社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－MPS－60 (5)
2006／6／26

人工化学における推論の一手法

冨永 和人

東京工科大学コンピュータサイエンス学部

人工化学を用いた研究では，ある設定を系に与え，その設定の下で系がどのような振舞いをするかを観察する研究手法が一般的である．これに対して本研究では逆の問題，すなわち指定された振舞いが起きるためには系にどのような設定が必要であるかを計算することを試みる．用いる道具は推論である．我々は文字列のパターンマッチと組換えに基づく人工化学を提案している．本稿ではこの人工化学を用いて構成される系において，指定された分子が生成されるか否かを計算する簡単な手続きを与える．この手続きは論理における後向き推論に類似している．すなわち，生成されるかどうかを調べる分子をゴールとして設定し，可能な化学反応経路を反応の逆順に求めて行く．

An Approach to Reasoning in an Artificial Chemistry

Kazuto Tominaga

School of Computer Science, Tokyo University of Technology

Artificial chemistries are usually used to study what phenomena occur in a given setup, and simulators are the primary tools. This paper attempts to address a reverse question, that is, what setup is required to produce given molecules, using reasoning as the tool. We give a procedure to answer whether specified molecules can be produced in a given system described in our artificial chemistry, which is based on string pattern matching and recombination with no spatial structure. The process of reasoning is similar to backward reasoning in logic: it starts with setting the specified molecules as the goal, and gradually constructs a possible reaction path in the reverse order of recombinations.

1 Introduction

Artificial chemistries are abstract models of chemical reactions, and recently they are becoming one of the main research methods in the field of artificial life [1]. They have been used to study important problems in artificial life such as evolution and self-replication [1], as well as modelling real biochemical systems [4]. A most common methodology is to model the target (real or virtual) system in an artificial chemistry and then to run it on a simulator of the artificial chemistry. A main objective of studies with this methodology is to see *what phenomena occur in the given setup*, and therefore simulators are the primary tools.

In this paper, using our artificial chemistry [5], we explore possibilities to answer a reverse question, *what setup is required to give rise to a desired phenomenon*, which, to the authors' knowledge, has not been addressed using a formal method. Since it is difficult to formulate a phenomenon, we instead try as our first step to answer what setup is required to produce specified *molecules*. The tool we use to make this attempt is *reasoning*.

2 An artificial chemistry

In this section, we give a brief explanation on a simple artificial chemistry based on string pattern matching and recombination [5]; it employs the well-stirred reactor model (i.e., the reactor has no spatial structure).

Elements and objects. An *element* is denoted by an upper-case character, or a capitalized sequence of alphanumeric characters. For example, `A`, `B`, `Atp` and `Cd4` are elements. An element corresponds to an atom or a group of atoms in nature. An *object*, which corresponds to a molecule or a compound of molecules, is a character string or a stack of strings as shown below.

Objects are denoted by strings; the above objects are denoted by `0#ABC/`, `0#ABAB/3#CC/` and `0#ABC/1#Ca/`, respectively. A number represents the displacement of a line relative to the first one. A displacement can be positive, zero or negative.

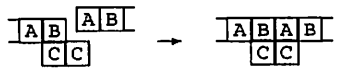
Patterns. A *pattern* matches (or does not match) an object, and it may include two kinds of *wildcards*. An *element wildcard*, which is denoted by a number surrounded by angle brackets such as `<1>`, matches any element. A *sequence wildcard*, denoted by a number and an asterisk with angle brackets such as `<2*>` and `<*2>`, matches any sequence of zero or more elements;

the position of an asterisk represents the direction in which the sequence can extend. The string notation for patterns are similar to the one for objects. For instance, the pattern `0#A<1>C/` matches the object `0#ABC/`, and `0#<*1>AB/1#CC/` matches `0#ABAB/3#CC/`. Note that the displacements are meaningful in pattern matching, and the length of a sequence wildcard is regarded as zero in the pattern notation.

Recombination rules. A group of objects is transformed into a group of objects by a *recombination rule*; it conserves elements like a chemical reaction does. A rule is expressed in terms of patterns as follows.

```
0#<*1>AB/1#CC/ + 0#AB<2*>/
                → 0#<*1>ABAB<2*>/1#CC/   (1)
```

This rule works as follows.



If this rule is applied to `0#ABAB/3#CC/` and `0#ABB/`, `0#ABABABB/3#CC/` is produced and the reactants vanish. We call the left-hand side of a rule *lhs*, and the right-hand side *rhs*.

Dynamics. A *system* is a construct $\langle \Sigma, R, P_0 \rangle$ where Σ is a set of elements, R is a set of recombination rules, and P_0 is the *initial pool*, which specifies objects in the *pool* at the initial state. The system is interpreted nondeterministically as follows: (1) Initialize the pool to be P_0; (2) Apply one recombination rule to a collection of objects in the pool; (3) Go to Step 2.

3 Definitions

Here we define some auxiliary notions and operations.

Pattern as a set of objects. Let O_Σ denote the (infinite) set of possible objects composed of elements in Σ. When a pattern is given, it matches a subset of O_Σ; in other words, a pattern represents a set of objects in O_Σ. Hereinafter we apply the conventional set relations and operations such as member ($\in$), union ($\cup$), intersection ($\cap$), difference ($\setminus$) and subset ($\subset$, $\subseteq$, etc.) to patterns.

Matching between patterns. In Sec. 2, matching is defined between a pattern and an object. We extend the notion to be between patterns: patterns p and q *match* iff $p \cap q \neq \phi$. For example, the patterns `0#A<1>C<2>/` and `0#<3>BC<4>/` match since the object `0#ABCD/`, say, is included in both patterns. This definition is also valid for a pattern and an object if an object is regarded as a literal pattern, which comprises only that object.

Easiness relation between multisets of patterns. We define a relation $\sqsubseteq$ between multisets of patterns: $W \sqsubseteq U$ holds for the two multisets of patterns W and U iff there exists a multiset of patterns V such that V contains, for each $p \in W$, an instance of pattern q that satisfies $q \subseteq p$, and $V \subseteq U$. $W \sqsubseteq U$ means "W is easier to generate than U": if U matches a collection of objects, W matches at least one of its subset (such an instance is V). $W \sqsubseteq U$ holds, say, for $W = [$`0#A<1*>/`$]$ and $U = [$`0#AB<1*>/,0#C/`$]$ with $V = [$`0#AB<1*>/`$]$. ("$[\,]$" denotes a multiset.)

Projecting patterns on elements. We define a projection operator ($/$): P/Σ is a multiset of elements (or literal element patterns) that are included in P, where P is a multiset of patterns and Σ is a set of elements. Wildcards are disregarded. For example, if $P = [$`0#<*1>AB/,0#AB<2>B/,0#CD/`$]$ and $\Sigma = \{$`A,B,C`$\}$, then $P/\Sigma = [$`A,A,B,B,B,C`$]$.

Narrowing patterns. If a pattern has one or more wildcards, assigning elements to a wildcard makes the pattern specialized. For example, if we have the pattern `0#A<1>C<2>/` and the element `B` is assigned to the wildcard `<1>`, the pattern becomes `0#ABC<2>/`, which is a subset of `0#A<1>C<2>/`. Now suppose we have the following recombination rule.

```
0#A<1>C<2>/ → 0#AC<1><2>/   (2)
```

If `<1>` in lhs is assigned `B`, the pattern on rhs should be understood as `0#ACB<2>/`; similarly, if `<1>` in *rhs* is assigned `B`, the pattern on *lhs* should be understood as `0#ABC<2>/`. We call this "specialization of patterns on one side *narrows* the patterns on the other side."

4 A simple reasoning procedure

In this section we give a simple procedure to answer whether a specified collection of objects can be produced in a system $\langle \Sigma, R, H_0 \rangle$. The collection is given as a multiset of patterns W_0, and the initial pool is also given as a multiset of patterns H_0. Recall that objects are regarded as literal patterns.

Intuitively, the procedure tries to construct in the reverse order the state transition of the pool induced by recombinations. A state of the pool is represented by a *world* $\langle W, H \rangle$. The goal world is $\langle W_0, H_0 \rangle$. A reasoning path is a sequence of worlds starting at the goal world. The process of creating worlds is illustrated in Fig. 1. Each W, which is shown in the upper part of an oval, is a required subset of the pool at that step. Hs are made accompany Ws in order only for reasoning, and they may not appear in the pool at that step. The procedure has several choice points; it should backtrack when a reasoning path fails.

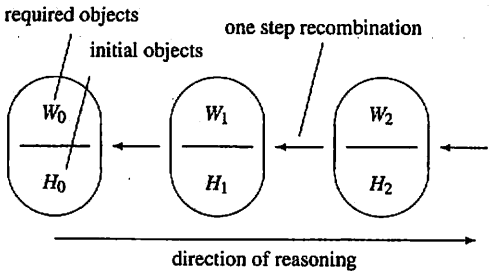


Fig. 1 A sequence of worlds.

W and H in the procedure are multisets of patterns, and so should they be treated; for example, removing an element e from W means removing one instance of e from W. The set relations and operations are construed to work on multisets.

Procedure 1 *to answer whether a multiset of objects W_0 can be generated in a system $\langle \Sigma, R, H_0 \rangle$:*

1. *Initialize: Let $\langle W, H \rangle = \langle W_0, H_0 \rangle$.*
2. *Check (a): If $W' \sqsubseteq W$ for any world $\langle W', H' \rangle$ downstream, fail.*
3. *Discharge: If p and q match for some $p \in W$ and $q \in H$, p and q can be removed from W and H, respectively, and compute $q' = p \cap q$. If $W = \phi$, answer "yes" (success).*
4. *Check (b): If H includes no wildcard and $(W/\Sigma) \setminus (H/\Sigma) \neq \phi$ holds, fail.*
5. *Reverse application of a rule:*
   (a) *Choose a rule $r \in R$: $s_0 + s_1 + \cdots + s_m \rightarrow t_0 + t_1 + \cdots + t_n$ such that there exists at least one pattern in W that matches some t_i $(0 \leq i \leq n)$. If there is no such pattern and rule, fail.*
   (b) *For each i $(0 \leq i \leq n)$ choose or not a pattern $p_i \in W$ that satisfies $p_i \cap t_i \neq \phi$, and let the multiset of such chosen p_i be P. P must satisfy $P \subseteq W$.*
   (c) *Specialize t_i to obtain t_i' $(0 \leq i \leq n)$:*

$$t_i' = \begin{cases} t_i \cap p_i & \text{if } p_i \in P \\ t_i & \text{otherwise} \end{cases}$$

   (d) *According to the specialization, narrow $s_0, s_1, \ldots, s_m$ to obtain $s_0', s_1', \ldots, s_m'$.*
   (e) *Remove each $p_i \in P$ from W, and add $s_0', s_1', \ldots, s_m'$ to W.*
6. *Go to Step 2 for the next world.*

□

Step 1 sets the goal world. Step 2 checks whether there exists a downstream world $\langle W', H' \rangle$ easier to create than the current world $\langle W, H \rangle$; in such a case, an answer (if any) should be reached directly from $\langle W', H' \rangle$ without going through $\langle W, H \rangle$, so we cut the search path. Step 3 uses some of the patterns given in H_0, and if W becomes ϕ, a reaction path is found. Step 4 checks whether there are still enough elements in H to produce W (note that this artificial chemistry conserves elements); if there are not, fail. Step 5 makes the world before one step recombination of the form

$$s_0' + s_1' + \cdots + s_m' \rightarrow t_0' + t_1' + \cdots + t_n' . \qquad (3)$$

Steps 2 through 6 are iterated; at the end of each iteration a new world is created, which is a possible previous state of the current world before the recombination expressed by Formula 3 above.

5 Example reasoning

Here is an example system that generates sequences of ABs, namely, `0#ABAB/`, `0#ABABAB/` and so forth, given many objects of the form `0#AB/`. The concatenation is mediated by `0#CD/`. Shown below are the recombination rules of this system.

```
0#<*1>AB/ + 0#CD/ → 0#<*1>AB/1#CD/                    (4)
0#<*1>AB/1#CD/ + 0#AB<2*>/
                  → 0#<*1>ABAB<2*>/1#CD/               (5)
0#<*1>ABAB<2*>/1#CD/
                  → 0#<*1>ABAB<2*>/ + 0#CD/            (6)
```

The example problem is

> *Can `0#ABABAB/` be generated from the initial pool with three `0#AB/` and one `0#CD/`?*

and the answer is yes. The goal world is

$$\langle W_0, H_0 \rangle = \langle [\texttt{0\#ABABAB/}], [3 : \texttt{0\#AB/}, \texttt{0\#CD/}] \rangle$$

where `3 : 0#AB/` represents three instances of `0#AB/` in the multiset. Figure 2 shows the search tree of the reasoning. A world is represented as a tree node by the form "W ; H." For example, Node 0, standing for the goal world $\langle W_0, H_0 \rangle$, is "`0#ABABAB/ ; 3 : 0#AB/ 0#CD/`," which means $W = [$`0#ABABAB/`$]$ and $H = [$`3 : 0#AB/, 0#CD/`$]$. The node numbers from 0 through 11 are the order of visiting the nodes; the reasoning starts at Node 0 and succeeds at Node 11 (where $W = \phi$), and it answers "yes."

At Node 6, Check (b) in Procedure 1 fails and backtracking to Node 4 occurs. Nodes at which backtracking occurs due to no further applicable recombination

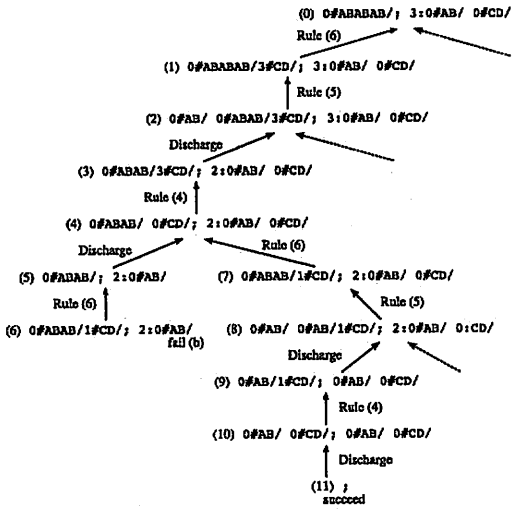


Fig. 2 A search tree for a possible reaction path.

rule (Step 5a) is not explicitly marked *fail*. A solid arrow represents a recombination, and the dashed arrows are other possible reasoning paths, which had not been taken because the reasoning succeeded at Node 11.

6 Discussion

Procedure 1 is sound, i.e., a given system can produce a specified multiset of patterns if the procedure says "yes," since a reaction path can be constructed from the result of reasoning (details omitted due to page limitation). On the other hand, since the artificial chemistry is computationally universal [5] the procedure cannot be complete, and therefore there is no such a general procedure. The process of reasoning may not necessarily terminate, though the procedure is designed so that H monotonously diminishes.

The procedure resembles backward reasoning by resolution: W_0 is the goal, H_0 is the set of literals, and the recombination rules R are implications. Since the artificial chemistry deals with multisets of patterns, it relates more to linear logic [3] . Fontana and Buss [2] designed an artificial chemistry using the following analogy to linear logic: (1) a chemical molecule and a proof net in normal form, and (2) chemical reaction and application/elimination of the "cut" inference rule. In contrast, our approach is regarded to have the following correspondence: (1) an object to a logical formula, and (2) a recombination to an application of inference rule. From this viewpoint, a particular execution of a system is regarded as a process of forward reasoning (like sequent calculus); this view has a good agreement with the present approach as backward reasoning.

The procedure cannot only answer whether a specified molecule is produced in the given system, but can also compute required molecules to produce it if wildcards are used in the patterns in H_0; details and examples are omitted due to page limitation.

Application of this approach to real-world problems would be as follows. As this artificial chemistry can describe natural biochemical systems [4] , reasoning about such systems might become possible. For example, it might be able to answer questions such as "can this substance be produced in this cell?", "what DNA sequence is required to produce this set of proteins?", "what pathway produces this substance?", and so forth.

7 Concluding remarks

In this paper we presented an approach to reasoning in an artificial chemistry. The procedure of reasoning tries to answer whether specified objects can be produced in a given setup, or what objects are required to produce the given objects. The procedure is similar to backward reasoning in logic. We showed example reasoning using a simple system described in the artificial chemistry. Though the given method is not yet efficient enough, we believe future enhancements will make this approach possible to deal with problems in the real world.

References

1) P. Dittrich, J. Ziegler, and W. Banzhaf. Artificial chemistries — a review. *Artificial Life*, 7(3):225–275, 2001.

2) W. Fontana and L. W. Buss. The barrier of objects: From dynamical systems to bounded organizations. In J. L. Casti and A. Karlqvist, editors, *Boundaries and Barriers: On the Limits to Scientific Knowledge*, chapter 4, pages 55–115. Perseus Books, Reading, Massachusetts, USA, 1996.

3) J.-Y. Girard. Linear logic: Its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, number 222 in London Mathematical Society Lecture Note Series. Cambridge University Press, 1995.

4) Y. Suzuki and K. Tominaga. Describing metabolic pathways using an artificial chemistry based on pattern matching and recombination. In M. Sugisaka and H. Tanaka, editors, *Proceedings of the 11th International Symposium on Artificial Life and Robotics (AROB '06)*, 2006. Published as CD-ROM.

5) K. Tominaga. A formal model based on affinity among elements for describing behavior of complex systems. Technical Report UIUCDCS-R-2004-2413, Department of Computer Science, University of Illinois at Urbana-Champaign, Mar. 2004.