The Emergent Computational Potential of Evolving Artificial Living Systems*

Jiří Wiedermann[1] Jan van Leeuwen[2]

[1]Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
e-mail: jiri.wiedermann@cs.cas.cz

[2]Institute of Information and Computing Sciences, Utrecht University
Padualaan 14, 3584 CH Utrecht, the Netherlands
e-mail: jan@cs.uu.nl

Abstract

The computational potential of artificial living systems can be studied without knowing the algorithms that govern their behavior. Modeling single organisms by means of so-called cognitive transducers, we will estimate the computational power of AL systems by viewing them as conglomerates of such organisms. We describe a scenario in which an artificial living (AL) system is involved in a potentially infinite, unpredictable interaction with an active or passive environment, to which it can react by learning and adjusting its behaviour. By making use of sequences of cognitive transducers one can also model the evolution of AL systems caused by 'architectural' changes. Among the examples are 'communities of agents', i.e. by communities of mobile, interactive cognitive transducers. Most AL systems show the emergence of a computational power that is not present at the level of the individual organisms. Indeed, in all but trivial cases the resulting systems possess a *super-Turing computing power*. This means that the systems cannot be simulated by traditional computational models like Turing machines and may in principle solve non-computable tasks. The results are derived using non-uniform complexity theory.

"What we can do is understand some of the general principles of how living things work, and why they exist at all."
From: R. Dawkins, *The Blind Watchmaker*, 1986.

1 Introduction

A tantalizing question in *computational mind modeling* is the following: if one accepts that the mind can be modeled by computational means, how can it be explained that mathematicians are often able to prove theorems whose truth or falsity cannot be proved algorithmically within

*This research was partially supported by GA ČR grant No. 201/02/1456 and by EC Contract IST-1999-14186 (Project ALCOM-FT). A preliminary version of this paper appeared as an invited talk in [29]. Version dated: January 10, 2002.

a given formal system (e.g. corresponding to a computer that simulates the mind) due to Gödel's incompleteness theorem. In an extensive discussion of this problem, R. Penrose [10] conjectured that there must be some so far unknown faculty of the brain (sometimes, even in Turing's original work [14], called "intuition") that gives it a non-computable, non-algorithmic, "super-Turing" power in some cases.

A similar, less straightforwardly formalizable question concerns the emergent behavior of societies, or colonies, of living organisms: what is the nature of the (presumably computational) mechanism behind the complex behavior of such societies that emerges, given the often far simpler behavior of the individual organisms? What is the computational potential of the resulting system as an information processing entity?

In this paper we offer a plausible explanation of this phenomenon in the realm of artificial living (AL) systems. We will describe a reasonable computational scenario that shows that the ability to surpass the computational limits of traditional Turing machines can emerge in non-uniformly evolving families or communities of far simpler computational devices, viz. finite transducers. The resulting AL systems will be said to posses a *super-Turing computing power* if and only if they can perform computational tasks that cannot be achieved by classical means, making use of the computational mechanism of standard Turing machines or its equivalents. The computational scenario that we describe originates from recent considerations in non-uniform complexity theory.

The plan of the paper is as follows. First, in Section 2 we introduce the basic tool for modeling a single living organism – an interactive cognitive transducer seen as a finite discrete-state computational device. In Section 3 we model the evolution of such devices by means of potentially infinite sequences of cognitive transducers of increasing size. We show that the resulting "families" possess super-Turing computing power, using basic notions from non-uniform complexity theory. In Section 4 we show that so-called active cognitive transducers, which can move in an interactive environment and modify it at will, gain the computing power equivalent to that of standard (interactive) Turing machines. Finally, in Section 5 we consider AL systems composed of evolving communities of communicating active cognitive transducers. We will show a 'super-Turing computing power' can indeed emerge in such systems.

All the above mentioned results are based on results from non-uniform computational complexity theory (cf. [1]) and some results recently proved by the authors. Our main aim in the present paper is to interpret the results in terms of cognitive and evolutionary systems, so as to shed new light on the computational potential of the respective systems. Results are mostly quoted and not proved here, as it is their interpretation that presents the main contribution of the present paper.

2 Cognitive transducers

When living organisms are modeled in order to study their computing potential, it is important to keep in mind that the computational power of a model can be studied without actually knowing the concrete algorithms that are used by the organisms in concrete situations. We only need to know the set of elementary actions which can be performed in the given model and the scenario of its interaction with its environment, i.e. what data can be input, whether and how this data depends on previous outputs, whether data can be "off-loaded" to the environment, and so on.

Next, one has to take into account that there is a crucial difference between the requirements placed upon a model in case one wants to simulate the behaviour or actions of the modeled (living) organism, and the requirements in case one merely wants to investigate its computing potential. In the former case the choice of a more powerful model than is necessary is acceptable since this can simplify the task of simulation. In the latter case the same choice would lead to an overestimation of the computing potential of the organism. Thus, in the latter case the model must neither be too powerful nor too weak: it must exactly capture the facilities that constitute the essence of the capabilities of the organism to compute.

2.1 Computational scenarios

Fortunately, in the latter case we are in a much better situation than one may think. Despite their unprecedented complexity, when measured in terms of the complexity of human artifacts, it is commonly believed that each living organism can enter into only a finite – albeit in most cases astronomic – number of distinguishable internal configurations. We cannot afford to give exhaustive arguments in favor of this fact here. Instead, we simply postulate for the purposes of this paper that *a living organism interacting with its environment can at each time be modeled by a finite discrete-state machine.* In the sequel we will call any finite discrete-state machine that is used in this context a *cognitive transducer.*

Finite transducers as known from automata theory are the paradigmatic example of cognitive transducers. Other examples of cognitive transducers are discrete neural (cf. [9], [24]) or neuroidal [16] nets, neuromata [11] and various other computational models of the brain (cf. [24], [25]). A more precise definition of a cognitive transducer will be given in definition 1. If the modeled organism is growing and/or evolving in time, while adapting itself to its environment, our postulate remains unaffected. The evolution of the organism, and thus the adaptive mechanism that underlies it, will be captured by making use of sequences of finite discrete-state machines in Section 3.

A suitable model has to capture the fundamental difference between the standard scenario of computations by finite transducers or Turing machines the non-standard scenario of 'computations' by a cognitive transducer. In the former case it is assumed that a finite sequence of inputs is known and given, prior to the start of the computation. No changes are allowed after the computation has begun, not even in the inputs 'further down' that are not yet read[1]. In this classical scenario, if the finite transducer or Turing machine is set to work on a next sequence of (new) input data, it must start again from the same initial configuration as in the previous run. No transfer of information from past runs to future runs takes place. Under this computational scenario, the respective machines are prevented from learning from past experience.

The computational scenario of cognitive transducers is quite different from the classical pattern. It takes after living organisms, interacting with their environment, that process signals (inputs) as these are delivered by their sensory systems without interruption. The inputs simply 'appear', in on-line manner and unexpectedly and *possibly as the answer to earlier responses of the organism.* Moreover, the inputs stream into the organism's cognitive system in parallel via numerous channels and are also processed in a parallel manner. The number of input channels

[1] Traditionally also *on-line* computations are considered, in which input elements are supplied as the computation goes. Even in this case there is normally no feedback or learning mechanism taken into account.

depends on the complexity (or size) of the organism at hand. As a rule, the input signals must be processed in real-time[2]. In most cases, the original inputs are no longer available after they are 'read'. In principle the ongoing 'computation' never terminates and is practically limited only by the lifespan of the organism. Given the ability of (especially complex) organisms to modify their environment or communicate with other organisms, the inputs may depend on their previous actions or the reactions of other organisms. In this way the systems gain a potential ability to 'learn'.

When applied to cognitive transducers, the resulting computational scenario is called *interactive computing*. However, aside form being a scenario for computing, the scenario allows for perpetual interactive *adaptation*, in the following sense. If, to an outside observer, the same 'situation' presents itself in terms of current inputs to an organism, then the organism may react differently from the past, due to the fact that its reactions depend on the whole history of inputs seen thus far. Thus, although a (finite) cognitive transducer may display only a finite number of reactions at any one time, over infinite input streams (the ordering of) its reactions can vary in an infinite number of ways.

2.2 Modeling a single organism

In order to obtain a suitable computational model of a single living organism, we follow the paradigm of automata theory [6]. Under the classical scenario, finite transducers like Mealy automata would be used: they are designed for processing finite sequences of symbols written on finite 'tapes' and, with every transition between states triggered by some input, a fixed output is associated. Under the interactive scenario, we consider *interactive finite transducers* (IFTs), a generalization of Mealy automata, as our basic organism model. IFTs process potentially infinite strings (called streams) of input symbols and produce a potentially infinite stream of output symbols, symbol after symbol. More importantly, we assume that there is no input tape: the transducer obtains ('reads') the inputs as they come in via a single input port. Likewise, we assume that the transducer produces outputs via a single output port. There is no way to return to inputs once they read, except when they are stored internally. An IFT follow a fixed finite, Mealy-type transition function. No adaptive and/or evolutionary abilities are taken into account yet; this follows in Section 3.

We assume throughout that the input and output symbols are taken from the alphabet $\Sigma = \{0, 1, \lambda\}$. The interpretation of a symbol λ appearing at a port is that 'presently, there is neither 0 nor 1 appearing at this port'. Let Σ^ω denote the set of all infinite streams over σ. Any IFT realizes a *transduction* ϕ that transforms streams from Σ^ω into similar output streams. The λ's are *not* suppressed in a transduction. The study of cognitive transducers is basically the study of the transductions that they realize.

Clearly, instead of IFTs one could consider any other finite-state device such as discrete neural (cf. [9]) or neuroidal (cf. [16]) nets, neuromata [11], and so on. The latter devices, which read their input in parallel, must be modified so as to process an infinite input stream in blocks that correspond to the number of input ports that they have. Moreover, in order to transfer information (if any) from the previous run to the current one, we assume that they start the processing of the next block of inputs in the configuration which they reached after processing

[2]This seems to be a necessary condition for the emergence of at least a rudimentary form of *consciousness*, cf. [3], [27].

of the previous block. The various models are called *non-uniform* because they are configured differently, possibly non-computably, depending on the size of the input blocks they process. The following result from [28] shows that these different models have the same computational power.

Theorem 1 . *For transductions $\phi : \Sigma^\omega \to \Sigma^\omega$ the following are equivalent:*

(a) ϕ is realized by an interactive finite transducer.

(b) ϕ is realized by a neuromaton.

(c) ϕ is realized by a discrete neural net.

(d) ϕ is realized by a discrete neuroidal net.

This theorem motivates the following, more precise definition of the class of cognitive transducers that we will use throughout.

Definition 1 . *The class $\mathcal{CT}$ of cognitive transducers is inductively defined as follows:*

(a) interactive finite transducers (IFTs) are in $\mathcal{CT}$, and

(b) any device computationally equivalent to IFTs is in $\mathcal{CT}$.

Theorem 1 expresses that the basic types of cognitive transducers are all equivalent. In complexity theory numerous other models of non-uniform computation are known – such as combinatorial or threshold circuits and many other types of neural nets, especially the biologically motivated ones (cf. [8]). Nonetheless, the computational equivalence of the respective models indicates that *computational cognition* is a rather robust phenomenon that can in principle be realized by a variety of computational models which are equivalent to IFTs.

We say 'in principle' because in practice much will depend upon the efficiency of such models. This might also be the case for *artificial consciousness* (cf. [3] for a recent report on the status quo in this field). For instance, in [27] an algorithmic principle for the emergence of consciousness in artificial cognitive systems is sketched. In the simplest case consciousness takes the role of a control mechanism that, based on feed-back information from the sensors of a system, verifies the correct realization of motoric actions to which orders have been issued. If these actions are not performed in accordance with these orders, the consciousness will realize it and take care of the appropriate remedy. In order to fulfill this role, consciousness must operate in real time w.r.t. the speed of the system. The system must react fast enough to be able to recognize the erroneous realization of its orders and take the appropriate measures in time so as to give the opportunity for realizing rescue actions. In practice such requirements disqualify 'slow' systems and support the specialized, fast or 'economical' solutions. For example, it is known that there are cognitive tasks that can be realized by a single biological neuron over n inputs whereas the equivalent neural nets require a quadratic number of standard neurons [8].

2.3 Learning potential

Cognitive transducers embody two features of computing that are not met under the classical computing scenario: *interactivity*, and *infinity of operation*. The interactivity enables one to describe (albeit *a posteriori*) the interaction between the transducer and its environment: inputs succeeding to some outputs may be reactions to these outputs. The infinity of operation refers to the property that a cognitive transducer is 'always on' and never stops processing inputs.

Although the learning potential of IFTs is not quite obvious, it can be easily observed e.g. in the case of neuroids [16]. Namely, they can be seen as 'programmable neurons' since this ability was their primary design goal. It appears that for understanding the cognitive abilities of organisms, the 'atomic' level of individual neurons or state-transitions of a finite transducer is too low. Thus higher-level models are sought (still equivalent to the elementary model of finite transducers) in which basic cognitive abilities, such as a potential of detecting frequently occurring patterns in a sequence of inputs, form the basic set of operations. Such a basic set of elementary operations of a cognitive transducer is proposed in [16], [25] and [27], in order to obtain the potential for the development of cognitive abilities via learning. Nevertheless, as mentioned above, the exact type of learning algorithm is quite unimportant for determining the computational power of the respective devices at a global level. Of course, due to their simplicity, cognitive transducers themselves do not posses universal computing power.

3 Sequences of cognitive transducers

So far we do not have any means to model the *evolution* of cognitive transducers. In this section we will elaborate on this aspect and introduce a framework in which adaptive behaviour and evolution can be facilitated. The framework will almost naturally achieve the potential of universal computing power.

3.1 Evolution through sequences

In order to support the evolvability property, we consider *sequences of IFTs* as introduced in [21]. This leads to a framework in which many more complicated transductions can be realized and the dependence of the computational efficiency on the size of the underlying devices is revealed. We first give a definition and then proceed to explain the computational scenario of sequences. Let $\mathcal{U}$ be some universe of possible states.

Definition 2 *Let $\mathcal{A} = \{A_1, A_2, \ldots\}$ be a sequence of IFTs over Σ, and let $Q_i \subseteq \mathcal{U}$ be the set of states of A_i. Let $G = \{G_1, G_2, \ldots\}$ be a sequence of nonempty finite sets of $\mathcal{U}$ such that $G_i \subset Q_i$ and $G_i \subseteq G_{i+1}$, for $i \geq 1$. Then $\mathcal{A}$ with G is called a* sequence of IFTs with global states. *We will often omit G from explicit mention.*

For a sequence $\mathcal{A}$, there need not exist an algorithmic way to compute the description of the A_i, given i. Thus, the only way to describe the sequence may be to enumerate all its members. The set $\bigcup_i G_i \subseteq \mathcal{U}$ is called *the set of global states* of $\mathcal{A}$. We always assume sequences of IFTs to have global states. We will often omit G from explicit mention when referring to a sequence.

On an infinite stream of inputs over Σ, a sequence $\mathcal{A}$ computes as follows. At the start, A_1 is the active transducer. It reads input and produces output for a while, until it passes control to A_2. In general, if A_i is the current active transducer, it performs its computation using the *local states* from the set $Q_i - G_i$ (which is non-empty). If an input symbol causes A_i to enter a global state $g \in G_i$, then A_i stops processing and passes control to A_{i+1}. The input stream is re-directed to the input port of A_{i+1}, A_{i+1} starts in state $g \in G_{i+1}$ and it continues processing the in-coming inputs as the new active transducer, starting with the next input symbol.

Thus, in effect the input stream is processed by transducers with increasing index. In a sequence of IFTs with global states, the next transducer can be seen as a 'next generation' transducer. This models the property of evolution. The 'transfer' of control to the next transducer is invoked by the transducer currently processing the input. The next transducer continues from the same state in which the previous transducer stopped, but with a possibly 'richer' set of internal configurations to work with. This mechanism enables the transfer of information from the previous stage, without requiring further detail. Note that in finite time only a finite part of a sequence of IFTs can become active.

Instead of sequences of transducers, one may also consider single transducers that 'evolve', with the transducer acting as A_i if and only if $A_i \in \mathcal{A}$ is the currently active transducer. That is, the transition function of the cognitive at hand is the same as that of A_i as long as A_i is active. Of course, as the evolution proceeds, the condition concerning the global states must still be maintained. The resulting type of transducer may appropriately be called an *evolving interactive finite transducer*. For our purposes, the framework of sequences will prove more fruitful. In the context of organisms, the evolution modeled by a sequence of cognitive transducers corresponds to the evolution of creatures either along an idealized (Darwinian) evolutionary scale, or along the line of their life experiences.

A sequence of IFTs is called *polynomially bounded* if and only if there is a polynomial p such that for every $i \geq 1$, the size of A_i is at most $p(i)$. The classes of transductions realized by sequences of IFTs with global states and polynomially or exponentially bounded size will be denoted as IFT-POLY or IFT-EXP, respectively. We will also consider the classes NA-LOG: the transductions realized by sequences of neuromata [8] of logarithmic size, and NN-POLY: the transductions realized by sequences of standard recurrent, or cyclic, discrete neural nets of polynomial size reading their inputs in parallel.

It is clear that, similar to sequences of IFTs, one can design sequences of any type of devices from the class of cognitive transducers. Given their computational equivalence, one can speak of *sequence of cognitive transducers* without specifying exactly which kind of cognitive transducers is used. To say more about the computational potential of sequences, we have to introduce a little bit of complexity theory.

3.2 Benchmarking by interactive Turing machines

Our main tool for characterizing the computational efficiency of sequences of cognitive transducers are *interactive Turing machines with advice*. We will first describe the model of an interactive Turing machine (ITM). The notion of 'advice' will be added subsequently.

Like cognitive transducers, ITMs are I/O-oriented machines that allow for an infinite,

never ending exchange of data with their environment[3]. As before, the input and output symbols are taken from the alphabet $\Sigma = \{0, 1, \lambda\}$ and we assume that in each step an ITM reads a symbol from its input port and writes a symbol to its output port. We normally require that an ITM reacts to any non-empty input by producing a non-empty output symbol within finite time after receiving the input: the *interactiveness* or *finite delay condition*. The condition ensures that if an input stream has an infinite number of non-empty symbols, then so does the output stream. ITMs differ from cognitive transducers in one important respect: they have the interior structure of a Turing machine and thus are (in principle) *infinite state*.

Definition 3 *A mapping $\phi : \Sigma^\omega \to \Sigma^\omega$ is called the* interactive transduction *computed by an ITM $\mathcal{I}$ if and only if for all $\mathbf{x}, \mathbf{y} \in \Sigma^\omega$, $\phi(\mathbf{x}) = \mathbf{y}$ if and only if $\mathcal{I}$ produces output $\mathbf{y}$ on input $\mathbf{x}$.*

Inspired by Wegner's seminal paper [23], the computational power of interactive Turing machines was studied in [17, 19]. Roughly speaking, the theory leads to a generalization of standard computability theory to the case of infinite computations, recently referred to as a 'theory of super-recursive algorithms and computation' in [2]. The results from [17, 19] indicate that merely adding interactive properties and allowing endless computations does not break the computational barrier of Turing machines. The resulting devices are not computationally more powerful than classical Turing machines because each of their computational steps is still Turing-computable from the current state and input: interactive machines simply compute something *different* from the classical machines — namely infinite transductions. Independent studies of the computational power of interaction, under a slightly different framework, were initiated also by Wegner and several other authors (see e.g. [4]). It appears that a new quality of computations is only brought into the computing behaviour of ITMs by letting *non-predictability* enter into the game (cf. [20]).

The aspect of unpredictability (viz. of program changes) can be captured using the notion of 'advice' as studied in nonuniform computational complexity theory (cf. [7, 1]). Advice functions allow the insertion of 'non-computable' external information into the course of a computation, in this way leading to a non-uniform operation over time. The resulting machines are called *interactive Turing machines with advice* (ITM/As).

Definition 4 *An advice function is a function $f : \mathbf{Z}^+ \to \Sigma^*$. An advice is called $S(n)$-bounded if for all n, the length of $f(n)$ is bounded by $S(n)$.*

A classical TM with advice, operating on an input of size n, is allowed to call for the value of its advice function during the computation only on the argument n. An ITM/A can call its advice at time t only for arguments t_1 with $t_1 \leq t$. To realize such a call an ITM/A is equipped with a separate *advice tape* and a distinguished *advice state*. By writing the value of the argument t_1 on the advice tape and by entering into the advice state at time $t \geq t_1$, the value of $f(t_1)$ is assumed to appear on the advice tape in a single step. By this action, the original contents of the advice tape is completely overwritten. Note that no specific computability assumptions are made for advice functions. As a result, the mechanism of advice is very powerful and can provide both classical TMs with advice and ITM/As with highly non-recursive 'assistance'. An important fact about ITM/As is that they are *provably more powerful* than ITM's without

[3] Turing machines that operate on infinite inputs, so-called $\omega-$Turing machines, have been studied extensively before cf. [12], but the interactive features described here are of more recent origin.

advice, i.e., *ITMs with advice can compute transductions that ITMs without advice cannot.* The result follows from a countability argument, and examples can be constructed by a careful diagonalization proof, see [20].

The computational power of sequences of IFTs, and thus of 'evolving cognitive transducers', is linked the theory of non-uniform interactive computing by means of the following fundamental result from [21].

Theorem 2 *A transduction $\phi : \Sigma^\omega \to \Sigma^\omega$ is realized by a sequence of IFTs if and only if it is realized by an ITM/A.*

3.3 Classifications of cognitive transducers

Using sequences of IFTs as the basic tool for modeling evolving families of interacting organisms, Theorem 2 opens the way to a number of related results that enable a classification of the 'information processing power' of various types of artificial living systems.

In order to state some of the pertinent results here, we will consider advice functions whose values are bounded in length by known (computable) functions of t, especially by polynomially or even logarithmically bounded functions. For the definition of complexity measures for ITM/As, see [19]. Let ITM-$\mathcal{C}$ denote the class of transductions computed by ITMs of $\mathcal{C}$-bounded complexity.

Definition 5 *The class ITM-$\mathcal{C}$/$\mathcal{F}$ consists of the transductions ϕ computed by interactive Turing machines from ITM-$\mathcal{C}$ using an advice function from $\mathcal{F}$.*

Common choices for ITM-$\mathcal{C}$ that we consider here are: ITM-LOGSPACE (deterministic logarithmic space), ITM-PTIME (deterministic polynomial time), and ITM-PSPACE (polynomial space). Common choices for $\mathcal{F}$ are *log* (logarithmically bounded advice functions) and *poly* (polynomially bounded advice functions).

In non-uniform computational complexity theory and in the theory of neurocomputing, relations between the complexity classes of Turing machines with advice and various instances of neural networks are studied (cf. [9]). These results can be extended to the interactive case [21]. To circumvent the different input-output conventions in some cases, we call two complexity classes 'equal' only when the devices corresponding to both classes read their inputs sequentially; otherwise, when the devices in one class read their inputs in parallel, we say that they 'correspond'.

Theorem 3 ([21]) *The following relations hold:*

(a) IFT-POLY *equals* ITM-LOGSPACE/*poly, i.e., the class of transductions computed by sequences of interactive finite transducers of polynomial size equals the class of logarithmically space–bounded transductions of ITM/As with polynomially bounded advice functions.*

(b) NA-LOG *equals* ITM-LOGSPACE/*log, i.e., the class of transductions computed by sequences of neuromata of logarithmic size equals the class of logarithmically space–bounded transductions of ITM/As with logarithmically bounded advice functions.*

(c) NN-POLY *corresponds to* ITM-PSPACE/*poly, i.e., the class of transductions computed by sequences of neural nets of polynomial size corresponds to the class of polynomially space–bounded transductions of ITM/As with polynomially bounded advice functions.*

(d) IFT-EXP *equals* ITM-PSPACE/*exp, i.e. the class of transductions computed by sequences of interactive finite transducers of exponential size equals the class of polynomially space–bounded transductions of ITM/As with exponentially bounded advice functions.*

Theorem 3 illustrates the varying degrees of efficiency of different classes of cognitive transducers. For instance, the family of neural nets of polynomial size has the power of ITM-PSPACE/*poly*, whereas families of finite transducers of the same size only have the power of ITM-LOGSPACE/*poly*. *It also demonstrates the emergence of super-Turing power in the course of evolution within sequences*, due to the fact that general computations of ITM/As do posses such power.

4 From cognitive transducers to cognitive Turing machines

In Section 2 cognitive transducers were introduced to model the signal processing capabilities of single living organisms interacting with their environment. This leaves an aspect of living organisms open, namely their ability to influence and *modify* the environment in which they operate. In this section we extend the model to incorporate it.

Consider a cognitive transducer enhanced by a facility that enables it to move around in its (potentially infinite) living environment and to mark the environment in a way that can later be recognized again by the transducer. The resulting device is called an *active cognitive transducer*. We assume that an active transducer can store and retrieve information in/from its environment, similar to a *cognitive robot system*. The question of what computational abilities are gained in this way, can be answered by looking at the versions of active cognitive transducers encountered in automata theory.

Models of finite transducers with a two-way input tape which can mark cells on their input tape, have been studied for years in automata theory (cf. [22]). The machines are computationally provably more powerful than their non-marking counterparts. When one allows a finite set of marks that can be placed to or removed from a potentially infinite environment, one obtains a model equivalent to the interactive Turing machine (ITM) described earlier. Several further conditions may be imposed on the way the machine interacts with its environment, e.g. to model the bounded delay property that cognitive systems often display in their response behaviour (cf. [18]). By suitably formalizing the concepts involved, the following result is immediate, illustrating once more the usefulness of ITMs as benchmark for the theory.

Theorem 4 *Active cognitive transducers have a computational power equivalent to interactive Turing machines.*

Theorem 4 shows that cognitive transducers indeed achieve a jump in computational power when they are equipped with *sensors* that can scan the environment and with *effectuators* by means of which they can modify their environment. By gaining the ability to 'off-load' and 're-load' data, individual active cognitive transducers (or cognitive robots) thus gain the power of interactive Turing machines. In other words, the original finite-state system turns into a

system that can reach a potentially unbounded number of configurations, by exploiting its environment[4]. It is clear that some types of cognitive transducers are more easily adapted to active form than others, but in principle it leads to the appropriate general model of a single living organism that we aimed for in Section 2. Using sequences as in Section 3, one can incorporate the notion of evolution again. In Section 5 this will be extended to *communities* of transducers that evolve over time.

In his design of the 'automatic machine (a-machine)', now known as the Turing machine, Turing [13] admits that he was motivated by a (human) 'computer', which in his days meant 'a person who calculates'. Such a person calculates with the help of a finite table (a 'program') that is held in the person's head, and further using a (square) paper, a pencil and a rubber. In accordance with Turing's motivation, generations of researchers working in artificial intelligence and philosophers of mind have believed that the Turing machine as a whole corresponds to the model of the human 'computer'. Hodges, Turing's biographer, writes in [5]: *Turing's model is that of a human mind at work.* But this is only true to a certain extent: in Turing's model, merely the machine's finite control corresponds to the mind of the modeled calculating person. Our previous discussion suggests that in modeling a computing person, one has to distinguish among *three* different aspects: the computer's finite control (its 'program'), its 'sensors, effectuators and motoric unit' (that enable it to actively interact with the environment), and the environment itself (the working area or tape). Theorem 4 shows that when in addition to the 'mind' of the computer one also takes the sensory, effectual and motoric capabilities and the interaction with a 'rewritable' and potentially infinite environment into account, a computationally more powerful model results.

5 Communities of active cognitive transducers

The final step in our exposition is the composition of 'artificial living systems' from individual organisms (active cognitive transducers) and to show how a Super-Turing computational potential can arise almost naturally in them. To emphasize the similarity of active cognitive transducers to robotic systems we will refer to them as 'agents' mostly.

Ultimately, active cognitive transducers are of interest only in large conglomerates, interacting like 'organisms' or 'agents' of individually limited powers. A *community of active cognitive transducers* (or: a community of agents) is a time-varying set of devices which at each moment consists of finitely many active cognitive transducers of the same type sharing the same environment. Each transducer of the set makes use of a piece of its immediate environment as its private, potentially unbounded external memory, giving it the computing power of an interactive Turing machine (as stated in Theorem 4). Each transducer has its own input and output port. The ports of all transducers together present the input and output ports of the community. The number of these ports varies along with the cardinality of the community. Within the set the transducers are identified by a unique name (or address).

The active transducers (agents) can communicate by sending their outputs as inputs to other transducers identified by their addresses, or by writing a message into their environment

[4]This is a nice argument that qualitatively illustrates e.g. the revolutionary contribution of the development of the script to the development of human civilization. Namely, along the lines of the given considerations, the development of the script has promoted each literate person's information processing capacity from that of a finite-state machine to that of an (interactive) Turing machine.

in a way such that it can be read by other transducers. One can see it as if the agents move in their environment and encounter each other randomly, unpredictably or intentionally, and exchange messages. Who encounters whom, who will send a message, and the delivery time of each message is unpredictable. The idea is to capture in the model any reasonable message delivery mechanism among organisms or agents, be it signals in some format, spoken language in direct contact, snail mail, via mobile phones, the Internet, and so on, depending on the entities that are modeled. Moreover, agents are assumed to be 'mortal': they emerge and vanish also unpredictably.

The description of a community of agents at each moment in time is given by the list of names and 'programs' of all living agents at that time, and the list of all transient messages at that time (including the respective senders and addressees, the time of the expedition of each message, and the message delivery times). Note that in general most of the required parameters needed in the description of a community at a given time are *non-computable*, since according to our description of how communities function they are unpredictable. Nevertheless, communities can be described by a finite table at each moment in time, i.e. the description of a whole community at any time is always finite.

For a given community and given data, a (potentially infinite) sequence of descriptions over time enables us to recover the *evolution* of the community (its 'dynamics') as a sequence of instantaneous descriptions (configurations) of the community. However, the mechanisms that 'implement' the dynamics (e.g., the way an agent enters or leaves a community, the way it gets its identity or changes its program) are not part of the model. In [18, 20] the following result is proved for the case of 'real' agents communicating via an Internet-like infrastructure.

Theorem 5 *Communities of agents have a computational power equivalent to interactive Turing machines with an advice function whose values grow at most linearly in size with the processing time.*

Returning to the context of artificial living systems, Theorem 5 asserts that a community of active cognitive transducers has a much greater computing power than just the 'sum' of the powers of the individual transducers. Here we see the emerging super-recursive, thus non-recursive computing power arising due to the unpredictable external information that can enter into a system. Do results like Theorem 5 really mean that the corresponding systems can solve undecidable problems? The answer of course is that they cannot, unless certain assumptions are made. In order for these systems to simulate a Turing machine with advice, they need a 'cooperating environment'. Its role is to deliver the same information as is offered by the advice. Thus the result is of a non-constructive nature: both the advice and the corresponding inputs from the environment exist in principle but there is *no algorithmic way* to obtain them. In practice the assumption of the existence of external inputs suitable for the solution of a concrete undecidable problem such as the Halting Problem (cf. [6]), is not fulfilled. Hence, without such 'right' inputs, no community of cognitive transducers will solve an undecidable problem.

On the other hand, no Turing machine without an advice could simulate e.g. the (existing) human society – simply because the society develops in a completely unpredictable, non-algorithmic way. One can say that the current human society realizes a transduction that, however, emerges somehow 'all by itself', by the joint interplay of all members of the society who interact with each other and change their environment and interact with it in a completely

unpredictable manner. All members jointly play the role of an 'advice' – nonetheless the respective advice keeps emerging on-line, incrementally, and is 'blind', possessing as a whole no specific information processing, or computational intention. The same holds e.g. for the development of the Internet – it also evolves in a non-algorithmic way and therefore cannot be modeled by a single computer (without advice).

Note that due to Theorem 2, both sequences of cognitive transducers and communities of active cognitive transducers have super-Turing computing power. This is a rather surprising result, from certain perspectives. It means that in 'practice' with some exaggeration e.g. a coral colony (i.e., a stationary but growing configuration of simple living organisms) has in principle the same computational potential as e.g. a developing, dynamically changing human society. The source of the power of both entities is given by their basic cognitive powers, the potentially unlimited cardinality of the community, by their potentially unlimited life span, and by the non-computable characteristics of the community at any given time, along with the unpredictable shape of the colony or the unpredictable interaction among society members (leading to non-uniformity of the resulting system). The difference between the two is, of course, in their computational efficiency, as captured by Theorem 3. Again, theoretically, to keep pace with a polynomially fast developing human society the coral colony should grow exponentially, as seen from Theorem 3 part (d).

What remains is to answer Penrose's question from the introduction of this paper. Consider the information computed and stored in the environment in a long run by a community of agents. By virtue of Theorem 5 this is information that cannot be computed classically. Now, since each member of the community has access to this information, this information effectively plays the role of an advice and, due to this, in principle each member of the community gains a super-Turing computing power.

6 Conclusion

The results in this paper can be seen as applications of non-classical computability theory to artificial living systems. The main result explaining the emergence of a super-Turing computing potential in such systems justifies the approach, and concretely proves what is often speculated on in informal explanations (cf.[23] or, more recently [2]). It also points to the increasing role that computer science will play in problems related to understanding the nature of the emergence of the mechanisms of life and of intelligence in particular (cf. [26]). The above results also point to quite realistic instances in which the classical paradigm of classical Turing machines as the generic model of all computers, capable of capturing all algorithmic computations, is clearly insufficient. It appears that the time has come to reconsider this paradigm and replace it by its extended version – viz. interactive Turing machines with advice or an equivalent of it. Our results have also shown that this extension amounts to considering interactive, evolutionary scenarios where finite agents operating the computational devices at hand play the role of advice. For a more extended discussion of the related issues, see [18] or [21].

References

[1] J.L. Balcázar, J. Díaz, J. Gabarró: *Structural complexity I*, Second Edition, Springer, 1995.

[2] M. Burgin: How we know what technology can do, *Communications of the ACM* 44 (2001) 83-88.

[3] G. Buttazzo: Artificial consciousness: Utopia or real possibility? *Computer,* July 2001, pp. 24-30.

[4] D. Goldin: Persistent Turing machines as a model of interactive computations, in: K-D. Schewe and B. Thalheim (Eds.), *Foundations of information and knowledge systems*, First Int. Symposium (FoIKS'2000). Lecture Notes in Computer Science, Vol. 1762, Springer-Verlag, Berlin, 2000, pp. 116-135.

[5] A. Hodges: *Turing – A natural philosopher*, Phoenix, 1997.

[6] J. Hopcroft, R. Motwani, and J.D. Ullman: *Introduction to automata theory, languages and computation*, 2nd Edition, Addison-Wesley, Reading, MA, 2000.

[7] R.M. Karp, R.J. Lipton: Some connections between non-uniform and uniform complexity classes, in *Proc. 12th Annual ACM Symposium on the Theory of Computing* (STOC'80), 1980, pp. 302-309.

[8] W. Maass, C. Bishop (Eds.): *Pulsed neural networks*, MIT Press, Cambridge, 1998.

[9] P. Orponen: An overview of the computational power of recurrent neural networks, in: Finnish AI Conference (Espoo, Finland, August 2000), Proceedings, Vol. 3: *AI of Tomorrow*, Finnish AI Society, Vaasa, 2000, pp. 89-96.

[10] R. Penrose: *The emperor's new mind. Concerning computers, mind and the laws of physics*, Oxford University Press, New York, 1989.

[11] J. Šíma, J. Wiedermann: Theory of neuromata, *Journal of the ACM,* Vol. 45, No. 1, 1998, pp. 155–178.

[12] W. Thomas: Automata on infinite objects, in: J. van Leeuwen (Ed.), *Handbook of theoretical computer science*, *Vol B: Formal models and semantics*, Chapter 4, Elsevier Science Publ., Amsterdam, 1990, pp. 133-191.

[13] A.M. Turing: On computable numbers, with an application to the Entscheidungsproblem, *Proc. London Math. Soc.,* 42-2 (1936) 230-265; A correction, *ibid*, 43 (1937), pp. 544-546.

[14] A.M. Turing: Systems of logic based on ordinals, *Proc. London Math. Soc. Series* 2, 45 (1939), pp. 161-228.

[15] A.M. Turing: Computing machinery and intelligence, *Mind* 59 (1950) 433-460.

[16] L.G. Valiant: *Circuits of the Mind*, Oxford University Press, New York, 1994.

[17] J. van Leeuwen, J. Wiedermann: On algorithms and interaction, in: M. Nielsen and B. Rovan (Eds.), *Mathematical Foundations of Computer Science 2000*, 25th Int. Symposium (MFCS'2000), Lecture Notes in Computer Science Vol. 1893, Springer-Verlag, Berlin, 2000, pp. 99-112.

[18] J. van Leeuwen, J. Wiedermann: The Turing machine paradigm in contemporary computing, in: B. Enquist and W. Schmidt (Eds.), *Mathematics unlimited - 2001 and beyond*, Springer-Verlag, 2001, pp. 1139-1155.

[19] J. van Leeuwen, J. Wiedermann, J: A computational model of interaction in embedded systems. Technical Report UU-CS-02-2001, Dept. of Computer Science, Utrecht University, 2001.

[20] J. van Leeuwen, J. Wiedermann: Breaking the Turing barrier: the case of the Internet. Manuscript in preparation, February, 2001.

[21] J. van Leeuwen, J. Wiedermann: Beyond the Turing limit: evolving interactive systems, in: L. Pacholski, P. Ružička (Eds.), *SOFSEM'01: Theory and Practice of Informatics*, 28th Conference on Current Trends in Theory and Practice of Informatics, Lecture Notes in Computer Science Vol. 2234, Springer-Verlag, Berlin, 2001, pp. 90–109.

[22] K. Wagner, G. Wechsung: *Computational complexity*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1986.

[23] P. Wegner: Why interaction is more powerful than algorithms, *C. ACM* 40, (1997) 315-351.

[24] J. Wiedermann: Toward computational models of the brain: getting started, *Neural Network World,* Vol. 7, No. 1, 1997, pp. 89-120.

[25] J. Wiedermann: Towards algorithmic explanation of mind evolution and functioning, in: L. Brim, J. Gruska and J. Zlatuška (Eds.), *Mathematical Foundations of Computer Science*, 23-rd International Symposium (MFCS'98), Lecture Notes in Computer Science Vol. 1450, Springer-Verlag, Berlin, 1998, pp. 152–166.

[26] J. Wiedermann: Simulated cognition: A gauntlet thrown to computer science, *ACM Computing Surveys*, Vol. 31, Issue 3es, paper No. 16, 1999.

[27] J. Wiedermann: Intelligence as a large-scale learning phenomenon, Technical Report ICS AV CR No. 792, 1999.

[28] J. Wiedermann: The computational limits to the cognitive power of neuroidal tabula rasa, in: O. Watanabe and T. Yokomori (Eds.), *Algorithmic Learning Theory*, 10th International Conference (ALT'99), Lecture Notes in Artificial Intelligence, Vol. 1720, Springer-Verlag, Berlin, 1999, pp. 63-76.

[29] J. Wiedermann, J. van Leeuwen: Emergence of super-Turing computing power in artificial living systems, in: J. Kelemen, P. Sosík (Eds.), *Artificial Life 2001*, 6-th European Conference (ECAL 2001), Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, 2001, pp. 55-65.