数理解析研究所講究録 1222 巻 2001 年 128-137

Pathway Graph Models for Molecular Computing *in situ*

LIU Jian-Qin and SHIMOHARA Katsunori

Information Sciences Division, ATR International

2-2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288, Japan

Abstract

This paper proposes a new class of models for molecular computing with biological plausible backgrounds, in which formalization, computability, logic forms and related prototype characteristics are presented and discussed. Some helpful conclusions for the further systematic studying of kinase computing and its formal theories are provided.

1. Introduction

As an unconventional computational paradigm, one of the most important purposes of our work is to surpass conventional technology. With regard to technology, this paper targets a number of challenging topics, such as scalability, speed and error-handling. Our motivation comes from optimizing the above three topics by exploring computability to construct a scalable and robust scheme for fault/defect-tolerance and speed by employing a "self-feeding" method, which is a helpful step to bridge the gap between formal systems and programming.

2. Molecular Computing *in situ* by Kinase-centered Signaling Pathways in Functional Cells

Among the various implementation technologies for molecular computing, DNA, RNA and other bio-molecules have been harnessed as components to build a molecular computer. Briefly speaking, kinase computing here refers to molecular computing by kinases and their related molecules in living cells. The biological background mainly comes from the Rho family [1]. The original contribution of our work on kinase computing can be summarized as the initiation of a novel unconventional computational paradigm to extract the inexplicit structure of the problems and the computability which leads to super-Turing machines. As we know, bio-informatics offers us a rich landscape of

biologically-inspired information mechanisms that can be classified into unconventional computing. Among them, proteomics helps us to understand biochemical reactions related to signaling pathways and cell communication [1]. Sieving the various types of proteomics-based processes in cells, kinase and related enzymes and functional proteins have become our starting point for studying these kinds of biochemical mechanisms existing in nature. This is the core spirit of inventing synthesis engineering by bio-molecules and/or inorganic molecular electronics for moleware building based on the analysis of bio-informatics.

3. Formalization, Graph Rewriting and Beyond

In this paper we present our theoretical work on formal models of kinase computing -- LS-systems. At the beginning of our work on LS-systems, computability was our focus but we later found important merits in the aspects of formal languages, algebraic systems (especially in semi-groups) and the theory of computation in the mainstreams of both theoretical computer science and multidisciplinary fields related to molecular biology, mathematics and information engineering. Clearly, a broad area emerging from kinase computing is within our scope.

3.1 LS-systems

The basic class of LS-systems is defined as follows:

Let the construct be $W_{ls} = < V, E, Y, A, B, Q, Z >$

V -- the set of vertexes;

E -- the set of edges;

Y -- the set of hyperedges for pathways;

A -- the alphabet set;

B -- the set of the constraints that can be described/represented as logic, function, set or other form;

Q -- the set of operators or operations for graphs or hypergraphs from V, E, and Y;

Z -- the set of local concentration.

The fundamental of the computing processes is constructed on a parallelism where the PEs (Processing Elements) are formalized as vertexes in graphs and the whole set of related graphs is denoted as G. The E set is generated according to links between different PEs.

Here the PEs can be implemented by the moleware form. The pathways refer to the strings defined by the symbolic set A. Under the condition of Y, the main applied operations in B include (1) feedback-making, (2) interacting, and (3) stable-checking. Under the circumstances of the cells where dynamical operations for signaling molecules exerted, local features (e.g., concentrations) reflect the heterogenous architecture of the computational processes constructed above.

3.2 Simulation Model

For an overview of the feasible implementation, our proposed simulation model is given in figure 1. It consists of three layers: (a) the biochemical molecular layer (the lowest layer), (b) the pathway layer (the middle layer), and (c) the operation layer (the highest layer).

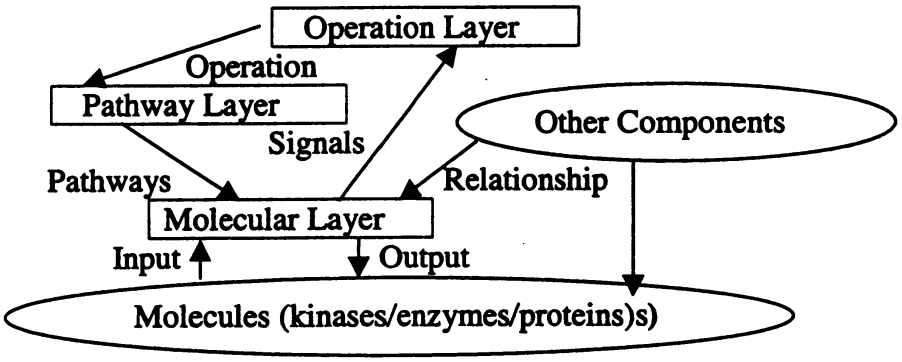

Fig. 1 Overview of the simulation model

In terms of software, the main dataflow includes the input and output of the molecules that are handled in the molecular layer, where measures such as the local concentrations in organelles and cells are defined. The number of molecules that we have used here in computing is limited. They are drawn from a database verified by experimental evidence from molecular biology (e.g., benchworks). Real values are assigned to objects and combinatorial ODEs (Ordinary Differential Equations) corresponding to the pathways. Theoretically speaking, infinite real numbers can be obtained in the computing system mentioned above through finite molecules. In essence mapping in functional space is necessary for determining the relationship between the finite and the infinite through corresponding topological constraints (e.g., high dimensional networks).

3.3 Graph Rewriting and More

From this simulation model we can derive the following graph rewriting system:

$< G, Q, B' >$

G -- the set of hypergraphs;

Q -- the operation set;

B' -- the set of related constraints.

Owing to the fact that graph rewriting is useful for parallel computing software and programs, we have developed a kernel of simulation software in an object-oriented way and designed a graph rewriting language with primitive forms independent of machines based on representation by a symbolic set of strings for hyperedges and their interactions with increasing complexity. The three factors of context dependence, language level's operation, and context sensitive grammar need to be handled in the corresponding landscape of grammars that are derived from formal models. This bids the question of how to infer structures in pathways and emerge/derive formal languages. The main results obtained are: (1) The non-feedback and linear combined pathways' model gives regular grammar, (2) The pathways' model with interactions under certain conditions gives context free and context sensitive grammar, and (3) The model gains the computability of Turing machines when the feedback is equipped with the interactions. Through a parallel structure, 0L can be inferred. But the relationship between our model and indexed languages still remains open and we plan to address this in the near future. With the high speed of molecular electronics, femto-second speeds can be expected and therefore the quantum effect must be considered, so our proposed non-Gödel construction schemes for non-Turing machines can be used to describe the uncertain related phenomena and related mechanisms, in which we use moleware *in quanta* to represent a molecular computer prototype where a single molecule level is adopted for the molecular logic and gate arrays. Also considering evolutionary computation in the natural computing domain, let us formalize evolutionary algorithms as

< We, Qe, Ge, Ye >

where,

We -- the population set;

Qe -- the evolutionary operator set;

Ge -- the selection criterion set;

Ye -- the termination criterion set.

When a population consisting of individual object-forms is provided to handle the scalability of NP problem-solving, the open end of evolution yields increasing complexity in parallel searching with adaptive fitness evaluation as a kind of pruning for the underlying structure (e.g., string, tree, graph or other high dimensional structures that are concrete in programming and data structure or abstract in networks). A certain degree of cost-cutting can be achieved from the parallel evolution of the population in an adaptive way.

4. Super-Turing Machine

Further, we envision the above model as a "Kinase Computing model with Local Concentrations (KCLC)", which is shown in Figure 1 and study its computability. Here, super-Turing machines refer to a kind of abstract machines which can compute the Turing-machine's in-computable information.

Theorem: The KCLC model is equipped with the computability of super-Turing machines.

Proof (sketch):

Step 1: The real valued domains for the input and output are defined as topological areas in measurable space, so the problem becomes a processes of recursively exerting the operations of constructing and reconstructing the combined mapping with increasing degrees of complexity.

Step 2: According to the theorems by Kolmogorov in 1957 and Sprecher in 1965 [2], the function sets and combined sets are constructive and the mapping set is feasible, so the mapping set from real space to real space is guaranteed.

Step 3: In order to prove the feature that the whole space of the real numbers for the input and output can be covered, we use the method of eliminating the assumption of controversial examples that certain uncoverable non-Turing computable numbers exist. If so, we can obtain the result that the pathway set can not be included in the Q set to operate them in order to add the resultant ones into the set already obtained. Assuming this, at least one pathway exists out of the pathways that are believed to have fallen into the proven domain, but they are reachable according to the links attached to them. This leads

to controversy. Any new pathway that occurs by adding an operation from Q is also reachable and acceptable in logic. So the coverability is assured.

Step 4: The number of kinases and enzymes is finite and the mapping determined by them is coverable for the real spaces above.

Step 5: Through finite operation from graph rewriting, the pathways reflecting the real-valued objects can be produced. They can be reduced into the pathways that are based on the three basic operators of feedback-making, interaction and survival checking. The non-Gödel logic introduced is necessary in the meaning of logic, leading to the framework for the inferring. Q.E.D.

This class should include the subclass with Turing computability discussed in the next section.

5. Universal Computation

We define the measure of fitness for judging whether the objects in KCLC should be allowed to exist. The strings, i.e., the sequences of the symbols for the reactants in the pathways, derived from the hyperedges construct a subclass of KCLC, which is denoted as KCLC1.

Theorem: KCLC1 is Turing-computable.

Proof (sketch):

On the basis of the previous formalization, the function set

$$\{f(.) \mid D_{pop} \rightarrow D_{fit}\}$$

is our starting point. Here, D_{pop} is the set of the objects, which means the domain of the population, and D_{fit} is the measure for the fitness. The primitive recursive function for constant 0, the successor function and the project function can be inferred directly from the quantitative relation inherited from the structure given.

(a) Assuming that L (.) are defined in the value domain of $[0, f_{max}]$, f_{max} is a sufficiently-large value acting as a threshold. There exists an a such that $L(a)=0$ (L(.) belongs to { f (.) }) where a belongs to D_{pop}. This is obvious owing to the physical meaning of fitness.

(b) Increasing the value for f (.) is also feasible, i.e., there exits a b such that $H(b) = b+1$, where H (.) belongs to f (.) owing to the existence of valuing mapping for the relationship between strings and numbers.

(c) Let $Y_p = f_p (x_1,x_2)$ represent the pathway from reactants X_1 and X_2 to product Y_p. According to the existence of the feedback, we obtain

$$f_p (x_1,x_2) = x_1 \text{ or } x_2.$$

We can infer that

$$f_{pi} (x_1,\ldots,x_n) = x_i \ (i = 1,2,\ldots,n)$$

exists.

We can also define the following three hypergraph representations:

G_1: for $(x_1, x_2, \ldots, x_n) \rightarrow f_{g1} = f (x_1, x_2, \ldots, x_n)$

G_2: for $(x_1, x_2, \ldots, x_n, x_j) \rightarrow f_{g2} = f (x_1, x_2, \ldots, x_n, x_j)$

G_3: for $(x_1, x_2, f(x_1, x_2, \ldots, x_n, x_j) \rightarrow h(.)$.

where f_{g1}, f_{g2}, h(.) belong to the set of { f(.) }.

Then, the rewriting of G_1 to G_2 and then G_2 to G_3 is feasible in the linear time of O(n+4).

For the relationship between the *k*-place function Z_k and the $(k+1)$-place function Z_{k+1}, we can obtain the following result:

$$Z_k (l) = m \text{ exists such that } Z_k (l,m) = 0,$$

when the index *l* corresponds to the notation of the reactants in the *k*-order of the pathway and *m* represents the added reactant (labeled $k+1$). This means that the pathways give the result that the final product with the index $k+1$ will stop when at least one enzyme is included in the *k* order of the pathway. If the *k* order of the pathway can not produce $k+1$, it is obvious that $Z_k (l)$ must be zero.

So, KCLC1 is proved to be equivalent to a class of μ-recursive functions. So finally its Turing computability is guaranteed. Q.E.D.

After the Turing computability is obtained, simple operation can easily be implemented. The number of these operations is not infinite for the universal computation in our model. In the cell environment we can control the Q operations by means of a kinase-based biochemical reaction mechanism, and a gene regulation function can also easily be embedded. This is a biologically supported argument.

6. Logic Description Based on Extended RPL

Based on the definition, above, the relation K_q on the Q set is defined as:

$q_i \ K_q \ q_j$ where i and j are integers, qi, q_j belong to the concerned objects defined in

the previous sections.

According to the truth value in the meaning of RPL (Rescher Probabilistic Logic), y (p) belongs to [0,1] (p belongs to A which is the set of all of the propositions; notice here that p is not probability).

The difference that we have constructed from RPL is

$$y (p H_y q) = m(p) * m(q)$$

where H_y is defined as a kind of K_q , m(p) and m(q) refer to the different measures related to the Q set , respectively.

As a of meta-theory, the description ability is enhanced after introducing the H_y operation.

Theorem: The logic construction for the Q set through extended RPL is close (it can also be understood as "complete" in the meaning of the non-monotonic logic of artificial intelligence, but this is not the completeness of Gödel's concept).

Proof (sketch):

For the structure of the relationship between domain and assignment, the assignment mapping y is given as

$$y \rightarrow [0,1]$$

So the atomic formula can be easily constructed. And we get

A [s] is true, s is an object defined in the previous sections.

So we have y (A [s]) = 1 and know that the atomic formula is true. Here T and F for the Q set are assigned to the feasible pathway and unsustainable or impossible pathway, respectively. Further, the probability for the final survival verification is employed to induce the multi-value logic forms by RPL as [0, 1] and "V_{and}" and "V_{or}" for the merging of two pathways and the extraction of the common part of two pathways, respectively. When the axiom set and separation rule are kept the same as the ones in RPL, an extended operator H_{ls} is introduced under the following true value constraint:

$$y (p V_{and} q) = H_{ls} (p,q)$$

where H_{ls} = min (v(p), v(q)) + min (v(p), v(q)) / (v(p) + v(q)) .

When the proposition from the extended RPL is not the atomic formula, through the dividability of the Q set, it can be

$S_q^{n+1} = S_q^n + S_q^*$ s.t. $y(.)$, S_q^{n+1}, S_q^n, S_q^* are the elements in the set of pathways, they are also corresponding to the elements in Y.

This is reasonable provided that S_q^n is close for Q operators, so it is sufficient to only prove the existence of S_q^*. S_q^* can be inferred as one of the following three situations: (a) the feedback-inserting, (b) interaction of the obtained pathways, or (c) survival pathway verification by duration. So the results can be concluded as (a) truth values unchanged, (b) the F or non-zero, or (c) the T or F. Therefore the truth values' feature in n+1 can be inherited from the one in n. Q.E.D.

7. A Prototype of Kinase Computing *in situ*:

The prototype we have built consists of the following major parts:

(1) Implementation of arithmetic and logic operations: In order to implement the basic units of computers, we must design an addition unit, which can be applied to carry subtraction, multiplication and division by means of hardware for operation in a CPU. Contrary to the molecules' ways in existing bio-molecular computing, we work out "pathways" that contract the MIMD molecular information flow and are also capable of topological coding and are controllable for integrated cell signaling and communication. The logic operations, such as AND, OR, NOT and others as XOR can be made mainly by three methods: (a) the hardware-like method for implementing the assembly-language level instructions of logic based on an addition unit, (b) the basic logic unit of AND, OR and NOT and more complicated logic built by gate circuits, and (c) tables applied to store the input-to-output information for data accessing, which is beneficial to the large-scale memory of bio-molecular computers and the whole-memory schemes of molecular computers employing molecular electronics.

(2) Algorithms for solving NP problems such as the classes of PSPACE-hard and other graph-related problems by P-cost.

(3) Robust coding schemes based on the formal systems proposed above and simulations.

(4) Unsupervised training for pattern mining and its extendability for non-parametric learning: The core of our method is that we do not need any prior knowledge from a known sample set. The principle here is that "Let the systems tell their own story by themselves (LSTX)", i.e. the LSTX principle. This is a requirement from the knowledge discovery tasks in molecular biology for real world problems from bio-informatics.

(5) Scalability for NP problem solving by 10 quadrillion molecules: We have simulated the 3-SAT (50, 20) with 10,000,000,000,000,000 molecules. The methods we developed here are helpful to potential applications in massive signal processing such as image or speech and high-dimensional clustering.

(6) Computer-aided design schemes for wetware implementation: The implementation is expected to be done by cell systems through methods of synthesis engineering, which is based on the analysis with simulation and experimental verification.

(7) Robust computational model for fault/defect-tolerant molecular computing: Enlightened by our discovery of the "ladders" phenomena in algorithm chemistry (also called artificial chemistry), we infer that their architecture is equipped with efficient task allocations over the whole system where dynamical decision-making is used for molecule changes, pathway changes and parameter changes.

8. An Open problem

Originating from the formal system here, an important question remains: "What is the complete language hierarchy in the space out of Turing machines and with a stronger computability than Turing machines?" To answer this, we are exploring the computational mechanism in the single molecular level where the quantum effect occurs, which is leading to molecular computing *in quanta*.

Acknowledgement

J.-Q. Liu sincerely thanks Prof. Masami Ito for his teaching and academic advice on formal languages and theory of computation, Prof. Tatsuhiko Sato, Prof. Teruo Imaoka, Prof. Hidenosuke Nishio, and Prof. Reiji Nakajima for their suggestions on algebraic theory, semi-groups, formalization and computability, respectively.
This work is partly supported by the Huo Yingdong Foundation (Project No. 71063) and the National Natural Science Foundation of China (Project No. 69985008).

References

[1] K. Kaibuchi, S. Kuroda, and M. Amano, Regulation of the cytoskeleton and cell adhesion by the Rho family GTPases in mammalian cells, Annu. Rev. Biochem. 1999.68: 459-486.

[2] J. Suykens et al., Artificial neural networks for modelling and control of non-linear systems. Boston: Kluwer Academic Publishers, 1996; 23-24.