Final Report on the Research Project

„Distributed models in unconventional computing”

supported by National Research, Development and Innovation Office – NKFIH

Grant. no K 120558

1. The aim of the project

Unconventional computing is a general term for models and new methods that go beyond or enrich standard models of computer science. In the research project, we planned to focus on variants of P systems or membrane systems, which are bio-inspired (unconventional) models of computation. Our main goal was to develop and study some existing models, introduce new concepts, demonstrate their usefulness and efficiency in computation processes, and thus to better understand the concept of computation. These investigations also aim to determine whether the distributed organization adds power and efficiency to the models under study. We planned to study the expressive power, complexity, and other properties of variants of P systems, especially P automata, and related models.

2. The project, main achievements, participants

Our investigations followed the goals of the project, but the research was also driven by the natural evolution of our achievements, as well as by the main trends and open problems of the related scientific fields. The research and the obtained results conform to the planned investigations, the work was performed in a competitive international scientific environment. We obtained important results, especially in the theory of P colony automata and their variants, and introduced promising new models. We also opened research directions by demonstrating connections between P systems theory and other scientific areas such as Petri net theory and the theory of reaction systems. Our results provide new knowledge on the expressive power, size, computational, and “functional” complexity of the considered constructs, and contribute to the development of the theory of membrane systems, automata theory, and computing. Our achievements obtained attention from the international scientific community. This is proved, among others, by our talks and invited talks at international workshops and conferences (see Publications). We published the obtained results in proceedings of international conferences (LNCS, Springer) and international journals (Theoretical Computer Science, Fundamenta Informaticae, Journal of Membrane Computing). One more paper is in status „to appear” (LNCS), and another paper, submitted to the journal Natural Computing, is under review.

Senior participants Erzsébet Csuhaj-Varjú and György Vaszil, as well as junior participant Gábor Kolonits, participated in the research for the entire duration of the project. After obtaining his Ph.D., junior participant Kristóf Kántor left the project, and junior participant Attila Bagossy joined the project with investigator contract for five months. Unfortunately, the COVID-19 epidemic made impossible the majority of the conferences planned for 2020, including those at which the project's senior researchers wanted to present the new results or were invited as invited speakers. The intensive work was made more difficult by the extremely increased teaching and management burden of the project's senior researchers, as well as the huge educational burden of the project's junior researcher. That is why we asked to extend the duration of the project by one year. Our application was accepted. For the same reason, we took advantage of the extension, which was made possible by government decree due to COVID-19.

3. Results

Our investigations focused on variants of P systems and related models. P systems are distributed computing devices inspired by the architecture and the functioning of living cells. The main ingredient of the basic P system is a hierarchically embedded structure of membranes; the tissue-like P systems have a network architecture. Each membrane encloses a region (a compartment) that contains multisets of objects, furthermore, the regions are associated with rules describing the evolution of the objects and their communication to other regions. The evolution of the system corresponds to a computation. P automata are P systems with purely communication rules; they combine features of classical automata and membrane systems. P systems theory is a well-established, active scientific area of unconventional computing.

3.1. P colony automata

The P colony is a finite collection of agents (or cells), represented by very simple membrane systems and their shared environment. The cell is given by a multiset of objects. In a P colony, each agent has a fixed number of objects. This number does not change in the course of the computation. The agents use programs for modifying their own objects and the objects in the environment. The programs consist of evolution rules (rules for changing their objects) and communication rules (rules for exchanging objects with the environment). A computation is a sequence of computation steps during which a maximum number of P colony members execute one of their programs in parallel until the system reaches a halting state (a state where no program can be applied).

In 2014, K. Kristóf and Gy. Vaszil introduced the concept of generalized P colony automata (genPCol automata), extensions of the P colony model to be able to characterize strings in a natural way. Certain rules (called tape rules) are designated to determine the input sequence which is read during the computation: when such a tape rule is applied, a multiset of symbols (depending on the rule) is assumed to be taken from the input. This way, the halting computations define accepted multiset sequences, which are turned into accepted strings by mapping them to symbol sequences (strings) over a previously given alphabet.

In this seminal paper some basic variants of the model were introduced and studied from the point of view of their computational power, structuring the results around the computational capacity of the systems, and different types of restrictions imposed on the use of tape rules in the programs of the systems. Three possible ways of dealing with tape rules in the programs were considered: (1) the unrestricted case, (2) the case when all programs must contain at least one tape rule (all-tape programs), and (3) the case when all communication rules are tape rules (com-tape programs).

In [Publications, item 7, 15, 16] we considered the case when the class of mappings defined by finite state transducers is allowed to be used as input mapping. We have examined the effect of the capacity (the number of objects in an agent) on the computational power. For all-tape and com-tape programs, systems with capacity one characterize a superclass of regular languages. In the unresticted case, even with capacity one, genPCol automata characterize the class of recursively enumerable languages. We have also shown that for systems with capacity at least two, all-tape programs are able to characterize the class of recursively enumerable languages, while the language class described by systems with com-tape programs is included in the class of languages that can be computed by so-called restricted one-way logarithmic space Turing machines. This class of languages is interesting, because it can be described by certain variants of standard P automata.

In [Publications, item 16], we also obtained another interesting result. Namely, genPCol automata with capacity two and with so-called permutation mappings and com-tape programs are able to accept languages that cannot be described by the above-mentioned variants of standard P automata.

We also studied the possibility of the deterministic parsing (that is, parsing without backtracking) of languages described by (generalized) P colony automata. We defined a subclass of these computing devices satisfying a property which resembles the LL(k) property of context-free grammars, and studied the possibility of parsing the characterized languages using a k symbol lookahead, as in the LL(k) parsing method for context-free languages [Publications, item 13].

In 2014, L. Ciencialová, L. Cienciala, and E. Csuhaj-Varjú introduced a variant of P colonies where the environment is given only by a string and the agents process the string similarly to automata. These constructs are called Automaton-like P colonies or APCol systems. It is known that certain variants of APCol systems are computationally complete. In [Publications, items 9, 10] we investigated the possibility of ''going beyond'' Turing by APCol systems. We used the notion of colored teams of agents as a restriction for the maximal parallelism of the computation. We have shown that we can simulate red-green counter machines with APCol systems with two-colored teams of minimal size. Red-green counter machines are computing devices with an infinite run on a finite input; their power exceeds the power of Turing machines.

In the case of standard APCol systems, the string is accepted if it can be reduced to the empty word. We studied a different mode of acceptance where the agents explore and verify their common environment. This means that an input environmental string of length n is accepted if there is a halting computation such that the length of the string remains unchanged during the computation and every agent applies at least one rule to a letter in each of the positions of the string before a halting configuration is reached. We have shown that verifying APCol systems simulate nondeterministic two-way multihead automata, thus, any language in NSPACE(log n) can be accepted by a verifying APCol system [Publications, items 17, 22]. The model and the results demonstrate that two-way multihead finite automata and the verifying APCoL systems correspond to each other.

We have also studied standard P automata, computing devices combining features of classical automata and P systems. In [Publications, item 34], we introduced a new concept, called P n-stack-automata, a restricted class of P automata that mimics the behavior of n-stack automata. We proved that for n = 1 these constructs describe the context-free language class and for n = 3 the class of quasi-realtime languages, thus demonstrating a new connection between classical automata and P automata.

We introduced a new way of representing computations in P colonies based on logical values, propositional logic, and rule-based systems. We presented the conversion of conditions of applicability of rules, programs, multisets of programs, and complete computational steps as propositional formulas in the disjunctive normal form. Using this representation, we obtained new results concerning the complexity of the execution of computational steps of a P colony. We proved that the problem of whether a certain configuration is halting is in the complexity class P or NP, depending on the type of the P colony [Publications, item 11].

In [Publications, items 4, 5], we provided a strong bi-simulation between P colonies and multi-stable (purely) catalytic P systems. Purely catalytic P systems are given with multiset rewriting rules where each rule has occurrences of distinguished symbols called catalysts. In the original model, catalysts cannot change, in the case of multi-stable catalytic P systems catalysts are allowed to change only to some other, distinguished catalysts. (Purely) catalytic P systems have intensively been studied in P systems theory. Our bi-simulation demonstrates that although the two models are formally different, one can be used to solve problems concerning the other one. We note that both computing devices are computationally complete.

We compared P colonies to kernel P systems or kP systems, models which integrate some of the most successfully used features of membrane systems. We described how the behavior of P colonies of arbitrary capacity, P colonies with senders and consumers, and P colonies with capacity two and evolving environment is mapped into equivalent kP systems. We also considered the problem of modeling the synchronization aspects occurring in a producer/consumer problem by using adequate P colonies and kP systems, exhibiting equivalent behavior [Publications, item 14].

In [Publications, item 1) we presented relationships between generalized communicating P systems and P colonies. Generalized communicating P systems (GCPSs) are networks of cells where each rule moves only two objects. In [Publications, item 6], we proved that GCPSs with three cells and with only join, or only split, or only chain rules are computationally complete computing devices. These bounds are improvements to the previous results.

We also provided a summary of the recent status of the theory of P colonies [Publications, item 23].

3.2. P systems and Petri nets

Petri nets are a well-known models of computing. We have explored several connections between P systems and time Petri nets, thus we have contributed to the development of both fields.

In [Publications, item 3] we have reported on the initial phase of research in which we relate so-called local time membrane systems to time Petri nets such that the Petri net corresponding to a given membrane system is suitable for answering questions in connection with the membrane system. We have taken the initial steps towards the investigation of a timed P system model where each rule is associated with a time interval during which the rule can be executed. We call our membrane systems local time membrane systems. There are several timed models for P systems in the literature. The attempts for the simulation, up to the present, seem to take the approach similar to timed Petri nets where certain values, the delay values, are assigned to rules. Our model, on the other hand, resembles more time Petri nets where an interval is assigned to every rule and a clock local to each compartment synchronizes when the rule can be executed.

Continuing this line of research, in [Publications, item 32] we showed that with time Petri nets, the description of the computations of standard P systems using the maximal parallel mode of rule application is possible without introducing maximal parallelism to the Petri net semantics. We also studied the so-called strong and weak semantics of time Petri nets: we defined local time P systems inspired by time Petri nets with these two types of semantics and showed that both kinds of local time P systems can be described by time Petri nets with the strong semantics. We also presented connections between catalytic Petri nets and catalytic P systems.

In [Publications, item 31], we continued the investigations of the connection between P systems and time Petri nets by extending simple symbol-object membrane systems with promoters/inhibitors, membrane dissolution, and priority for rules. By constructing the simulating time Petri net, we retained one of the main characteristics of the Petri net model, namely, the firings of the transitions can take place in any order: instead of introducing maximal parallelism in the Petri net semantics, we substantially exploited the gain in computational strength obtained by the introduction of the timing feature for Petri nets.

3.3. P systems, basic constituents, complexity

Multiset approximation spaces provide the possibility to define the lower and upper approximations of multisets, thus, the difference between the two approximations can be considered as the border or the boundary region of a multiset. Since membrane compartments are multisets, this framework can be used to define communication channels based on the idea that interaction between compartments can only involve objects which are present within the boundaries of the compartments. In [Publications, item 18] we used a similar approach to define a dynamic notion of closeness in the framework of generalized P systems. We investigated the influence of variants of the neighborhood relation on the functioning of these systems. We proved that using a simple model of chemical stability based on the set of base sets of the approximation space is not sufficient to increase the computing capabilities, but with a refined notion involving also the idea of membrane boundaries, an increase of the computational power can be obtained.

In [Publications, item 19] we also studied generalized P systems, but we only considered lower approximations with boundaries of compartments and restricted the applicability of the rules accordingly. It turns out that the use of rules which can only manipulate objects on the boundaries of compartments results in computational completeness of certain variants of generalized P systems, while the restriction of rule applications only to rules that manipulate multisets that are in the inner approximations of the membranes is not enough to obtain this power. A short version of the paper can be found in [Publications, item 21].

According to a well-known conjecture by Gh. Păun, polarizationless P systems with active membranes cannot solve NP-complete problems in polynomial time. The conjecture is proven only in special cases so far. We have made important progress in this area. We introduced the concept of object division polynomials and based on this notion we gave a new approach to simulate certain computations of a certain basic variant of P systems. Furthermore, we have shown how to compute efficiently the result of computations using these polynomials [Publications, items 26, 29].

We have also established a connection between P systems theory and a certain area of data science: clustering analysis which is a widely used tool in data mining. One of the well-known clustering algorithms is called density-based clustering of applications with noise (DBSCAN). In [Publications, item 28] we gave an implementation of this algorithm in terms of P systems, conceptually simpler than other existing ones. The system exploits the parallelism of P systems to speed up the DBSCAN algorithm which in its original version works with O(n^2) time complexity on a sequential machine (where n is the number of points to be clustered). On P systems, the running time can be reduced to O(n).

3.4. P systems and reaction systems

We studied the relations between P colonies and reaction systems (R systems). R systems are computing devices, components of which represent basic chemical reactions that take place in a shared environment given with a set. Unlike P colonies which operate with multisets of objects, R systems work with sets, thus provide a qualitative model of bio-chemical reactions. We introduced a new variant of P colonies, called sP colonies, where the environment is given by a set. In [Publications, item 33], it was shown that interactive processes in a given reaction system can be simulated by an sP colony. In [Paper, 2] we studied the reversibility in sP colonies and reaction systems. We introduced the concept of a particular variant of reversible sP colonies in which an input is provided only to initialize a process of computation. We proved that for such reversible sP colony, there exists an sP colony performing inverse computation. We also showed that a reversible reaction system and inverse reaction system can be defined in similar way and they can simulate reversible logic gates and reversible Turing machines. These results can also be interpreted for sP colonies.

Important results on reversibility in reaction systems were published in [Publications, item 30]. Our goal was to establish an Undo-Redo-Do-like semantics of reversibility with environmental control over the direction of the computation using a so-called no-memory approach, that is, without introducing modifications to the model of reaction systems itself. We established requirements the systems must satisfy in order to produce processes consisting of states with unique predecessors, and defined reversible reaction systems in terms of reversible interactive processes. For such reversible systems, we constructed simulator systems that can traverse between the states of reversible interactive processes back and forth based on the input of a special “rollback” symbol from the environment. As a continuation of research on reversibilty in reaction systems, we studied their transition graphs, and thus, the possible computational paths of reaction systems which are reversible according to different notions of reversibility. In [Publications, item 36] we proved that systems which are reversible in the sense of [Publications, item 30], see above, produce very simple types of transition graphs. As main result, we introduced the notion of reversibility with lookbehind, and showed that systems which are reversible in this sense produce the same transition graphs (and thus, the same computations) as the state transition diagrams of reversible finite transition systems.

Networks of reaction sytems can be considered tissue-like P systems where in the course of computation each cell contains only at most one occurrence of any type of objects. (Such P sytems were defined by A. Alhazov in 2006). We introduced and examined two variants of networks of reaction systems, called communicating reaction systems with direct communication where the reaction systems send either products (objects) or reactions (rules) to each other. We proved that these types of networks of reaction systems can be obtained by simple mappings (projections) from single reaction systems and discussed some aspects of communication within these networks [Publications, item 38]. We plan to examine bio-inspired properties of communicating reaction systems and their counterparts in P systems (P colonies), we have initiated research on steady states and mass conservation in these systems [Publications, item 37].

We also studied the reversibility of communicating reaction systems, in case of communication reaction products to specific target components. We first considered the possibility of “backtracking” their computations, then defined the causal consistent counterpart of these systems which implement controlled reversibility in a causally consistent manner [Publications, item 40]. The article has already been published, it is available online from June 2, 2022, although the official publication date of the volume containing it is August 18, 2022.

We assigned languages to networks of reaction systems in [Paper, 1] and provided representations of well-known language classes by these networks.

A paper on string assembling systems and Watson-Crick finite automata was published [Publications, item 39]. Watson-Crick finite automata can be related to P colony automata.

4. Applicability of the results

Our research is basic research, so our results cannot be directly applied in practice. However, the application of the obtained results and new ideas as models, tools, or new approaches can be expected in other scientific fields (for, example Petri nets) or in classical computing. But, direct application of the theory is also possible. In paper [Publications, item 28] the well-known clustering algorithms DBSCAN was implemented in terms of P systems and the new algorithm lead to significant speed up.

5. Further achievements

Papers:

[Paper, 1] L. Ciencialová, L. Cienciala, E. Csuhaj-Varjú, Languages of Distributed Reaction Systems. In: J. Durand-Lose, Gy. Vaszil (eds.), MCU 2022, LNCS 13419, Springer, to appear.

[Paper, 2] L. Cienciala, L. Ciencialová, E. Csuhaj-Varjú, About Reversibility in sP Colonies and Reaction Systems. Natural Computing, under review.

The project supported the research included in the PhD thesis of junior participant Kristóf Kántor.

The PhD dissertation of junior participant Gábor Kolonits is almost finished, his research was supported by the project.

Attila Bagossy joined the project as a junior participant with a five-month investigator contract. In this period, he took part as co-author in the research reported in the paper [Publications, item 40]. He also prepared an online simulator for different models of computation, including reaction systems and P systems. The simulator can be found at

https://github.com/battila7/computation-sandbox