Particular Results for Variants of P Systems with One Catalyst in One Membrane

Rudolf Freund

Faculty of Informatics
Vienna University of Technology
Favoritenstr. 9–11, A–1040 Vienna, Austria
rudi@emcc.at

Summary. Purely catalytic P systems can generate all recursively enumerable sets of natural numbers with only three catalysts in one membrane, whereas we know that one catalyst in one membrane is not enough. On the other hand, P systems also allowing (non-catalytic) non-cooperative evolution rules with only two catalysts in one membrane are already computationally complete, too. We here investigate special variants of P systems with only one catalyst in one membrane that are not computationally complete, i.e., variants of P systems with only one catalyst in one membrane that cannot generate all recursively enumerable sets of natural numbers.

1 Introduction

P systems with catalysts were already introduced in the original paper on membrane systems in [10], with the additional feature of using priority relations on the rules and proved to be computationally complete; in [15], [16] it was shown that priority relations on the rules are not necessary to obtain this completeness result. In [7] it finally was shown that P systems with only one membrane can generate any recursively enumerable set of natural numbers (when not counting the catalysts in the membrane) with only two catalysts. On the other hand, in [8] [purely] catalytic P systems (where all evolution rules are catalytic ones) were introduced and from results obtained in [6], [15] it was observed that seven catalysts are enough if we only allow rules with catalysts; in [7], even three catalysts were shown to be sufficient. As in [8] catalytic P systems with one catalyst in one membrane were shown not to be able to generate all recursively enumerable sets of natural numbers, the main open questions remaining are the computational power of catalytic P systems with two catalysts in one membrane and the corresponding question for P systems with one catalyst in one membrane (obviously, P systems with no catalyst and only non-catalytic non-cooperative rules in one membrane cannot generate all recursively enumerable sets of natural numbers; in fact, as an immediate consequence of Theorem 1 we see that exactly the regular sets can be generated).

In the following section, after some prerequisites from formal language theory, we define the specific variants of P systems with catalysts considered in this paper. In the main part of the paper we show particular results proving that specific variants of P systems with one catalyst in one membrane cannot generate all recursively enumerable sets of natural numbers; unfortunately, the main problems, i.e., the computational power of P systems with one catalyst in one membrane and of catalytic P systems with two catalysts in one membrane remain unsolved.

2 Definitions

For well-known notions and basic results from the theory of formal languages, the reader is referred to [4] and [14]. We only give some basic notations first. By $\mathbb{N}$ we denote the set of non-negative integers (i.e., natural numbers). In the following, two sets of natural numbers are considered to be equal if they only differ at most by 0. By NRE we denote the set of all recursively enumerable sets of natural numbers, and by $NREG$ the set of all regular sets of natural numbers.

2.1 Multiset Grammars

In this paper we consider several types of multiset grammars (various interesting models of grammars for generating multisets of objects were considered in [9], multiset automata were investigated in [3]). In fact, we could also define the corresponding types of grammars generating strings which then are interpreted as multisets.

In the following, a multiset of objects will simply be represented by any string containing exactly the same number of each object as the multiset.

A *multiset grammar* is a construct

$$G = (V, T, P, S)$$

with V being a set of *symbols* (or *objects*), $T \subset V$ is the set of *terminal symbols (terminal objects)*, $S \in V - T$ is the *start symbol*, and P is a (finite) set of *multiset productions* of the form $u \to v$ with u and v being multisets over V. The application of the production $u \to v$ to a multiset x has the effect of replacing the multiset u contained in x by the multiset v. We consider all derivations starting from the multiset S and using productions from P and finally yielding a terminal multiset (i.e., a multiset only consisting of objects from T); the set of terminal multisets generated in that way is denoted by $L(G)$.

If all productions in P are of the form $A \to v$ with $A \in V$, then G is called context-free.

A *random context multiset grammar* is a construct

$$G = (V, T, R, S)$$

with the rules in R being of the form $A \to v|\,(P,Q)$ where $A \to v$ is a context-free multiset production and P and Q are disjoint subsets of V; the application of the rule to a multiset x has the effect of replacing an object A from x by the multiset v, but only if all objects from P appear in x *(permitting contexts)* and no object from Q occurs in x *(forbidden contexts)*. Again we consider all terminal derivations starting from S and using rules from R, and we denote the set of terminal multisets generated in that way by $L(G)$. If all the rules in R are of the form $A \to v|\,(\emptyset, Q)$, i.e., we only have forbidden contexts, then G is called a *forbidden random context multiset grammar.*

As multisets over a one-letter alphabet V correspond with natural numbers, the set of multisets $L(G)$ generated by a (random context) multiset grammar with a one-letter terminal alphabet can be seen as the corresponding set of natural numbers, and we denote the sets of natural numbers generated by context-free multiset grammars, random context multiset grammars, and forbidden random context multiset grammars by NCF, NRC, and $NfRC$, respectively. It is well known (e.g., see [9]) that $NCF = NREG$; moreover, from the results known for the corresponding families of string languages, we immediately infer $NfRC \subsetneq NRE$ (e.g., see [5]).

2.2 The Standard Model of (Catalytic) P Systems

The standard type of membrane systems (P systems) has been studied in many papers and several monographs; we refer to [2], [5], [10], [11], and [12] for motivation and examples. In the definition of the P system below we omit all ingredients (like priority relations on the rules) not needed in the following.

A *P system (of degree d, $d \geq 1$)* is a construct

$$\Pi = (V, C, \mu, w_1, \ldots, w_d, R_1, \ldots, R_d, i_o),$$

where:

(i) V is an alphabet; its elements are called *objects*;
(ii) $C \subseteq V$ is a set of *catalysts*;
(iii) μ is a *membrane structure* consisting of d membranes (usually labeled with i and represented by corresponding brackets $[_i$ and $]_i$, $1 \leq i \leq d$);
(iv) w_i, $1 \leq i \leq d$, are strings over V associated with the regions $1, 2, \ldots, d$ of μ; they represent multisets of objects present in the regions of μ (the multiplicity of a symbol in a region is given by the number of occurrences of this symbol in the string corresponding to that region);
(v) R_i, $1 \leq i \leq d$, are finite sets of *evolution rules* over V associated with the regions $1, 2, \ldots, d$ of μ; these evolution rules are of the forms $a \to v$ or $ca \to cv$, where c is a catalyst, a is an object from $V - C$, and v is a string from $((V - C) \times \{here, out, in\})^*$;

(vi)i_o is a number between 1 and d and it specifies the *output* membrane of Π.

The membrane structure and the multisets represented by w_i, $1 \leq i \leq d$, in Π constitute the *initial configuration* of the system. A transition between configurations is governed by an application of the evolution rules which is done in parallel: all objects, from all membranes, which *can be* the subject of local evolution rules *have to* evolve simultaneously (maximally parallel derivation mode).

The application of a rule $u \rightarrow v$ in a region containing a multiset M results in subtracting from M the multiset identified by u, and then in adding the multiset identified by v. The objects can eventually be transported through membranes due to the targets *in* and *out* (we usually omit the target *here*). We refer to [2] and [12] for further details and examples. According to [8], the P system Π is called *catalytic,* if every evolution rule involves a catalyst.

The system continues maximally parallel derivation steps until there remain no applicable rules in any region of Π; then the system halts. We consider the number of objects from V contained in the output membrane i_o at the moment when the system halts as the *result* of the underlying computation of Π, but *in contrast to the original definitions we do not take into account the catalysts.* The set of results of all halting computations possible in Π is denoted by $N(\Pi)$. The set of all sets of natural numbers computable by P systems/(purely) catalytic P systems (as the numbers of objects different from the catalysts to be found in the output membrane i_o at the end of halting computations) of the above type with d membranes and the set of catalysts containing at most m elements is denoted by $N_{-c}OP_d(cat_m)$ and $N_{-c}OP_d(pcat_m)$, respectively.

2.3 Specific Variants of P Systems

The results we elaborate in the following section are formulated for P systems with only one membrane, hence, we can omit the membrane structure and describe the P system in the following way, where w denotes the initial multiset and R the set of rules in the membrane:

$$\Pi = (V, C, w, R).$$

Now let us consider P systems of that kind where in addition the set V is divided into three disjoint subsets V', V'', and V''' such that the rules in R obey to the following constraints:

- for $a \in V'$, there exists no rule in R with a on the left-hand side;
- for $a \in V''$, the rules in R are of the form $ca \rightarrow cv$ with $c \in C$ and $v \in (V' \cup V'')^*$;
- for $a \in V'''$, the rules in R are of the form $a \rightarrow v$ with $v \in (V' \cup V''')^*$.

P systems fulfilling all the requirements stated above are called 1-*separated*, and the corresponding family of sets of natural numbers generated by such P systems is denoted by $N_{-c}OP_1(sepcat_m)$.

The main idea of 1-separated P systems is that catalytic rules and non-catalytic rules are separated with respect to the symbols they generate and therefore cannot interfere, i.e., the objects generated by catalytic rules cannot be affected by non-catalytic rules and vice versa.

If we relax the third condition stated above for 1-separated P systems and allow the non-catalytic rules to be of the most general form $a \to v$ with $v \in V^*$, then these P systems are called *weakly* 1-*separated* and the corresponding set of sets of natural numbers generated by such P systems is denoted by $N_{-c}OP_1\,(wsepcat_m)$.

The only restriction remaining in the case of weakly 1-separated P systems is that "catalytic objects" $a \in V''$ cannot generate "non-catalytic objects" $a \in V'''$, yet this feature already will allow us to show that $N_{-c}OP_1\,(wsepcat_m) \subsetneqq NRE$ (see Theorem 1).

Another quite simple restriction is to require that there is no object such that the only rules affecting this object are catalytic ones. This requirement guarantees that if a symbol cannot be taken by a catalytic rule then it will be affected by a non-catalytic rule (observe that we are working in the maximally parallel derivation mode). We shall call P systems with catalysts in only one membrane obeying to this condition 1-*complete*. The corresponding set of sets of natural numbers generated by 1-complete P systems is denoted by $N_{-c}OP_1\,(complcat_m)$.

3 Results

We first cite some well-known results from [7]:

$$N_{-c}OP_1\,(cat_2) = N_{-c}OP_1\,(pcat_3) = NRE.$$

Moreover, from [8] we know that $N_{-c}OP_1\,(pcat_1) \subsetneqq NRE$, hence, for catalytic P systems, the main open question remained if $N_{-c}OP_1\,(pcat_2)$ were already computationally complete, too. In the following, we now investigate the generative power of the specific variants of P systems with one catalyst in one membrane as defined in the preceding section.

Theorem 1 $NREG = N_{-c}OP_1(cat_0) = N_{-c}OP_1(sepcat_1) = N_{-c}OP_1(wsepcat_1)$.

Proof. As $N_{-c}OP_1\,(cat_0) \supseteqq NREG$ is obvious and well-known and, moreover, the inclusions

$$N_{-c}OP_1\,(cat_0) \subseteqq N_{-c}OP_1\,(wsepcat_1) \subseteqq N_{-c}OP_1\,(sepcat_1)$$

are clear from the definitions, we only have to prove $N_{-c}OP_1\,(wsepcat_1) \subseteqq NREG$.

Now let $\Pi = (V, \{c\}, w, R)$ be a weakly 1-separated P system with only one catalyst c and with the rules in R fulfilling the condition that the objects appearing on the right-hand side v of a catalytic rule $ca \to cv$ can only be affected by a catalytic rule. Moreover, according to the definition of weakly 1-separated P systems given above, the set V is divided into three disjoint subsets V', V'', and V''' such that the rules in R obey to the following constraints:

- for $a \in V'$, there exists no rule in R with a on the left-hand side;
- for $a \in V''$, the rules in R are of the form $ca \rightarrow cv$ with $c \in C$ and $v \in (V' \cup V'')^*$;
- for $a \in V'''$, the rules in R are of the form $a \rightarrow v$ with $v \in V^*$.

Then we construct a context-free multiset grammar

$$G = ((V - V') \cup \{S\}, \{d\}, P, S)$$

with d, S being new objects not contained in V and P containing the following multiset productions

- $S \rightarrow h(w)$;
- $a \rightarrow h(v)$, for $a \in V''$ and the rule $ca \rightarrow cv$ in R with $c \in C$ and $v \in (V' \cup V'')^*$;
- $a \rightarrow h(v)$, for $a \in V'''$ and the rule $a \rightarrow v$ in R with $v \in V^*$;

the morphism $h : V \rightarrow \{d\} \cup V'' \cup V'''$ is defined by

- $h(a) = d$ for $a \in V'$, and
- $h(a) = a$ for $a \in V'' \cup V'''$.

Obviously, G generates the same results as Π: by the definition of weakly 1-separated P systems, the objects generated by catalytic rules can only be affected by catalytic rules; for the application of the catalytic rules the maximally parallel derivation mode has no regulating effect, because all the possible evolutions of the catalytic objects from V'', even those generated at some time by the non-catalytic rules, can be simulated sequentially by the multiset grammar G; on the other hand, as we only consider halting computations in Π, the possible evolutions of the non-catalytic objects from V''' in Π can be simulated sequentially in G, too, because there is no regulating interplay with the objects from V''. On the other hand, each derivation in G yields a result – represented by the number of terminal symbols d – that can also be obtained as the result – represented by the number of terminal objects from V' – of a halting maximally parallel derivation in Π. These observations complete the proof. □

Theorem 2 $N_{-c}OP_1(complcat_1) \subsetneqq NfRC$.

Proof. We start with the 1-complete P system $\Pi = (V, \{c\}, w, R)$, i.e., there is no object in V such that the only rules in R affecting this object are catalytic rules (observe that there must be symbols that are not affected by any rule, because otherwise no halting derivation would be possible), and then construct a forbidden random context multiset grammar

$$\begin{aligned} G &= (V \cup V_1' \cup V_1'' \cup \{S, X, Y, Z\}, \{d\}, P, S), \\ V_0 &= \{b \mid b \in V, \text{ there is no rule } cb \rightarrow cv \in R \\ &\qquad\qquad \text{and no rule } b \rightarrow v \in R\}, \\ V_1 &= V - V_0, \\ V_1' &= \{b' \mid b \in V_1\}, \\ V_1'' &= \{b'' \mid b \in V_1\}, \end{aligned}$$

with d, S, X, Y, Z being new symbols not contained in $V \cup V_1' \cup V_1''$, and P containing the following rules (as we are using only rules with forbidden context, we simply write $A \rightarrow v|Q$ instead of $A \rightarrow v|\,(P, Q)$):

1. $S \rightarrow h_0\,(w)\,X|\emptyset$;
   h_0 is the morphism $h_0 : V \rightarrow V_1 \cup \{d\}$ with
   - $h_0\,(b) = b$ for $b \in V_1$ and
   - $h_0\,(b) = d$ for $b \in V_0$

   (we start with the control symbol X and the morphic image of the axiom w where all terminal objects $b \in V_0$ from Π are replaced by the single terminal symbol d from G);
2. $a \rightarrow a''|V_1' \cup V_1'' \cup \{S, Y, Z\}$ for every a with a rule $ca \rightarrow cv \in R$
   (such a rule can only be applied once in the presence of the control symbol X);
3. $X \rightarrow Y|V_1' \cup (V_1'' - \{a''\}) \cup \{S, Y, Z\}$ for every a with a rule $ca \rightarrow cv \in R$
   (after having introduced at most one symbol a'' in the presence of the control symbol X, the control symbol is changed from X to Y);
4. $a'' \rightarrow h(v)|V_1' \cup (V_1'' - \{a''\}) \cup \{S, X, Z\}$ for every a with a rule $ca \rightarrow cv \in R$ and
   $a \rightarrow h(v)|V_1'' \cup \{S, X, Z\}$ for every a with a rule $a \rightarrow v \in R$;
   h is the morphism $h : V \rightarrow V_1' \cup \{d\}$ defined by
   - $h\,(b) = b'$ for $b \in V_1$, and
   - $h\,(b) = d$ for $b \in V_0$

   (in the presence of the control symbol Y, first eventually the single symbol a'' has to be affected by simulating one of the corresponding catalytic rules and then for all other symbols $b \in V_1$ a non-catalytic evolution rule is simulated thereby replacing every terminal object $b \in V_0$ by the terminal symbol d);
5. $Y \rightarrow Z|V_1 \cup V_1'' \cup \{S, X, Z\}$
   (if all symbols from V_1 are primed – after having applied at most one rule for a marked symbol a'' and rules for every other symbol $b \in V_1$ – we change the control symbol from Y to Z);
6. $a' \rightarrow a|V_1'' \cup \{S, X, Y\}$ for every $a \in V_1$
   (we then rename the primed symbols to the corresponding non-primed ones until finally we get a multiset of symbols over $V_1 \cup \{d\}$ and the control symbol Z);
7. $Z \rightarrow X|V_1' \cup V_1'' \cup \{S, X, Y\}$
   (we now may start again with simulating another derivation step in Π by changing the control symbol from Z to X, or we may stop the derivation by using the next rule);
8. $Z \rightarrow \lambda|V_1 \cup V_1' \cup V_1'' \cup \{S, X, Y\}$
   (if the multiset only consists of terminal symbols d, we may erase the control symbol Z thus obtaining the same result for the number of terminal symbols as in the halting derivation in Π – there counting the number of objects from V_0).

The reader should observe that we can mark (with two primes) at most one symbol, but on the other hand we cannot enforce that this one symbol is marked in any case, which is the main reason why we have to demand the P system to be complete, i.e., if we do not simulate the catalytic rule, then the symbol will be affected by simulating a non-catalytic rule (which has to exist because of the completeness of the P system). In fact, it is easy to see that G simulates exactly the derivations in Π. □

Instead of solving the original problem concerning the computational power of P systems with one catalyst in one membrane, the results proved above give rise to new interesting problems, e.g., we may investigate the generative power of complete or (weakly) separated P systems with more than one catalyst in possibly more than one membrane.

4 Conclusion

In this paper, specific variants of P systems with one catalyst in one membrane were shown not to be able to generate all recursively enumerable sets of natural numbers. For the original variant of P systems with one catalyst in one membrane, we also conjecture a similar result, but unfortunately were not able to prove it. The problem of the generative power of purely catalytic P systems with two catalysts in one membrane remains open, too.

References

1. C.S. Calude, Gh. Păun, G. Rozenberg, A. Salomaa, eds.: *Multiset Processing – Mathematical, Computer Science and Molecular Computing Points of View*. LNCS 223, Springer, Berlin, 2001.
2. C.S. Calude, Gh. Păun: *Computing with Cells and Atoms*. Taylor & Francis, London, 2001.
3. E. Csuhaj-Varjú, C. Martín-Vide, V. Mitrana: Multiset automata. In [1], 69–84.
4. J. Dassow, Gh. Păun: *Regulated Rewriting in Formal Language Theory*. Springer, Berlin, 1989.
5. J. Dassow, Gh. Păun: On the power of membrane computing. *J. of Universal Comput. Sci.*, 5, 2 (1999), 33–49.
6. R. Freund, M. Oswald, P. Sosík: Reducing the number of catalysts needed in computationally universal systems without priorities. In *Fifth International Workshop Descriptional Complexity of Formal Systems* (E. Csuhaj-Varjú, C. Kintala, D. Wotschke, Gy. Vaszil, eds.), Budapest, Hungary, July 12-14, 2003, MTA SZTAKI, Budapest, 2003, 102–113.
7. R. Freund, L. Kari, M. Oswald, P. Sosík: Computationally universal P systems without priorities: two catalysts are sufficient. *Theoretical Computer Science*, 330, 2 (2005), 251–266.

8. Z. Dang, O. Egecioglu, O.H. Ibarra, G. Saxena: Characterizations of catalytic membrane computing systems. In *28th International Symposium on Mathematical Foundations of Computer Science 2003,* MFCS 2003, Bratislava, Slovakia, August 25-29, 2003 (B. Rovan, P. Vojtás, eds.), LNCS 2747, Springer, Berlin, 2003, 480–489.
9. M. Kudlek, C. Martín-Vide, Gh. Păun: Toward a formal macroset theory. In [1], 123–134.
10. Gh. Păun: Computing with membranes. *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143.
11. Gh. Păun: Computing with membranes: an introduction. *Bulletin EATCS*, 67 (1999), 139–152.
12. Gh. Păun: *Membrane Computing: An Introduction.* Springer, Berlin, 2002.
13. Gh. Păun, G. Rozenberg, A. Salomaa, C. Zandron, eds.: *Membrane Computing. International Workshop, WMC-CdeA 2002*, Curtea de Argeş, Romania, August 2002, LNCS 2597, Springer, Berlin, 2003.
14. G. Rozenberg, A. Salomaa, eds.: *Handbook of Formal Languages*, Springer, Berlin, 1997.
15. P. Sosík, R. Freund: P Systems without priorities are computationally universal. In [13], 400–409.
16. P. Sosík: The power of catalysts and priorities in membrane systems. *Grammars*, 6, 1 (2003), 13–24.