Point Mutation Colonies with Restricted Rules

Adam Kožaný
Institute of Computer Science, Faculty of Philosophy and Sciences, Silesian University in Opava, Bezručovo náměstí 13, Opava, Czech Republic

Keywords: PM Colony, Artificial Life, Multi-agent Systems.

Abstract: Point mutation colonies (hereinafter referred to as PM colonies) are multi-agent systems. Development of the environment in these systems is determined by rewriting rules which allow the agent to influence other agents and environmental symbols in its strict neighbourhood. The rules enable the agent to erase, substitute or insert neighbouring agents/symbols, to change its position with neighbouring agents/symbols or to disappear. In this paper we will focus on the impact of forbidding some of the rule-type or their combination in the development of the entire family of PM colonies with such restriction and we will also look into the impact of restrictions on the generative power of PM colonies.

1 INTRODUCTION

PM colonies were introduced in two papers (Martín-Vide and Paun, 1998). This type of colonies (Csuhaj-Varjú and Kelemenová, 1992) is motivated by biology, or more precisely by communities of organisms living in a common environment. In the field of informatics this kind of co-existence represents a multi-agent system.

In (Martín-Vide and Paun, 1998) there was an open question of decidability results flowing into the necessity of studying restricted classes of PM colonies. Similar topics were studied in (Kelemenová, 2002), (Kožaný, 2009) and (Kožaný, 2010).

In this paper we are restricting the set of possible rule-types in PM colonies. Then we discuss decidability problem if two agents can reach a conflict in a given PM colony from a given starting string. We will also discuss the generative capacity of such PM colonies.

2 PM COLONIES

Colonies are grammar models of multi-agent systems motivated by subsumptial architecture and they are characterised as special forms of cooperating grammars. A colony consists of a finite number of simple components (agents) each generating a finite language. More about grammar systems and especially about colonies is presented in (Harrison, 1978) and in (Csuhaj-Varjú et al., 1994).

Environment in colonies is represented by a string of symbols, and it is influenced via components which make changes in it. The set of all possible states of the environment, which can be generated from a given starting string, forms the language of the colony.

In a PM colony environment the locations of agents are fixed. The area, where the PM colony works, is represented by a string of agents and environment symbols (which can be changed) and boundary markers of the environment. Boundary markers label the beginning and the end of a word, it is not allowed to erase them, to overstep or to produce them.

The actions take place only in strict vicinity of the symbol representing the agent. Each action can add one environment symbol or one agent symbol (a new agent can be also created), can move agent one step to the left or right, can erase neighbouring agent or environment symbol, or can substitute an environment symbol to another one.

All agents work in parallel. The activity of an agent depends totally on one symbol in front of it and one symbol behind it. To solve a conflict, when agents have a common neighbour, we arrange the set of agents by a priority relation. An agent can‘t change its own name or name of any other agent. Agents with the same name may be present on more than one position in the string. Formally:

Kožaný A..
Point Mutation Colonies with Restricted Rules.
DOI: 10.5220/0004180902450249
In *Proceedings of the 4th International Joint Conference on Computational Intelligence* (ECTA-2012), pages 245-249
ISBN: 978-989-8565-33-4

Definition 1. *PM colony* is a construct $C = (E, \#, N, >, R_1, ..., R_n)$, where

- E is the alphabet of the environment,
- # is the boundary marker,
- N is the alphabet of agents names,
- $>$ is the partial order relation over N (the priority relation for agents),
- $R_1, ..., R_n$ are finite sets of action rules for agents from N. The action rules can be of the following forms:
- Deletion:

$(a, A_i, b) \to (\varepsilon, A_i, b)$, where $a \in E \cup N$, $b \in E \cup N \cup \{\#\}$,

$(a, A_i, b) \to (a, A_i, \varepsilon)$, where $a \in E \cup N \cup \{\#\}$, $b \in E \cup N$,

- Insertion:

$(a, A_i, b) \to (a, c, A_i, b)$, where $a, b \in E \cup N \cup \{\#\}$, $c \in E \cup N$,

$(a, A_i, b) \to (a, A_i, c, b)$, where $a, b \in E \cup N \cup \{\#\}$, $c \in E \cup N$,

- Substitution:

$(a, A_i, b) \to (c, A_i, b)$, where $b \in E \cup N \cup \{\#\}$, $a, c \in E$,

$(a, A_i, b) \to (a, A_i, c)$, where $a \in E \cup N \cup \{\#\}$, $b, c \in E$,

- Move:

$(a, A_i, b) \to (A_i, a, b)$, where $a \in E \cup N$, $b \in E \cup N \cup \{\#\}$,

$(a, A_i, b) \to (a, b, A_i)$, where $a \in E \cup N \cup \{\#\}$, $b \in E \cup N$,

- Death:

$(a, A_i, b) \to (a, \varepsilon, b)$, where $a, b \in E \cup N \cup \{\#\}$.

Let $(a, A_i, b) \to \alpha$ be an action rule of an agent, then symbols a, b represent the context of agent A_i.

PM colonies are devices, where agents work parallel. Similarly as in the other parallel working systems, conflicts can occur between agents.

Definition 2. If in a word $w \in (E \cup N)^*$ context overlay of two agents A_i and A_j happens or if agent A_j takes part in context of agent A_i, we call it *direct conflict* between agents. If in w the pairs of agents (A_1, A_2), (A_2, A_3)... (A_n, A_{n+1}) are in direct conflict then the whole set of agents $A_1, A_2, A_3, ..., A_n, A_{n+1}$ are in conflict.

The conflict of agents $A_1, A_2, A_3, ..., A_n, A_{n+1}$ in PM colony can be solved by the agent with the greatest priority, which takes action. So, to solve the conflict, conflicting agents have to be ordered in such a way, that there is an agent with priority higher then all other agents in the conflict. Moreover the agent with the greatest priority occurs in the conflict set only once.

Definition 3. A *configuration* in a PM colony C is a string $\#w\#$, where $w \in (E \cup N)^*$.

Let A be its agent and $\#w\# = xaAby$ be a configuration in C, where $a,b \in (E \cup N) \cup \{\#\}$. This occurrence of agent A is active with respect to configuration $\#w\#$, if (1) in C an action rule exists, whose left side is in the form (a, A, b), and (2) A is not conflicting with any other agent occurrence, or A has the highest priority from all agents from those in conflict.

An agent occurrence is inactive, if it is not active.

Definition 4. A *derivation step* in a PM colony denoted as $\Rightarrow$ is a binary relation on a set of configurations. We write $\#w\# \Rightarrow \#z\#$ if and only if each active agent A in the string w replaces its context in w by corresponding rule and the resultant string is $\#z\#$. Derivation $\Rightarrow^*$ is the reflexive and transitive closure of relation $\Rightarrow$.

Definition 5. *Deterministic PM colony* is such PM colony where each agent A has for any context (a, A, b) at most one action rule.

3 (UN)DECIDABILITY RESULTS IN PM COLONIES

In (Martín-Vide and Paun, 1998) there are several problems focused on decidability mentioned. To explain those problems we have to mention some structural properties of PM colonies, which determine structures in environmental states introduced in the work above.

Definition 6. Let $C = (E, \#, N, >, R_1, .., R_n)$ be a PM colony. A state $y \in \#(E \cup N)^*\#$ is *reachable* in C if there is a state $z \neq y$ such that $z \Rightarrow y$ with respect to C. A state which is not reachable is said to be *unreachable*. A state $y \in \#(E \cup N)^*\#$ is said to be *alive* if there is a state $z \neq y$ such that $y \Rightarrow z$. A state which is not alive is said to be *dead*.

By intersecting the classes in the two classifications above, we get four classes of states. We denote by *Reachable(C), Unreachable(C), Alive(C), Dead(C)* the languages of all reachable, unreachable, alive, and dead states, respectively, with respect to C. We also denote:

Garden-of-Eden(C) = Unreachable(C) ∩ Alive(C)
Life(C) = Reachable(C) ∩ Alive(C)
Doomsday(C) = Reachable(C) ∩ Dead(C)
Non-Life(C) = Unreachable(C) ∩ Dead(C)

Proposition 1 (Martín-Vide and Paun, 1998)**.** All the languages *Reachable(C), Unreachable(C), Alive(C), Dead(C), Garden-of-Eden(C), Life(C),* *Doomsday(C), Non-Life(C)* are regular for PM colony *C*.

This proposition implies that most problems about the mentioned languages are decidable. For instance, one can decide whether or not they are empty, finite or infinite, equal to any given regular language, included or including any given regular language.

The previous decidability results hold with respect to all states which are alive, reachable, dead, etc, with respect to any given PM colony. Let us assume PM colony and its starting string ω. We can relate the reachable strings to be reachable from ω and use following modifications of the above structures

$\omega Reachable(C) = \{u: \omega \Rightarrow_C^* u\} \cap Reachable(C)$.

This leads to a corresponding *ωGarden-of-Eden(C), ωLife(C), ωDoomsday(C), ωNon-Life(C)*.

When we consider the same problems with respect to these *ω*structures, then the results are quite opposite, most problems are no longer algorithmically solvable. We consider several problems with such a status and which are of a clear interest for predicting the development of a colony: will a given agent become sometimes active, does the colony reach a state when a conflict appears, does the colony enter a deadlock? Unfortunately, as we have mentioned above, if the starting state is prescribed, these problems (and others similar) are undecidable.

Proposition 2 (Martín-Vide and Paun, 1998)**.** Given a PM colony *C*, an agent A_i, and a state *w*, we cannot decide whether or not a state *z* can be derived from *w* with respect to *C* such that the agent A_i is active on *z*.

4 DECIDABILITY RESULTS IN PM COLONIES WITH RESTRICTED RULE-SETS

Problems mentioned in the paragraph above can be represented as a group of problems on the same base. (Un)Decidability status of any one of these problems can be quiet easily transformed to that one of any other problem from that group. The questions we are dealing with are: “Will a given agent in a PM colony with given initial state become active?” “Will a PM colony reach a state from which it is not possible to continue in its development?” “Will a PM colony reach a state in which a conflict happens?” In (Martín-Vide and Paun, 1998) authors indicate, that problem can become solvable when we consider modifications in the PM colony. By the change it can be understood application of different kind of restrictions on the PM colony: determinism, restricted rule-set or reduced parallelism.

In this paper we decided to investigate decidability results in PM colonies with restricted rule-set. From the group of decidability problems we selected the one considering if a PM colony will reach a state in which a conflict happens: “Is in a PM colony *C* with a given initial string w_0 the problem if two agents *A* and *B* will reach conflict decidable?” This problem we want to discuss on PM colonies with specific restrictions in its sets of rules.

In PM colonies there exists five types of rules. As a restricted rule-set we consider each set of rules, where there is at least one rule-type missing. In the subsections we will study PM colonies with no deletion and move as well as PM colonies with no insertion.

4.1 PM Colony No Deletion Rules and No Moving

Assume PM colonies, where moving of agents and deletion of the agents and environment is forbidden. The derived environment in such a system cannot be reduced even if the agent itself can die. New agents can appear but the mutual positions of the already existing agents do not change.

Theorem 1. In a PM colony C with no deletion rules and with no rules for move and with a starting string w_0 it is decidable that two agents *A, B* will reach a conflict.

Proof. Consider an algorithm simulating the development of given (deterministic) PM colony. The inputs of the algorithm are: C = $(E, \#, N, >, R_1, .., R_n)$, starting string w_0 and agents *A* and *B.* We have to consider the longest substring of conflicting agents in the starting string – we denote it by *s*. Outputs of the algorithm are messages if conflict happens or not, number of derivation steps and reached string (state of the colony).

The agents *A* and *B* will enter the conflict if $w_0 \Rightarrow^* uAvBw \Rightarrow^* u'Av'Bw'$. In the string *uAvBw* there was at least one of agents *A, B* inserted during the last derivation step. In the next derivation step the agent cannot be rewriten or erased, but it still has not to be in conflict with the second agent even during the next derivation step. In the string *u′Av′Bw′* agents *A* and *B* are in conflict for the first time. For these cases there must exist variables *k, l* such that $w_0 \Rightarrow^k uAvBw$ a $uAvBw \Rightarrow^l u'Av'Bw'$ (*k, l* depends on properties of PM colony *C* and on its starting string w_0).

In PM colony without rules for erasing agent or environmental symbol and without rule for moving of agent, we consider an algorithm, that for $k+l$ derivating steps simulates the development of the colony. We have to determine the value of k and l (respectively $k+l$).

Consider deterministic PM colony first. We assume that the conflict between agents *A* a *B* appears in finite number of derivation steps (the exact number depends on properties of *C* and w_0). If the conflict does not appear in number of derivation steps counted below, then it does not appear at all (all the possible parts of string allowing changes in the development of the colony – originating from w_0 – will be exhausted by changes caused by development of the colony and existing agents will repeat the same actions in cycles). Awaited output of the algorithm is a message telling if conflict happens or not.

This would be attended with information about number of done derivation steps and reached string. In case of development not reaching conflict of specified agents includes output message the last algorithmically reached string, number of derivation steps done and message that conflict would not come.

When analysing development of PM colony from its initial string we are interested into the whole string w_0. In the colony, there can be some parts of the string causing complicated development (e. g. collision of more agents brings complications with determining which agent is active and which agent will be active in the next derivation step), we have to focus on these parts. In the development plays the role parts consisting of environmental symbols only. The most complicated development can be observed in parts where more agents collide. There can be more than one such part in the starting string. In all of the parts many events can happen, but to determine the "worst possible case of development" (in the sense of highest number of derivation steps which has to be done to determine if the conflict between agents *A* and *B* arrive) we have to consider the longest string of conflicting agents. Considering the rules in colony it is pointless to think about situation where two short conflicting strings become one longer (such rule-type is in this type of PM colony forbidden).

In this type of PM colony are only these rule-types: substitution of environmental symbol, insertion of an agent of environment symbol, death of an agent. Only two types of rules can produce the conflict 1.) death of an agent and 2.) insertion of an agent.

When considering possibility of conflict due to rule for agent insertion, we have to consider the priority relation <. This rule-type can cause conflict during $a*a^2$ derivation steps, where a is the number of agent names in colony (each agent can produce up to a agents on both is sides, but then the actions are repeating. Repeated is also the whole life of the colony and nothing new can happen).

In the case when conflict appears due to the rule for death of an agent we have to consider s – the longest substring of conflicting agents in the starting string. In this case agents can create new agents or to exchange environmental symbols on both its sides. It means $(e+a)^2$ possible combinations of development. If the conflict should appear, then it has to happen no later than in $s*a*(e+a)^2$ derivation steps and this is the value matching to $k+l$. If the conflict does not appear in this number of derivation steps, then it does not appear in this colony with given starting string at all.

Running of this simulation gives sense only when in colony exists a rule generating new copy of agent *A* or agent *B* or in case of existing rule for erasing an agent neighboring with any of these agents.

In case of non-deterministic PM colony it is necessary to bifurcate the computation every time when it is possible to use more than one rewriting rule for any context. It is also necessary to follow all branches of computation until the number of derivation steps mentioned before. If a conflict is reached in any of the branches, the problem has a solution for given non-deterministic PM colony and given starting string. □

Note. Generative power of this restricted class of PM colonies is lower than the power of original PM colonies. The absence of a rule for deletion causes that strings cannot be shortened. Example of a language, which cannot be derived by this type of PM colonies is a set {*a, aa, aaa*}.

4.2 PM Colony with No Insertion Rules

Assume PM colonies, where insertion of agents and environment symbol is forbidden. No growth is possible in these colonies.

Theorem 2. In a PM colony with no insertion rule and with an initial string w_0 it is decidable if two agents *A, B* will reach a conflict.

Proof. In this type of PM colony the length of string cannot be prolonged. Because of finite language produced by this type of PM colony, the problem if agents *A* and *B* will enter a conflict is solvable. □

In the topic of PM colonies with restricted rule-set, there are more problems, which we are interested in. We want to explore if the problem: „is it decidable if two agents *A, B* will reach a conflict?“ is solvable in PM colony with given initial string w_0. In our „to-do list“ there remain PM colonies with no rules for

a) deletion
b) substitution
c) move and
d) death of an agent.

4.3 Note on Generative Power

With restrictions on a generative system it is always interconnected question of the impact on the generative power. The generative power with limitations on possible rewriting rules falls.

E.g. when we consider a restricted PM colony, without rule for insertion, it is impossible to generate any infinite language. A PM colony without this restriction needs only one agent and one environmental symbol to generate the infinite language a^+.

Without proof we present the theorem.

Theorem 3. Generative power of deterministic PM colonies with restricted rule-set is lower then generative power of deterministic PM colonies without any restrictions on those types of rewriting rules.

5 CONCLUSIONS

In this paper we focused on influences of restrictions in the form of reduced rule-set on decidability problems in PM colonies. As a part of the restrictions influence we explored changes in the generative power of restricted forms of PM colonies.

These restrictions give a possible algorithmic solution to the problem if two agents will enter a conflict. To find out if all suggested restrictions give algorithmic solutions it is necessary to deal with the resting – so far not solved restricted forms.

At this opportunity it is also necessary to inform, that these restrictions cause declination of generative power. The generative power of original PM colonies is higher than generative power of PM colonies influenced by restrictions introduced in this paper.

ACKNOWLEDGEMENTS

This work was supported by grant SGS/5/2010 “Síla a forma spolupráce v generativních systémech”. (The power and forms of cooperation in generative systems)

REFERENCES

Csuhaj-Varjú, E., Dassow, J., Kelemen, J., Paun, G. *Grammar Systems: a grammar approach to distribution and cooperation.* Gordon and Breach Science Publishers S. A., Yverdon.

Csuhaj-Varjú, E., Kelemenová A., 1992. On the power of colonies. In *2nd Colloquium on Words, Languages and Combinatorics*. World Scientific.

Harrison, M. A., 1978. *Introduction to Formal Language Theory*. Addison-Wesley Publishing Company, Inc. Reading.

Kelemenová, A., 2002. Colonies, In *Lecture notes*. Tarragona.

Kožaný, A., 2009. PM kolonie se zúženou sadou pravidel. In *Kognice a umělý život IX*. FPF SU.

Kožaný, A. 2010. Vliv odebrání konkrétních typů pravidel na vývoj derivace v PM koloniích. In *Kognice a umělý život X*. FPF SU.

Martín-Vide, C., Paun, G. 1998. New Topics in Colonies Theory, In *Progreedings of the MFCS '98 Satellite Workshop on Grammar Systems*. Silesian university.

Martín-Vide, C., Paun, G. 1998. PM-Colonies. *Computers and Artificial Intelligence 17*, World Academic Publishing, Inc.