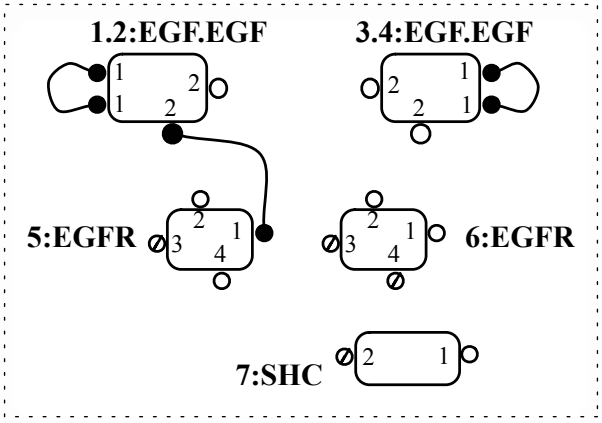
Port graphs represent a unifying structure for representing both molecular complexes and the reaction patterns between them.

Example

A molecular graph G representing the initial state of the system modeling a fragment of the EGFR signaling cascade, and a subsequent state modeled by G':

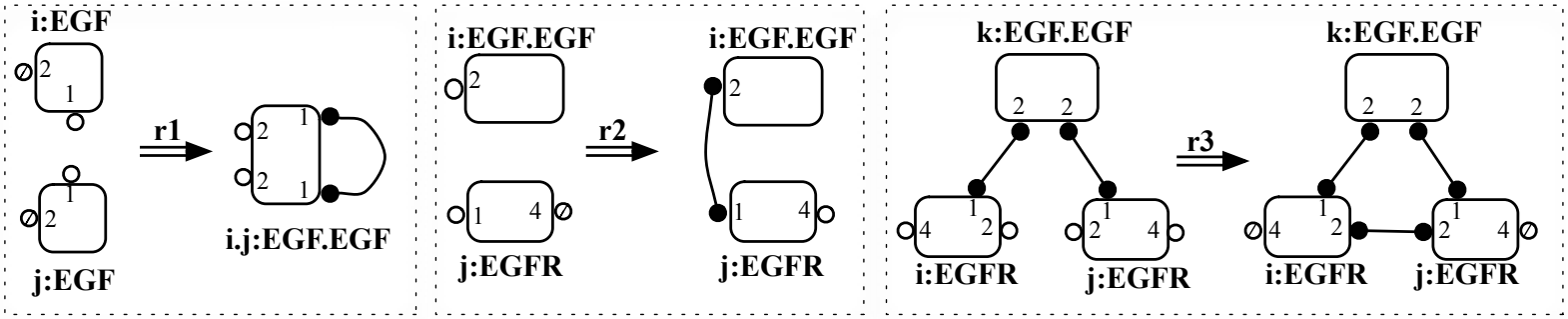


G



G'

Some reaction patterns:



Extending the chemical model

The chemical model of computation – the Γ language [BM86]:

- a chemical solution where molecules interact freely according to (conditional) reaction rules
- multisets for chemical solutions
- multiset rewrite rules for reaction rules
- extensions:
  - the CHemical Abstract Machine (CHAM) [BB92],
  - the γ-calculus and HOCL [BFR06]

A high-level biochemical calculus

A rewriting calculus [CK01] for molecular graphs with higher-order capabilities:

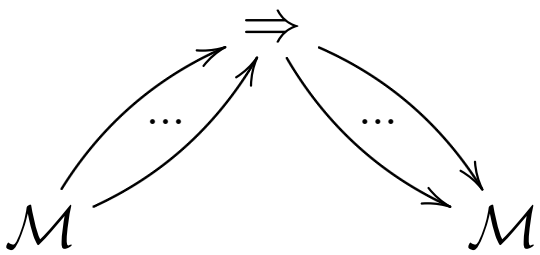
- first citizens: molecular graphs, abstractions (molecular graph rewrite rules), and rule application.
- abstractions may introduce other abstractions (the right-hand side of an abstraction may contain other abstractions)
- control mechanisms encoded as entities of the calculus (as strategies)
- extends the chemical model (Γ, CHAM, the γ-calculus) with high-level features by considering the structure of port graphs for data and for the computation rules.

Syntax

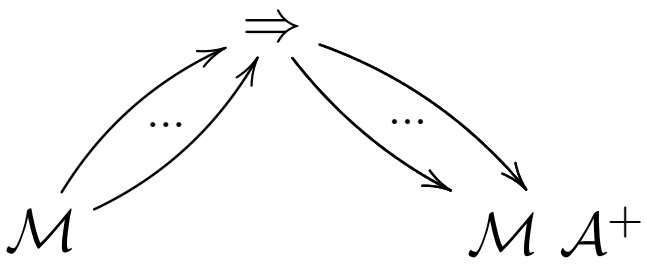
$\mathcal{M}$ the class of molecular graphs

Abstractions:

$\mathcal{A} \ ::=$



$|$

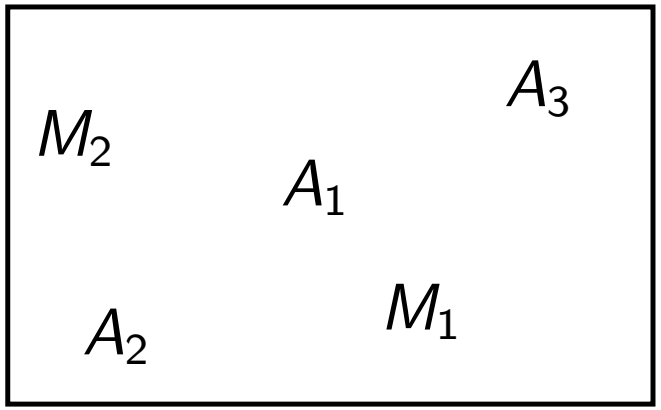


Objects of the calculus: $\mathcal{G} \ ::= \ \mathcal{X} \mid \mathcal{M} \mid \mathcal{A} \mid \mathcal{G}\ \mathcal{G} \mid \varepsilon$

State or simple world: $\mathcal{V} \ ::= \ \mathcal{Y} \mid [\mathcal{G}]$

Syntax

Box-based representation of a simple world consisting of the abstractions A_1, A_2, and A_3, and the molecular graphs M_1 and M_2:



for $[A_2\ M_1\ A_1\ A_3\ M_2]$.

Reduction Semantics

$$
\begin{array}{rll}
\textbf{(Heating)} & [X\ A\ M] \longmapsto [X\ A@M] & (1) \\
\textbf{(Application/Success)} & A@M \longmapsto G \quad \textbf{if} \quad M \rightarrow_A G & (2) \\
\textbf{(Application/Fail)} & A@M \longmapsto A\ M \quad \textbf{otherwise} & (3)
\end{array}
$$

Reduction Semantics

$$(\textbf{Heating}) \qquad [X\ A\ M] \longmapsto [X\ A@M] \tag{1}$$

$$(\textbf{Application/Success}) \qquad A@M \longmapsto G \quad \textbf{if} \quad M \rightarrow_A G \tag{2}$$

$$(\textbf{Application/Fail}) \qquad A@M \longmapsto A\ M \quad \textbf{otherwise} \tag{3}$$

By introducing an explicit object (node) for failure, stk, we gain in expressivity:

$$(\textbf{Application/Fail}') \qquad A@M \longmapsto \mathrm{stk} \qquad \textbf{if } M \textbf{ is } A - \textbf{irreducible} \tag{4}$$

Reduction Semantics

$$(\textbf{Heating}) \qquad [X\ A\ M] \longmapsto [X\ A@M] \tag{1}$$

$$(\textbf{Application/Success}) \qquad A@M \longmapsto G \quad \textbf{if} \quad M \rightarrow_A G \tag{2}$$

$$(\textbf{Application/Fail}) \qquad A@M \longmapsto A\ M \quad \textbf{otherwise} \tag{3}$$

By introducing an explicit object (node) for failure, stk, we gain in expressivity:

$$(\textbf{Application/Fail}') \qquad A@M \longmapsto \mathrm{stk} \qquad \textbf{if } M \textbf{ is } A - \textbf{irreducible} \tag{4}$$

Possible extension: consider a structure of all possible results for application

Strategies

Instead of this highly non-deterministic (and possibly non-terminating) behaviour of abstraction application, one may want to introduce some control to compose or choose the abstractions to apply, possibly exploiting failure information.

Strategies as abstractions:

$$
\begin{aligned}
\mathsf{id} &\triangleq X \Rightarrow X \\
\mathtt{fail} &\triangleq X \Rightarrow \mathsf{stk} \\
\mathtt{seq}(S_1, S_2) &\triangleq X \Rightarrow S_2 @ (S_1 @ X) \\
\mathtt{first}(S_1, S_2) &\triangleq X \Rightarrow (S_1 @ X) \;\; (\mathsf{stk} \Rightarrow (S_2 @ X)) @ (S_1 @ X) \\
\\
\mathtt{try}(S) &\triangleq \mathtt{first}(S, \mathsf{id}) \\
\mathtt{repeat}(S) &\triangleq \mathtt{try}(\mathtt{seq}(S, \mathtt{repeat}(S)))
\end{aligned}
$$

Improving the calculus using strategies

1. Failure catching: if $S@M$ reduces to the failure construct stk, then the strategy $\texttt{try}(\text{stk} \Rightarrow S\ M)$ restores the initial entities subjects to reduction.

$$\textbf{(Heating')} \qquad [X\ S\ M] \longmapsto [X\ \texttt{seq}(S, \texttt{try}(\text{stk} \Rightarrow S\ M))@M]$$

2. Persistent strategies: $S!$ applies S to an object and, if successful, replicates itself.

$$S! \triangleq \texttt{seq}(S, \texttt{first}(\text{stk} \Rightarrow \text{stk}, Y \Rightarrow Y\ S!))$$

Conclusions and Perspectives

Conclusions:

- we defined a higher-order calculus with high-level capabilities for modeling interactions in molecular complexes.
- from the verification point of view we have:
  - classical rewriting techniques for checking properties of the modeled systems: verification of confluence, termination for port graph rewriting (under strategies)
  - ideas for runtime verification of properties in such systems

Conclusions and Perspectives

Conclusions:

- we defined a higher-order calculus with high-level capabilities for modeling interactions in molecular complexes.
- from the verification point of view we have:
  - classical rewriting techniques for checking properties of the modeled systems: verification of confluence, termination for port graph rewriting (under strategies)
  - ideas for runtime verification of properties in such systems

Perspectives:

- verification
- interactions between abstractions
- control mechanisms

O. Andrei et H. Kirchner – "A Rewriting Calculus for Multigraphs with Ports.", *Proceedings of RULE'07*, Electronic Notes in Theoretical Computer Science, vol. 219, 2008, p. 67–82.

G. Berry et G. Boudol – "The Chemical Abstract Machine.", *Theoretical Computer Science* **96** (1992), no. 1, p. 217–248.

J.-P. Banâtre, P. Fradet et Y. Radenac – "A Generalized Higher-Order Chemical Computation Model", *Electronic Notes in Theoretical Computer Science* **135** (2006), no. 3, p. 3–13.

J.-P. Banatre et D. L. Metayer – "A New Computational Model and Its Discipline of Programming.", Tech. Report RR-566, INRIA, 1986.

H. Cirstea et C. Kirchner – "The Rewriting Calculus - Part I and II", *Logic Journal of the IGPL* **9** (2001), no. 3, p. 427—498.

V. Danos et C. Laneve – "Formal Molecular Biology.", *Theoretical Computer Science* **325** (2004), no. 1, p. 69–110.