Bodies, Interactions, and Hypercomputation[1]

Jozef Kelemen

Institute of Computer Science, Silesian University

Bezruč Square 13, 746 01 Opava, Czech Republic

and Gratex International, a. s., Bratislava, Slovakia

kelemen@fpf.slu.cz

Abstract: After sketching the role of embodiment in present models of intelligent computer-based systmes, and making some notes concerning the computational power of the emboidied systemst, the contribution proposes EG-systems as a suitable formal framework for study some of the relevant properties of embodied autonomous agents acting in computationally complicated environments, and provides a result concerning the super-recurcsivness of a variant of EG systems in this context.

Keywords: Computability, autonomous agents, embodiment, interactions, eco-grammar systems,super-recursivness, hypercomputation.

1 Introduction

Brownstone: [*In a more frenzied tone of voice*.] It's difficult for me to tell you the exact nature of our problem with Max. I've been working with computer systems as a professional for almost thirty years, but nothing like this has ever happened before.

Worthmore: Relax, Harry. Take a deep breath. [*Brownstone sits back and breathes deeply.*]

Worthmore: Now tell me exactly what I need to know.

Brownstone: It seems that Max – Max – [*with great resolve*] Max fell in love with a beautiful co-ed, and he is suffering because he cannot consummate that relationship.

---

[1] The author’s research on the subject is supported by the Grant Agency of the Czech Republic, grant No. 201/04/0528.

Worthmore: Do that again.

Brownstone: Max is completely and totally obsessed with one of our co-eds. Yet, he cannot embrace her because he does not have – he does not have arms. He does not have a body. [*Pause*] Max wants a body. That's what it all boils down to.

The previous conversation between Mr. Browstone and Mr. Worthmore is taken from Richard Epstein‘s theatre play *Mad Max – Beyond Turing Drone*, from the end of the past Century[2]. It documentates not only some of possible (?) professional difficulties with the perfect use of robots of the future but also the increasing public interest on problems of this kind, as well. Moreover, it documentates also our todays embarrassment about about how to continue with science. *We have become very good at modeling fluids, materials, planetary dynamics, nuclear explosions and all manner of physical systems. Put some parameters into the program, let it crank, and out come accurate predictions of the physical character of modeled system. But we are not good at modeling living systems, at small or large scales. Something is wrong. What is wrong? There are a number of possibilities: (1) we might just be getting a few parameters wrong; (2) we might be building models that are below some complexity threshold; (3) perhaps it is still a lack of computing power; and (4) we might be missing something fundamental and currently unimaginable in our models* wrote Rodney A. Brooks (2001, p. 401), a top-specialist in the field of robotics and artificial intelligence.

The situation depicted by Brooks is similar to that appearing in many other fields, e.g. with studies of cognition, intelligence, perception, etc. We use the standard traditional conceptual framework for studying computing devices and their behaviors interpreted as computation to dissolve the miracle of the mentioned phenomena. Having at hand the prepared set of notions and scientific rules which express possible relations between them we try to explain the nature of these phenomena. We are in certain extent successful in doing that. But fundamental difficulties remain open even with formulation of some questions concerning these miraculous phenomena, but there have high actuality for better understanding of the just arising completely new meaning of the concept of machines.[3]

---

[2] R. G. Epstein is a professsor of computer science at the Westchester university of Pannsylvania, Westchester, PA, and a playwriter. His mentioned play was performed first after the conference banquet oorganized during the *Future of the Turing test Conference* at Darthmout College, Hanover, NH, January 28.-30, 2000.

[3] For more details on the cultural and scientific evolution of the concept of the machine during the 20th century see (Horáková, Kelemen, 2003).

However, the concept of machine is not the only one which has been changed considerably during the second half of the 20th century. Moreover, important changes have been started already at the end of the 19th century in viewing the relation of human psyche and body thanks to the pioneering work of Sigmund Freud. In his unpublished during his life paper written in 1895 (Freud, 1954) he sketched the picture like this one in the Fig. 1.

Wee can clearly recognize Freud's division of the psychic and the somatic (in the picture the horizontal *somat.-psych. Grenze*) as well as the humans "I" from the outer environment (in the picture the vertical *Ich-granze*) in his representation of the sexual function; cf. e.g. (Panhuysen, 1998). The continuous interaction of the human body with its environment is the base for complicated processes the psychic processes inclusive, which results in a specific state of the human mind and in performing some sexual behavior. Without the body as certain kind of "interface" the state of mind nor the related with it behavior does not emerge in human beings. From this fact, among other, follows the importance of the human body for human mind and behavior.

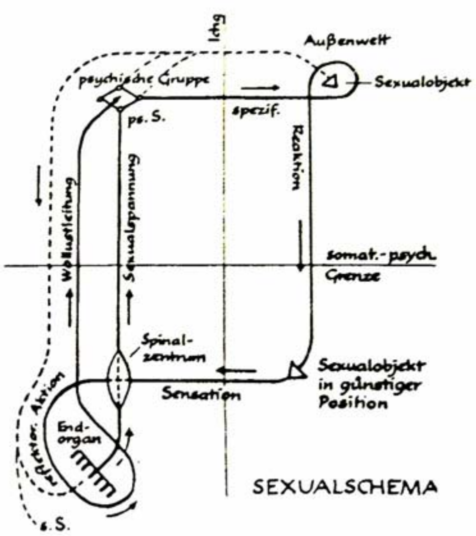

Fig. 1: Freud's representation of the sexual function; from (Penhuysen, 1998).

Situation is very similar in nowadays research in many disciplines focused to human beings, to machines, or to the intersection of the both categories in some branches of science, cf. e.g. (Humphrey, 2000). Some of the specialists, esp. some of those working in the fields of cognitive science, artificial intelligence, and advanced robotics, argue that the source of problems with discovering more adequate and effective ways how construct (esp. how to program) machines in order to provide their continuous functioning in dynamically changing environments consists in the embodiment of systems, the phenomenon which remained almost completely ignored in our recent computationalistic models. The traditional so called *mind-body problem* of philosophers and cognitive scientists, and the actual *software-hardware problem* of computer programmers and robot builders are form the perspective of embodiment in certain sense identical.

The core of the problem consists, according (Scheutz, 2002), in the simple fact that since symbols are abstract entities, computations cannot be performed on them, but have to be mediated through something physical (like organic bodies of living beings or inorganic machines) that can be manipulated by some physical operations that correspond systematically to the ones performed during the abstract computational processes over abstract symbols. Moreover, because of the same reasons the symbols themselves must be represented in certain ways by suitable physical entities. These entities are then manipulated by the above mentioned physical processes and the results of manipulation are reinterpreted as the results of an abstract computation.

The just described *symbolic--physical dichotomy* remains up to now in our theories unmentioned at all. But when we concentrate to build embodied systems acting in a dynamic, often unpredictable environments, we are confronted with the question how the abstract and the physical is related and how this relation influences the behavior of our robots, for instance. This is the core of the *problem of embodiment* (at least for the purposes of this paper) – the problem which is highly actual e.g. because of effective construction of different physically embodied autonomous agents. Unfortunately, we have no effective tools at hand to study such systems with required theoretical rigor up to now.

The aim of this contribution is to sketch a computationally relevant sub-problem of the general problem of embodiment. The question is whether we are able to construct certain formal theories which reflect some of the formally expressible properties of massive interactions between the agents and their environments, and mutually between the agents, and to sketch a way how to deal with this sub-problem in the theoretical framework of the so called eco-grammar (or EG, for short) system.

2 A Very Short Story of Artificial Intelligence

An important achievement of Artificial Intelligence (AI) was the discovery of the methodologically new possibility how to test our hypotheses on how (some of) the intellectual processes run. The history of AI is full of different hypotheses on how to “automatize” processes like problem solving, theorem proving, natural language understanding and communication, diagnostics, image processing and recognition, scene analysis, etc. in order to obtain working computer-based systems performing these tasks at the similar (or at the better) qualitative level as (specially trained) human beings perform them. In all these cases:

1. (1) A working hypothesis is produced first – in the majority of the cases it is based on author's own introspection, then
2. (2) the formulated hypothesis is implemented (often using a suitable programming language that might be developed for such purposes), and
3. (3) the developed system of programs (the implemented version of the hypothesis) is then tested on real (or more or less similar to the real ones) data.

To proceed according such methodological guidance seems to us as something natural. It might be because intellectually we feel prepared for contemplations about our own intellectual capacities. Perhaps the most deeply developed system of this kind is the well-known system GPS (General Problem Solver) by A. Newell, H. A. Simon and their collaborators (Newell, Simon, 1972), (Ernst, Newell, 1969) continued in the frame of the project SOAR (Newell, 1990).

The most illustrative achievements of the use of the above-sketched methodology are the knowledge systems (Stefik, 1995) having symbolically represented ontologies of notions, their chunks, taxonomies, relations between them, etc. As the consequence of that, knowledge systems do not need any bodies (in the physical sense). The situation is completely different in the cases when the artificially created systems (intended to be intelligent in certain sense, e.g. cognitive robots) are situated and execute tasks in real physical environments. In such a case the systems are faced with physically grounded ontologies of objects with real physical properties that exist and act in real time scales. Very hard problems appearing in such situations in the traditional good old fashioned AI were pointed out firs from very different positions and with very different conclusions by M. Minsky (1986) and R. Brooks (1999).

Brooks (1999) in his concept of the new AI emphasizes the principal role of systems reactivity, which is necessary for their low-level rationality, while Minsky (1986) emphasizes the principle of decentralization and organization of simplest units (agents) into more complex ones (agencies) and presupposes that an agency may play the role of an agent in a more complex agency. Both of these positions might be – according to our conviction – combined into one unified approach. The main idea consists in two basic steps:

1. (1) in emphasizing the role of as direct as possible interaction of the cognitive systems with their environments at least at the lowest level of sensing and acting, and
2. (2) in exploiting the power of organization and of the emergence in highest levels in order to receive more complex behaviors.

Both of the above mentioned steps lead us to realize the principal difference between *implementation* of our ideas on how cognitive processes run in natural systems and how they may run in artificial ones, into more or less traditional but in certain sense rigid computers usually equipped with suitable input-output devices which isolate them form their environments by providing data from it for them, and between *embodiment* of our ideas into artificially created systems equipped by sensors providing signals for them, by units for processing signals and perhaps compute the decisions, and by actuators for making changes in their environments, and situated and working continuously in real, dynamic, and noisy environments; for more details see e.g. (Ačová et al., 2004).

The bodies of our more or less smart machines became the principal problem of our scientific consideration. We have very deep experience with understanding physical machines as physical systems, e.g. in mechanics. However, as we have mentioned above, the mechanistic view of bodies is not sufficient when we are interested in behavioral aspects of functioning of machines. In computer science we are interested rather in virtual machines, in machines in the case of which we make a shift in abstracting behavioral aspects of these machines (the software) and exclude from any considerations their bodies (the hardware). This type of separation has been and still is fruitful in certain situation appearing when the computers are used in traditional ways, but is not sufficient in some other cases.

3 A Very Short Story of Traditional Computing

According the traditional understanding of computation we can recognize any computing device as an *externally passive entity* which internal activity is based strictly on activities of a finite number of externally passive components with predefined message passing and transformation possibilities of this entity. Thank to the internal activities of components and their addressed communication the whole system transforms the inputs provided to it from certain environment into required outputs. This activity – if it satisfies a dozen of previously well-specified requirements – is interpreted as a computation in the traditional sense developed during the modern history of computing which started in 30ties of the 20. century with definition and first studies of (abstract devices equivalent with) the Turing machine.

The Turing machine working in an environment gets its input in advance at a beginning of its work, and outputs the result to the environment at the end of its activity. During the computation, the environment is – from the perspective of the Turing machine – completely passive. Computing and computation are understood, applying this traditional paradigmatic view to the systems understood as computing devices, as specific processes corresponding to mathematically defined functions. While the function *declares* a specific relation between variables and values in a set theoretic sense (to definition of a function coincides with a defining a suitable subset of the Cartesian product of its domain of variables and domain of its values), the traditional view of a computation (of a function) is *procedural* one: a computation define a function by means of specifying a step-by-step process of elementary computable transformation steps which transform the given input variable to a corresponding output value (of the corresponding function).

The central problems of (theoretical) computer science originated from the point of view of the just described traditional paradigm of computing are related with the possibility, description, execution, and the effectiveness of an idealized rule governed *algorithmic* transformation of input data into the desired outputs.

Inside the above sketched overall picture of the traditional understanding of computation, the property of computability – or in other words the (partial) recursiveness (of mathematically defined functions) – is derived from the computing power of the Turing machine. This is the core idea of the so-called Church-Turing thesis, which, in a more precise formulation, states Turing machines, logics, lambda calculus, algorithmic computing, and the generative capacity of centralized rule-based systems (Chomsky-type formal grammars) as equivalent mechanisms for solving problems; cf. (Wegner, Goldin, 2003).

4 With Interactions Beyond Turing Machines?

However, in present there are strong efforts to prove that the notion of computation might be enlarged beyond the traditional boundaries defined by Turing computability[4]. In (Burgin, Klinger, 2004) it is proposed to call algorithms and automata that are more powerful than Turing machines as *super-recursive*, and computations that cannot be realized or simulated by Turing machines as *hyper-computations*. In our following consideration on the possible views of computation we will respect this proposal.

Another possibility of viewing systems as computing devices consists in considering a computing device as an *externally active entity* perceiving its dynamic (might be hardly predictable, noisy, or completely unpredictable) outer environment, and acting in it continuously according the perceived stimuli and the own inner rules governing the behavior of the system in order to complete given tasks. This is the core idea of the third period of the history of modern computing when the more or less freely cooperating and communicating interacting processors individual behaviors result in a behavior interpretable as a solution of a given problem. The interactivity, as stated in (Copeland, 2004) in connection with the analysis of the computational power of the Turing machine *coupled* with its environment, or with the same device appearing as the *interactive* Turing machine

---

[4] For more details on the effort see e.g. (Eberbach, Wegner, 2003) or the monothematic issue of *the Theoretical Computer Science* **317**, No. 1-3 (2004) 1-269.

in (Wegner, 1997) leads to the hyper-computational power of the interacting in the Turing sense computationally universal devices.

The activity of the above mentioned type of systems is based on their own coupling of sensed data with appropriate acts performed in their environment, or on the activities of individually autonomous components forming these systems, and communicating (directly or indirectly) with other components forming them. Systems of this type are usually called *agents*, and the structures formed by these agents are called *multi-agent systems*; cf. e.g. (Ferber, 1999). In (Kelemen, 2003) we called the emerging new paradigm of considering computing systems are emerging from considering such kind of autonomous "open" systems as computing devices instead of the isolated ones as the *agent paradigm* of computing.

Interactions of agents with other agents and with their (dynamically changing, unpredictable, noisy, etc.) external environment during their activities in it are a real promise how to enlarge computational power of systems; cf. e.g. (Wagner, 1997). In general, interactions inside a multi-agent system involve the external word and the activities of individual agents into the behavior (interpreted as a computation) of the whole system *during* the computation (rather than *before* and *after*, as it is in the case of the traditional algorithms) which may lead to the computations that cannot be carried out by a Turing machine, as stated in (Eberbach, Wegner, 2003). So, agents and multi-agent systems might be considered as very powerful computational devices and may contribute with many innovative concepts to our traditional picture of the (theoretical) computer science and engineering.

An important dimension of the agent paradigm consists in considering agents not only as products of the development of computer programming techniques and as innovative tools for computer use, but also as products of development of electro-, mechanical-, and computer-engineering, as electro-mechanical (usually computer guided) devices for automation of different physical processes – as real autonomous machines which do physical (mechanical) work. From such a point of view, as we have mentioned already, there exists an important difference between real computers and the abstract Turing machine.

In (Sloman, 2002) is stated, for instance, that computers, as built and used, are the result o a convergence of the development of the machine- and electro-engineering, and the progress in understanding computations as processes of performing actions on symbols as the Turing machine do that. In our terminology, real computers as well as the real agents – (artificially) intelligent systems, esp. the cognitive robots – are entities which cannot be divided into their hardware and software parts without missing something fundamental (might be something which emerges) form the functioning of both of that their parts. According (Sloman, 2002), this difference makes computers useful, but Turing machines irrelevant for research in Artificial Intelligence, for instance. These two dimensions of agents – interaction with dynamically changing environment and embodiment – converge into a new understanding of machines as embodied, autonomously sensing, acting and deliberating agents – into the form of *robots*. Looking to the Figure 2 we easily realize the substantial difference between the abstract universal computing devices equivalent with the Turing machine with respect of their computational power in over-simplified environment of symbols written on a tape, and the real robots equipped with computers programmed in order to control the behavior of these robots in the real dynamically changing physical environments.

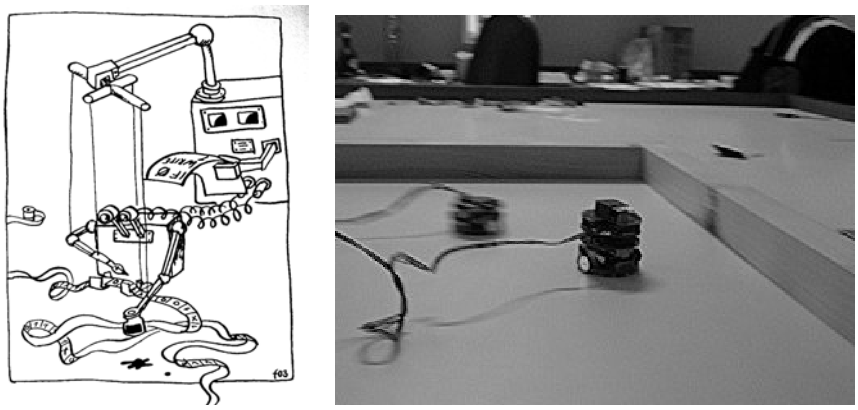

Fig. 2: A (caricature of a) Turing machine in its ennvironment, taken from (Markoš, Kelemen, 2004), and two real Khepera robots in their environment in the Robotic Lab of the Silesian University Institute of Computer Science at Opava.

The mentioned above difference, the properties like the autonomy and continuity of machines behavior, the relevance of embodiment, and other physical constrains and limitations (esp. the problematic concept of infinity with respect their behavior), the importance of communication between individually independent, autonomous computational units in order to achieve common goals (intentionally or as an emergent effect of their co-existence in a shared environment), etc. seems to be crucial for embodied systems like robots; cf. e.g. (Parker, 2003).

Many computational processes in robots processors run continuously and autonomously in different types of environments. Good examples are computing processes running in autonomous mobile robots. When – for instance – a collision avoidance module is programmed, its role is to process the input sensory data continuously during the robot mission into the data manipulating with robots actuators in order to avoid obstacles in robots environments. Of course, all the programs of a robot may be decomposed into the set of interrelated programs of traditional type. However, this type of reduction does not contribute to the solution of the problem of collision avoidance at all! Instead of particular programs considered as translation of mathematical functions into some more procedural languages we must think in terms of autonomy and continuity of functioning of systems modules based on their ability to sense the environment and act in it, and on their massive interactions.

In order to apply this new experience in modeling complicated systems (e.g. in economics, sociology, biology, robotics, etc.), the following methodological experience seems to be important: Instead of the necessity to aggregate specific particular data on individual objects as the base for (mathematical) modeling, the agent paradigm provides a tool for model each individual behavior separately and then study the emerging behavior of the society of these individuals. This methodology is present in many nowadays experiments with biological, ecological, economic, electro-mechanical (robotic) or conceptual societies of agents.

In the following part of this contribution, especially the interactions of modules (individual agents) inside systems will be in the centre of our attention. We will sketch the influence of communications of collections of individually autonomous agents with traditional computin gpower to the computing power of the wholek szstem set up from these agents, we will consider the activity of the whole szstem as a *computation*, and we will show that the power of such computation mzy under certain circumstances owercome the border of traditional Turing-computability.

5 A Grammar-Theoretic Framework for Hypercomputation

From computational point of view an appropriate sub-problem of the above described problem of embodiment is that one which consists in rigorous specification of the computational character of results of interaction of the rule governed algorithmic symbol-manipulating processes which run inside of the agents which interact massively with their computationally complicated behaving dynamic environments. Usually we are interested in as precise as possible knowledge of the behavior of an agent or a multi-agent system in its environment despite of the fact that we have no complete knowledge of the behavior of the environment. The solution of this problem is twofold: We may study the *possibility* of performing such and such behavior under such and such conditions put to the behavior of the environment. To solve this type of problem is in certain extent the traditional role of theoretical computer science. Another possibility is to concentrate to *feasibility* (of course, it will be necessary to define rigorously what we mean by feasibility in our considerations).

For instance, according (Csuhaj-Varjú et al., 1997), an *eco-grammar* (or an *EG*) *system* Σ consists of an alphabet V, a fixed number (say n) of agents, and evolving according set of rules $P_1, P_2, ..., P_n$ applied in a parallel way as it is usual in L systems (Rozenberg, Salomaa, 1980). The rules of agents depend, in general, on the state of the environment. The agents act in commonly shared environment (the states of the environment is described by strings of symbols w_E , the initial one by w_0) by sets of sequential rewriting rules $R_1, R_2, ..., R_n$. The environment itself evolves according a set P_E of rewriting rules applied in parallel as in L systems.[5] The model is schematically depicted in the figure Fig. 3.

The evolution rules of the environment are independent on agents' states and of the state of the environment itself. The agents' actions have priority over the evolution rules of the environment. In a given time unit, exactly those symbols of the environment that are not affected by the action of any agent are rewritten.

In the EG systems we assume the existence of the so called *universal clock* that marks time units, the same for all agents and for the environment, and according to which the evolution of the agents and of the environment is considered.

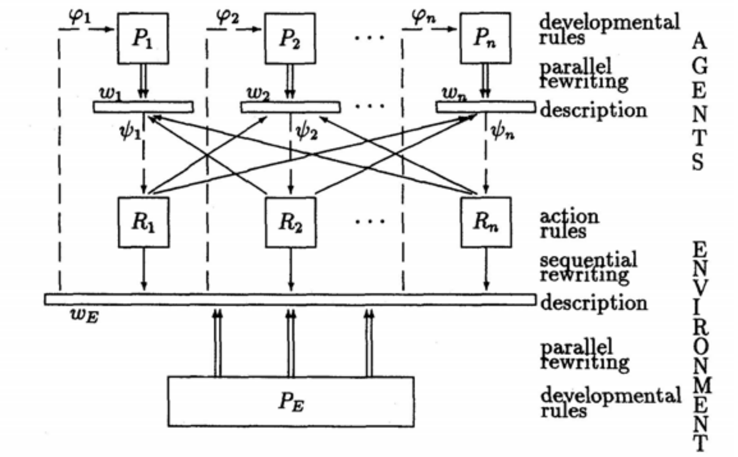

Fig. 3: A schematic view of a traditional EG systém; form (Csuhaj-Varjú et al., 1997).

In (Csuhaj-Varjú, Kelemenová, 1998) a special variant of EG systems have been proposed in which the agents are grouped into subsets of the set of all agents –

---

[5] .So, the triplet (V, P_E, w_E) is (and works as) an L-system.

into the so-called *teams* – with fixed number of members, and the generative power of such type of EG systems have been studied.

In (Wätjen, 2003), where – similarly as in (Csuhaj-Varjú, Kelemenová, 1998) – a variant of EG systems without internal states of agnets is studied, the teams with fixed number of members proposed in (Csuhaj-Varjú, Kelemenová, 1998) are replaced by dynamically changing number of agents in teams. As the mechanism of reconfiguration, a function, say f, is defined on the set N of integers with values in the set $\{0, 1, 2, \dots n\}$ (where n is the number of agents n the corresponding EG system) in order to define the number of agents in teams: For the i-th step of the work of the given EG system, the function f relates a number $f(i) \in \{0, 1, 2, \dots n\}$. The subset of the set of all agents of thus EG system of the cardinality $f(i)$ is then selected for executing the next derivation step of the EG system working with Wätjen-type teams. Wätjen (2003) proved, roughly speaking, that there exist EG systems such that if f is (in the traditional sense) non-recursive function, then the corresponding EG system generates a non-recursive language.

The language is defined in the case of Wätjen's type EG systmes, say Σ, using in each derivation step onlz agents formn the corresponding subset of the cardinality $f(i)$ of the set of agents of Σ, so:

$$L(\Sigma, f) = \{v \mid w_0 \Rightarrow^{f(1)} w_1 \Rightarrow^{f(2)} \dots w_r \Rightarrow^{f(r)} = v,\ r \in N,\ w_0 \dots w_r \ \in V^*\}$$

The proof is in (Wätjen, 2003) given by contradiction. A recirsive language is generated by a special EG system using arbitrary computabel function f. Wätjen uses the EG system

$$\Sigma = (V, P_E, R_1, R_2, \dots, R_n, w_E)$$

where

$V = \{a,\ b,\ b_1, b_2, \dots, b_n\}$,

$P_E = \{a \to a^2, b \to b^2\} \cup \{b_i \to b^2 \mid i = 1, 2, \dots, n\}$,

$R_i = \{b \to bb_i\},\ 1 \leq i \leq n$,

$w_0 = a^2 b^{2n+3m},\ m \in N.$

which generates the following language:

$$L(\Sigma, f) = \left\{a^2 b^{2n+3m}\right\} \cup$$

$$\bigcup_{k \in N} \left(\bigcup_{\substack{1 \leq i_1, \dots, i_{f(k)} \leq n \\ i_j \neq i_{j'}, j \neq j', 1 \leq j, j' \leq f(k))}} \left\{a^{2^{k+1}}\right\} perm \left(bb_{i_1}, \dots, bb_{i_{f(k)}}, \underbrace{b^2, \dots\dots\dots\dots, b^2}_{2^{k-1}(2n+3m)-f(k)\ \ times} \right) \right)$$

This language is recursive, if the function f is recursive. Then the Wätjen's proof is based on demonstratin of a contradiction:

For the non-recursive *f* it is supposed that the that the language *L(Σ, f)* remains recursive. This leadns to the contradiction in the following way: If *L(Σ, f)* is a recursive language, then the words of it can be effectively listed in some order. Now, chose an arbitrary $k \in N$. Then there exists an word w_k in *L(Σ, f)* which is listed after finite number of steps, and we can compute the value *f(k)* for it. So, it follows that *f* is computable, and this is a contradiction. Because of that the language *L(Σ, f)* is non-recursive, and the corressponding EG system generating it is a super-recursive generative device.

6 Some Ideas How to Proceed with EG Systems

Thank to Wätjen we have proved that we are able to imagine EG systems in which subsequently changing groups of active agents interact with the dynamic environment such that the result of interaction result in the non-recursive behavior of this EG system. So, we have a grammatical model of non-recursive behavior based, similarly as in the case of the interacting Turing machines, on the interactions. This proves the hyper-computational power of interactions. The requirement of strict isolation of teams from each other, as well as the style of changes of teams during the run of the derivation process we can – at least metaphorically – interpret as the requirement of some kind of embodiment (or "embodiment"). So, a system of some simple (finite) derivative units – agents – created in some complicated (non-computable) way which interact with a specific shared dynamic environment provide the hyper-computational power of the behavior of the whole system set up from the agents and their environment.

The range of question (with certain relevance to Artificial Life or to the computational studies of evolution, for instance) we can ask about an EG system contains e.g. the following items:

- Is the evolution of a system bounded in time or not? In other words: Enters the system a deadlock? When, under what conditions?
- In the case of the infinite evolution, are non-cyclic evolution chains possible?
- What is the effect of „small changes" either in the initial configuration or in the evolution/action rules of agents and of the evolution rules of the environment?
- What is the effect of introducing further life-like features (the Wätjen-type teams are naturally interpretable in the frame of Artificial Life) into the model to the answers to the previous questions?

The role of the body – at least in the case when we discuss it from computational point of view – consists first of all in generating the behavior of the embodied system with respect the situation appearing in its environment. In order to study the conditions and the power of the generation of behaviors of this type formally we require at least the following:

1. to have an opportunity to study the system with respect of its constituent parts and their interactions, and
2. to have an opportunity to study the interaction of the whole system or its parts with their environments.

In such a situation, we may ask the following questions, for instance:

- What is the computational power of the EG system working on the environment generated by a super-recursive device?
- Are EG systems “regulate“ the super-computable behavior of their environment into the form of a computable one? Under what kind of circumstances?

Let we suppose now that we are able construct on the base of our knowledge in theoretical computer science and with respect the Turing hypothesis only devices with behaviors computable in the sense of the traditional Turing-computability. Suppose that a good theoretical framework for describing this type of devices is the framework of some variant of EG systems. Suppose that the behavior of the environment of EG systems might be very unconventional, exotic ones. Suppose it might be un-computable in the sense of the traditional Turing-computability; for other alternatives see e.g. (Eberbach, Wegner, 2003). Why not? Technically it means that instead of an L system supposed as the generative base for autonomous changes of the environment in a given EG system we suppose that the environment may change in some non-computable in the sense of Turing computability manners. What can we say concerning the behavior of such type of EG systems? More technically: What type of the behavior of an EG system we may expect when this system works in a non-computably changing environment?

Concerning the study of feasibility, we may be interested not only in the exact knowledge of the behavior of an EG system, but also in the (importance of) difference between the behavior of an EG system and the required behavior of it.

References

Ačová, M., Kelemen, J., Kubík, A.: Embodied hypotheses – preliminary notes and case studies. In: *Proceedings of the RoMoCo* (K. Kozlowski, Ed.) Poznan University of Technology, Poznan, 2004, pp. 35-40

Brooks, R. A.: *Cambrian Intelligence*. The MIT Press, Cambridge, Mass., 1999

Brooks, R. A.: The relationship between matter and life. *Nature* **406** (2001) pp. 409-4

Burgin, M., Klinger, A.: Preface – three aspects of super-recursive algorithms and hyper-computation or finding black swans. *Theoretical Computer Science* **317** (2004) 1-11

Copeland, B. J.: Hypercomputation – some philosophical issues. *Theoretical Computer Science* **317** (2004) 251-267

Csuhaj-Varjú, E., Kelemen, J., Kelemenová, A., Paun, Gh.: Eco-grammar systems – a grammatical framework for lifelike interactions. *Artificial Life* **3** (1997) 1-28

Csuhaj-Varjú, E., Kelemenová, A.: Team behaviour in eco-grammar systems. *Theoretical Computer Science* **209** (1998) 213-224

Eberbach, E., Wegner, P.: Beyond Turing machines. *Bulletin of the EATCS* **81** (2003) 279-304

Ernst, G., Newell, A.: *GPS – A Case Study in Generality and Problem Solving.* Academic Press, New York, 1969

Ferber, J.: *Multi-Agent Systems*. Addison-Wesley, Harlow, 1999

Freud, S.: Project for a scientific psychology. In: *The Origins of Psycho-Analysis* (M. Bonaparte, A. Freund, and E. Kris, eds.). Basic Books, New York, 1954

Horáková, J., Kelemen, J.: Čapek, Turing, von Neumann, and the 20th century evolution of the concept of machine. In: *Proc. Intern. Conf. in Memoriam John von Neumann*. Budapest Polytechnic, Budapest, 2003, pp. 121-135

Humphrey, N.: *How to Solve the Mind-Body Problem*. Imprint Academic, Thorverton, 2000

Kelemen, J.: The agent paradigm – foreword. *Computing and Informatics* **22** (2003) 513-519

Markoš, A., Kelemen, J.: *Berušky, andělé a stroje* (Ladybugs, Angels, and Machines; in Czech). Dokořán, Prague, 2004

Minsky, M.: *The Society of Mind*. Simon and Schuster, New York, 1986

Newell, A.: *Unified Theories of Cognition*. Harvard University Press, Cambridge, Mass., 1990

Newell, A., Simon, H. A.: *Human Problem Solving.* Prentice Hall, Englewood Cliffs, N. J., 1972

Parker, L.: Current research in multirobot systems. *Artificial Life and Robotics* **7** (2003) 1-5

Panhuysen, G.: The relationship between somatic and psychic processes – lessons from Freud's *Project*. In: *Neuroscience of the Mind on the Centennial of Freud's Project for a Scientific Psychology* (R. M. Bilder, and F. F. LeFever, eds.). NYAS, New York, 1998, pp. 20-42

Rozenberg, G, Salomaa, A.: *The Mathematical Theory of L-systems*. Academic Press, New York, 1980

Scheutz, M.: Computationalism – the next generation. In: *Computationalism* (M. Scheutz, Ed.). The MIT Press, Cambridge, Mass., 2002, pp. 1-21

Sloman, A.: The irrelevance of Turing machines to artificial intelligence. In: *Computationalism* (M. Scheutz, Ed.). The MIT Press, Cambridge, Mass., 2002, pp. 87-127

Stefik, M.: *Introduction to Knowledge Systems*. Morgan Kaufmann Publ., San Francisco, Cal., 1995

Tecuci, G.: *Building Intelligent Agents*. Academic Press, San Diego, Cal., 1998

Wätjen, D.: Function-dependent teams in eco-grammar systems. *Theoretical Computer Science* **306** (2003) 39-53

Wegner, P.: Why interaction is more powerful than algorithms. *Communications of the ACM* **40** (1997) No. 5, 81-91

Wegner, P., Goldin, D.: Computing beyond Turing machines. *Communications of the ACM* **46** (2003) No. 4, 100-102