DCFS 2007

V. Geffert, G. Pighizzini (eds.): Proceedings of the 9th workshop Descriptional Complexity of Formal Systems. High Tatras, Slovakia, July 20 – 22, 2007. (Pages 41 – 56)

Wireless Parallel Computing and Its Links to Descriptive Complexity

Jiří Wiedermann[(A)] Dana Pardubská[(B)]

[(A)]Institute of Computer Science – Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2 – 182 07 Prague 8 – Czech Republic
jiri.wiedermann@cs.cas.cz

[(B)]Department of Computer Science – Comenius University
Mlynská dolina – 842 48 Bratislava – Slovakia
pardubska@dcs.fmph.uniba.sk

Abstract. Wireless Parallel Turing Machine (WPTM) is a new computational model recently introduced and studied by the authors. Its design captures important features of wireless mobile computing. In this paper we survey the results related to the descriptive complexity aspects of the new model. In particular, we show a tight relationship of wireless parallel computing to alternating and synchronized alternating Turing machines. This relationship opens, e.g., the road to circuit complexity by offering an elegant WPTM characterization of bounded uniform circuit families, such as NC and NC^{i}. The structural properties of computational graphs of WPTM computations inspire definitions of new complexity measures capturing important aspects of wireless computations: energy consumption and the number of broadcasting channels used during computation. These measures do not seem to have direct counterparts in alternating computations. We mention some results related to these new structural measures, e.g., a polynomial time–bounded complexity hierarchy based on channel complexity, lying between P and PSPACE which seems to be incomparable to the standard polynomial–time alternating hierarchy.

Keywords: alternating Turing machine, simulation, simultaneous time–space complexity, wireless parallel Turing Machine

1 Introduction

Rather than asking what dynamic computational resources are needed for solving a problem, descriptive complexity is interested in the complexity of expressing a solution of a given problem using formal descriptive systems. A major part of traditional descriptive complexity has been based on alternation. This is because the known theory of alternating Turing machines (ATMs) offers a direct link from alternating computations to their description in the formalism of quantified

[(A)]The research was carried out within the institutional research plan AV0Z10300504 and partially supported by grant No. 1ET100300517.
[(B)]The research was partially supported by grant VEGA 1/3106/06.

Boolean formulae or that of uniform families of Boolean circuits (cf. [5], [6]). The main goal of this paper is to bring into attention of specialists a new universal computational model designed and studied recently by the authors, the so–called wireless parallel Turing machine (WPTM) and to demonstrate its usefulness also for investigating the descriptive complexity of problems.

There are several good reasons for studying the WPTMs. First, they capture the paradigm of wireless mobile computing. Second, they offer a new view of alternation and synchronized alternation. Third, they reveal new structural complexity measures inspired by wireless computation. Fourth, they induce a natural polynomial–time complexity hierarchy based on the channel complexity. Last but not least, the WPTMs suggest a possible realization of synchronous parallel computations using wireless technologies.

The history of WPTMs dates back to 2005 when the first conference *Computability in Europe* was organized. The present authors have been invited to contribute to the post–conference proceedings entitled *New Computational Paradigms: Changing Conceptions of What Is Computable* (with B. Cooper, B. Löwe, A. Sorbi, editors) which should appear in Springer-Verlag [8]. The paper was accepted in August 2006 but, unfortunately, as of the time of this writing (April 2007), the volume did not appear. So far, the original paper is available as a technical report [8].

The authors contributed to the volume by a novel unconventional model whose design has been inspired by the trends in wireless communication technologies. What would be the computational power and efficiency of a device consisting of a number of processors communicating one with each other via radio, on different channels? In the computational complexity theory there seem to be no computational models based on wireless communication between processors. There is an essential difference between broadcasting and a link or address–based communication: broadcasting can be heard by any number of processors which are within the reach of a sender. On the other hand, there is no possibility to "address" a certain selected processor, except dedicating a special channel to it. None of the existing computational models captures similar principles. Therefore it was of interest to design a model which would exploit the possibility of wireless parallel communication on different channels. There is one additional bonus when considering broadcasting: a link–free communication is a prerequisite of mobility. Thus, wireless parallel computing mirrors important aspects of computing performed by dynamically changing sets of mobile processors.

In order to facilitate computational complexity considerations, we have chosen a model based on Turing machines. Since we were interested in parallel computations over a dynamic sets of identical processors, we have decided to adopt the process multiplication mechanism used in the ATMs (cf. [1]). That is, a process can spawn the finite number of new processes which start to compute from the same configuration as their parent. However, any similarity with the ATMs ends here. In our model, there is no nondeterminism and there is no implicit acceptance mechanism which "automatically" terminates a computation when there is

an accepting subtree in the respective computational tree. Unlike in the ATMs, where "fathers" can transfer information only to their "sons", in our model processes can communicate in a more flexible manner. In order to communicate, the processes must establish a connection, i.e., they must "tune" to the same channel. This is achieved by writing the same number to each process's channel tape. Doing so, all processes tuned to the same channel can exchange messages. A computation terminates when all processes agree upon a result of a computation and terminate.

In the computational complexity theory, in the past there have been attempts to introduce a restricted type of additional communication into the standard model of ATMs. The respective investigations were mostly inspired by scientific curiosity, rather than by an existing technology. The prime examples of such attempts have been the synchronized alternating Turing machines (cf. [3], [4], [7]). The polynomial time on these machines turned up to be equivalent to PSPACE what has qualified these machines into a so–called second machine class. This machine class is characterized by the Parallel Computation Thesis stating that parallel time complexity on any second machine class model is polynomially related to sequential space complexity of an equivalent deterministic Turing machine (cf. [2]). However, it has also turned up that the synchronized Turing machines are slightly more powerful than the ATMs (under the standard assumptions like LOGSPACE $\neq$ PTIME, etc.), since the former machines make an optimal use of their space: polynomial time on these machines equals their logarithmic space [4], [7]. Nevertheless, the WPTMs represent a completely different model than alternating or synchronized alternating machines. They represent a parallel deterministic model of computation, with an explicit, reconfigurable inter–processor communication mechanism which serves also for acceptance purposes.

The results from [8] have shown that the WPTMs are computationally related both to the alternating and synchronized alternating Turing machines. Under the self-explaining notation and reasonable assumptions on $T(n)$, in [8] it has been established that $\mathsf{ATIME}(\mathsf{T}(\mathsf{n})) \subseteq \mathsf{WTIME}(\mathsf{T}(\mathsf{n}))$ and $\mathsf{WTIME}(\mathsf{T}(\mathsf{n})) \subseteq \mathsf{ATIME}(\mathsf{T}^2(\mathsf{n}))$, i.e., a polynomial-time equivalence of both models: WPTIME = APTIME. Both respective simulations work in space of size $O(T(n))$. From these results it has followed that the WPTMs do belong to the second machine class and similarly as the synchronized Turing machines they use their space optimally: WPTIME = WLOGSPACE.

In the subsequent paper [9] we continued our studies of the computational relations between WPTMs and ATMs. Our new simulations have shown an ultimate improvement of the previous results—namely mutual simulations of both models preserving their simultaneous time and space complexities for a wide range of these complexity measures. The new results have opened the road to the descriptive complexity. For instance, these results allowed a neat characterization of uniform circuit families NC^{i} of polynomial size and of depth $(\log n)^i$ in terms of WPTM computations.

In the present survey paper we will briefly describe the results mentioned previously and complement them by new results related to the structural complexity. We will introduce new parallel complexity measures induced by the structural properties of the underlying computational graphs. First, we will be interested in the number of (parallel) wireless connections realized by a processor during a computation. This number is captured by so–called listening complexity. When simulating an ATM machine it will turn out that this complexity is linearly related to the number of alternations of the simulated alternating computation. The reverse relation is open. We will also be interested in the maximal number of channels used in a WPTM computation. So far, we do not know of any direct counterpart of this measure in alternating computations. This measure leads to a new interesting hierarchy of polynomially time–bounded complexity classes positioned between P and PSPACE (assuming the disjointness of the latter two classes).

The structure of the paper is as follows. In Section 2 we recall an informal definition of a WPTM and that of its computation. In Section 3 we sketch a simulation of a WPTM by an ATM which is simultaneously linear in time and space. In Section 4 we prove a reverse simulation with the similar complexity bounds as in the previous case. Both simulations allow computations in sub-linear space. In Section 5 we recapitulate the relation of the basic WPTM complexity classes to the standard fundamental complexity hierarchy and in Section 6 we sketch the characterization of bounded circuit families by the WPTMs. In Section 7 the (one way) relationship between the listening complexity and the number of alternations is established. Finally, in Section 8 we introduce a hierarchy based on channel complexity and study its basic properties. Conclusions appear in Section 9.

From space reasons, the proofs of our theorems and corollaries are only sketched and the reader should usually refer to the original sources. Results from Section 7 and 8 are new.

2 Wireless Parallel Turing Machine

Wireless Parallel Turing Machine (WPTM) is a parallel deterministic Turing machine in which the mechanism of the process multiplication used in ATMs (cf. [1]) and communication "via radio" has been adopted. That is, a process can spawn a finite number of new processes which start to compute from the same configuration as their parent. In order to communicate, the processes must establish a connection, i.e., they must "tune" to the same channel. This is achieved by writing the same *channel–number* to each process's *channel tape.* All processes tuned to the same channel can then exchange "messages". We model a message by an element of a state control, so-called *communication state.* A message broadcasted by a process at time t is heard at time $t+1$ by all listening processes[1] which have

[1]Note that this description and the subsequent definition differs from those given in [8] and [9] by explicitly splitting the set of working states into so–called broadcasting, listening and

tuned in the same channel at time $t+1$. We require that all messages broadcasted on a specified channel in a given time are the same. A computation terminates when all processes agree upon a result of computation and terminate.

Definition 1 (Informal). A k-tape wireless parallel Turing machine (WPTM) with a separate read–only input tape and a separate channel tape is a twelve–tuple $M = (k, B, L, Q, R, \Sigma, \Gamma, \Delta, q_0, \varepsilon, q_{accept}, q_{reject})$, where

- $k \geq 1$ is the number of work tapes;
- B, L, Q, and R are the disjoint finite sets of *broadcasting, listening, local* and *communication* states respectively;
  $K = B \cup L \cup Q$ is called a set of *working states*. *Initial state* $q_0 \in Q$;
  R contains three distinguished states: empty communication state ε and states q_{accept}, q_{reject};
- Σ, Γ are finite input and work tape alphabets; $\$ \notin \Sigma$ is an end-marker, $\sharp \in \Gamma$ is the blank symbol, $\sharp \notin \Sigma$
- $\Delta \subseteq K \times R \times (\Sigma \cup \{\$\}) \times \Gamma^{k+1} \times K \times R \times (\Gamma - \{\sharp\})^{k+1} \times \{-1, 0, 1\}^{k+2}$ is the *next move relation*; the elements of Δ are called transitions. A transition with the new communication state $r' \neq \varepsilon$ is called a *broadcasting transition*[2]; it is said to broadcast state $r' \in R$. The broadcasting transitions pertinent to the same head configuration[3] must all broadcast the same communication state.

A *configuration* of a WPTM M is an element of $K \times R \times \Sigma^* \times ((\Gamma - \{\sharp\})^*)^{k+1} \times \mathbb{N}^{k+2}$. Based on the type of a working state a configuration is called a *broadcasting, listening* or *local* configuration.

A configuration is *tuned to channel* c if it has string c written to the left from the current channel tape head position. A broadcasting configuration tuned to c, to which a transition with the new communication state $r' \neq \varepsilon$ applies is said to broadcast r' on channel c.

A *computational step* can be seen as a two–phase process. In the first phase—in a *simple step*—all applicable transitions are applied to each configuration. Two or more transitions applicable to a particular configuration result in spawning of new processes. Formally, a spawning corresponds to the universal branching in ATMs. In the second phase, the communication between the processes is realized: the communication states of listening configurations are rewritten by the symbols broadcasted by transitions applied in the simple step. Of course, doing so, both the broadcasting and the listening configurations must have been tuned to the same channels. The computation gets aborted if there are at least two processes broadcasting different communication states on the same channel.

For any input w to M the *computational graph* $G(w)$ of M on that input is a rooted, directed, possibly infinite acyclic multi-graph whose nodes are configura-

local states.

[2] W.l.o.g. we assume that in any broadcasting state only symbols $\neq \varepsilon$ are broadcasted

[3] A *head configuration* of M represents the working and the communication state of the finite control and the contents of cells scanned by each head.

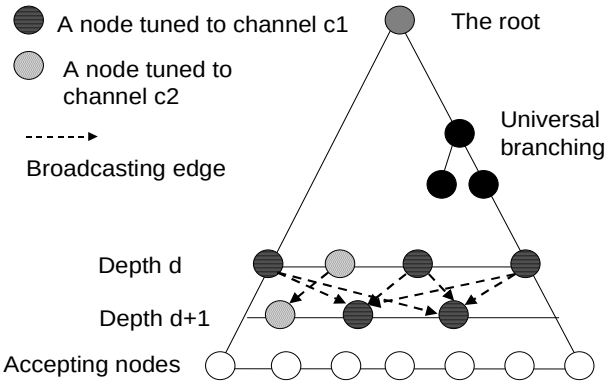

Figure 1. The computational graph of M

tions of M and edges correspond to transition and communication links (Fig. 1). The *depth* of a node in $G(w)$ is its distance from the root. For any $d \geq 0$, there exists a communication link between any node α in depth d and any node γ in depth $d+1$ if and only if α is a broadcasting node, γ is a listening node and both nodes are tuned to the same channel. We say M accepts input word w if the computational graph $G(w)$ is finite and all leaves of $G(w)$ are terminal configurations in communication state q_{accept} tuned to the same channel. Note that the computational graph is defined in an entirely deterministic manner.

There are several complexity measures relevant to WPTMs. The space complexity of a computational graph $G(w)$ is the maximum space, over all tapes, of any configuration in $G(w)$; the channel space of $G(w)$ is defined similarly, but only channel tape is considered. We say a WPTM M *operates in space* $S(n)$ if for every string w accepted by M ($w \in L(M)$) of length n the computational graph $G(w)$ is of space complexity at most $S(n)$; similarly, M *operates in time* $T(n)$ if for every string $w \in L(M)$ of length n a computational graph $G(w)$ is of depth at most $T(n)$.

The *listening complexity* $LC(n)$ of M is defined as the maximal number of listening configurations along any root–to–leaf path following the transition edges only, in the computational graph $G(w)$, taken over all accepted inputs w of length n. The *channel complexity* $CC(n)$ of M is defined as the maximal number of channels used in a computation of M, taken over all accepted inputs w of length n.

Both listening complexity and channel complexity are complexity measures of great importance in wireless computing. For wireless computations in which a processor broadcasts and listens for approximatively the same time the listening complexity captures the maximal number of wireless connections realized during a computation by a processor and hence is related to the energy consumption of that processor in broadcasting and listening mode during a computation. The channel complexity is related to the broadband on which the processes can broadcast on various channels simultaneously.

In order to be able to study also computations with sub–linear times, we extend our model by so–called *index tape* to which non-negative binary numbers can be written. If there is a number i written in binary on the index tape and the machine enters a special, so–called *index state* q_{index}, then the input head on the input tape relocates itself in a single move to the i-th cell of that tape, for any $1 \leq i \leq n$, and to the $(n+1)$-st cell otherwise. For a similar mechanism in the context of ATMs, cf. [1]. In general, one can also consider a model with several channel tapes and also nondeterministic versions of the previous models, but in this paper we will restrict ourselves to the basic variant of a WPTM as defined above.

In a given space, the WPTMs are able to compute for super-exponentially long time.

Proposition 2. *If M is a WPTM of time complexity $T(n) \geq \log n$ and space complexity $S(n)$, then there exists a constant $c > 0$ depending only on M such that for any $n > 0$, $S(n) \leq T(n) \leq c^{c^{S(n)}}$.*

Sketch of the proof: Clearly, $T(n) \geq S(n)$ since time at least $S(n)$ is needed in order to write down $S(n)$ symbols on any of M's tapes. On the other hand, for a suitable constant $d > 0$ depending only on M, a WPTM of space complexity $S(n)$ can generate up to $O(d^{S(n)})$ different processes, each of space complexity $S(n)$ which all together represent space of size $O(S(n)d^{S(n)})$. In such a space, one can generate up to $c^{c^{S(n)}}$ different configurations for a suitable $c > d$. □

3 Linear time-space simulation of wireless communication by alternation

Now we present the theorem stating the relationship between the computations of WPTMs and those of ATMs; we will also sketch the main ideas of the respective proof. For details, the reader is referred to [9].

Theorem 3. *Let W be an one-tape WPTM of time complexity $T(n) \geq n$ and of space complexity $S(n)$. Then W can be simulated by an alternating Turing machine A simultaneously in time $O(T(n))$ and in space $O(T(n))$. Moreover, if $S(n)$ is fully space constructible in time $T(n)$ then A is of simultaneous time complexity $O(T(n))$ and space complexity $O(S(n) + \log T(n) + n)$.*

Sketch of the proof: Consider the computational graph $G(w)$ of W on an input w of size n. Obviously, A has no problem in simulating the computation of W along the spawning (transition) edges in G (we will omit parameter w in G whenever no confusion can arise) since the respective configurations are computed deterministically from their predecessors. There is no problem when arriving at local or broadcasting configurations. However, arriving into a new

listening configuration in G, A guesses whether there was a broadcast to that configuration, or not.

In the former case, when the new configuration has been modified by broadcasting, in addition to computing the new configuration by applying a simple transition to its predecessor, A must also guess the new communication state r' in the new configuration. Doing so, A will immediately start a recursive verification process. Seen in G, this process "climbs" from the given listening configuration ℓ towards the root of G. In such ℓ, the process splits itself in a universal mode into three processes: the first one verifies the upward path along the transition edge, the second one along the broadcasting edge; the third process verifies whether the broadcast has been conflict–free (i.e., whether on the given channel all processes have broadcasted the same symbol).

The first process has to guess the simple transition δ which has been applied in order to reach ℓ. After guessing δ, the process computes the spawning predecessor of ℓ and recursively verifies this predecessor. The second process guesses the entire broadcasting configuration (note that it must have been tuned to the same channel as ℓ, i.e., the contents of the channel tape need not be guessed) and recursively verifies the guessed configuration. The recursion terminates successfully by reaching the initial configuration. The third process recursively checks in universal mode for all communication symbols different from r' whether there were broadcasting configurations sending on the same channel as ℓ has been tuned into. This recursion ends successfully when no such configurations reachable from the root of G have been found, in the given depth or size (depending on the assumption of the theorem).

Discovering an accepting verification subtree by the verification process must happen in time $T(n)$ at the latest if such a subtree exists. Therefore the size of guessed configurations can grow up to $O(T(n))$. However, if $S(n)$ is constructible in time $T(n)$, then we can stop the verification process whenever a configuration longer than $S(n)$ is reached.

In the case when no modification via broadcasting has been guessed, A has to verify this guess. This is done similarly as in the previous case by computing all configurations that could have influenced the given configuration by broadcasting. Note that such configurations need not occur in G since they need not be reachable from the root. The verification ends successfully when such configurations do not exist or are not reachable from the initial configuration in the given depth or space (again depending on the assumption of the theorem).

It is important that thanks to the deterministic evolution of WPTMs computations the previously described verification processes can be repeated with the same results when restarted from the same configuration. Such repetitions will inevitably happen since the verification processes climbing towards the root of G get invoked at each level in G.

Proceeding in accordance with the previous idea we get a quadratic time simulation since climbing against the direction of broadcasting edges at each level we have to guess the entire configuration (this has been the algorithm described in

[8]). However, this can be avoided by using a novel technique developed in [9]. At the heart of this technique there is the idea of successive building configurations from so–called kernels. A kernel of a configuration is given by the contents of its channel tape, position of all heads and contents of all cells scanned by the tape heads on the working tapes and on the input tape. The previous verification process can then be modified so that it starts by only guessing the kernel in a constant time (since the channel tape is "copied" in one step, thanks to the mechanism of configuration multiplication in universal states) and then, in the subsequent recursive calls, at each step on each tape the currently built part of a configuration is being "encased" by adequately guessed symbols, in accordance with the guessed movements of the respective tape head, so as the kernel finally develops into a full configuration. If properly done, the whole verification process can run in linear time as required. □

The simulation from the proof of Theorem 3 can be modified for WPTMs and ATMs equipped by the index tape. The treatment of the index tape is not completely trivial, see [9] for the details.

Corollary 4. *Let A, W be as in Theorem 3. If both A and W are equipped by the index tape, $T(n) \geq \log n$ and $S(n)$ is fully space constructible in time $T(n)$, than A is of simultaneous time complexity $O(T(n))$ and space complexity $O(S(n) + \log T(n))$.*

4 Linear time-space simulation of alternation by wireless communication

Now we turn our attention to the simulation of an ATM by a WPTM. We assume that both machines are equipped by the index tapes.

Theorem 5. *Let $T(n)$ be a time–constructible function, let $S(n) \geq \log n$ be a space constructible function in time $T(n) \geq \log n$. An ATM A of time complexity $T(n)$ and of space complexity $S(n)$ can be simulated by a WPTM W simultaneously in time $O(T(n))$ and in space $O(S(n))$.*

Sketch of the proof: The basic idea of simulation is simple: W simulates A by spawning the same processes as A would. In addition to the current configuration W also remembers its father's configuration for the purpose of future communication. Reaching level $T(n)$ W checks whether the acceptance condition of A has been satisfied. To that end W starts a verification process in each reached configuration of A. Seen in T, the computational tree of A, the purpose of this process is to propagate the *quality* (accepting, rejecting, undefined) of each reached configuration, combined with the quality of its sibling, upwards to the root of T. The rules for combining the quality are the standard ones holding for the alternating computations (cf. [1]). In more details, the verification proceeds as follows: the sons of the common father report, one after the other, in an agreed-upon

ordering, their quality to their father. They do so on a channel whose number is given by the father's configuration. Obviously, the cost of this step is linear in the space complexity of A. After obtaining the qualities from all its sons, a father computes its own quality and proceeds recursively. This approach leads to simulation of A by W in quadratic time and linear space. The quadratic time is caused by the necessity of "re–tuning" each time we climb one level in tree T (note that a WPTM can broadcast only on a channel whose number finds itself to the left of the head on the channel tape).

The complexity of the lastly mentioned re–tuning can be decreased by using a different representation of configurations on the tapes of W combined with the technique of amortized complexity. The idea is as follows: when descending T, we use the so–called *explicit representation* of the corresponding A's configurations only in depths which are multiplies of $S(n)$ (here we need the assumption on the constructibility of $S(n)$). The respective depths in T are called *reference levels*. This term is needed for introducing the notion of an implicit representation of a configuration. An *implicit representation* of a configuration c_2 w.r.t. both the sequence $\sigma = \delta_1, \delta_2, \ldots, \delta_k$ of transitions and configuration c_1 is the string $c_1\#\delta_1\#\ldots\#\delta_k$ if and only if c_2 is obtained from c_1 by subsequently applying transitions from σ to c_1. Thus, a configuration c_2 which is at distance $k < S(n)$ from configuration c_1 on some reference level will be represented as $c_1\#\delta_1\#\ldots\#\delta_k$. Now the idea emerges: at the reference levels, the configurations of A will be remembered by W both in the explicit form and in the implicit form w.r.t the configuration at the next upper level. At all other depths we only use implicit representations. Then, using the previous algorithm, on non–reference levels we broadcast at channels whose numbers are given by the implicit configurations. In such cases, re-tuning to the next level can be done in time $O(1)$. Reaching a reference level, the channel number to which the process should re–tune (i.e., a configuration on the next upper level in the implicit form) has already been prepared in the node at hand. The re–tuning to the implicit form takes time $O(S(n))$, but this time gets amortized since the re–tuning happens only once after each $S(n)$ steps. □

5 WPTMs and fundamental complexity classes

In what follows, in addition to the standard deterministic complexity classes such as LOGSPACE, PTIME, PSPACE, and EXPTIME we will also use their analogues defined for the alternating and wireless parallel Turing machines. These classes will be denoted by prefixing the standard complexity classes listed before by A or W, respectively.

Concentrating merely to the time complexity of the respective simulations in the preceding two theorems the following corollary characterizing the power of WPTM's polynomial time is a consequence of the latter theorems and of known properties of alternating complexity classes (cf. [1]).

Corollary 6. *For any time constructible function $T(n)$ with $T(n) \geq n$,*

$$\bigcup_{k>0} \mathsf{WTIME}(\mathsf{T}^{\mathsf{k}}(\mathsf{n})) = \bigcup_{k>0} \mathsf{ATIME}(\mathsf{T}^{\mathsf{k}}(\mathsf{n})) \quad \text{i.e.,} \quad \mathsf{WPTIME} = \mathsf{APTIME} = \mathsf{PSPACE}$$

Next, we turn our attention to the relationship between deterministic and wireless space. This relationship cannot be inferred directly from the previous theorems by focusing merely to the space complexity, since in Corollary 4 the space on the simulating ATM machine depends both on the space complexity $S(n)$ and the logarithm of time complexity $T(n)$ of the simulating WPTM machine. Due to Proposition 2, $\log T(n)$ might be as high as $c^{S(n)}$ for some machines.

Theorem 7. *For any $S(n) \geq \log n$*

$$\mathsf{WSPACE}(\mathsf{S}(\mathsf{n})) = \bigcup_{c>0} \mathsf{DSPACE}(\mathsf{c}^{\mathsf{S}(\mathsf{n})})$$

$$\mathsf{WLOGSPACE} = \mathsf{PSPACE} = \mathsf{WPTIME},\ \mathsf{WPSPACE} = \mathsf{EXPSPACE} = \mathsf{WEXPTIME}$$

Sketch of the proof: We sketch the simulations yielding the result. In what follows D is used to denote a single tape deterministic Turing machine while M denotes a WPTM.

Simulation of D by M mirrors the proofs of similar theorems known in the theory of synchronized alternation (cf. [3], [7]). I.e., the contents of cells of D are kept in processes of M spawning a new process each time the head of D on its work tape enters a blank cell. On its working tape, process number i remembers the symbol written in the i–th cell of the $D-$th tape and the state of D if the head scans the i–th cell of D. Number i in binary is stored at the channel tape of the i–th process and it is used to communicate Ds head movements with the neighboring cell/process. M simulates D by following its moves and updating the information in the processes corresponding to the respective updated cells and state changes of D. The "head movements" from the i–th process (cell) are realized by broadcasting the respective "message" to the cell's left or right neighbor on channel $(i-1)$ or $(i+1)$ whose number equals the index of that cell. Thus, re–tuning a channel amounts to adding or subtracting 1 from the current channel number. Clearly, channel space complexity of the simulating machine is $O(\log S(n))$ while working space complexity is $O(1)$.

To simulate M by D consider the computational graph T_M of M on input w. Each configuration in T_M is of size $O(S(n))$; in this estimate the input head position is included thanks to our assumption on the size of $S(n)$. If w is accepted by M then at most $O(c_1^{S(n)})$ different processes can occur in M. It follows that for a suitable $c > c_1$ D has enough space to write down all the respective configurations and within this space it can easily keep track of all M's actions. D will accept if and only if its computation will correspond to an accepting tree of M on w.

With the help of Corollary 4 the result holds also for the WPTMs and TMs equipped by the index tapes. □

From Corollary 6 and Theorem 7 we see that logarithmic space and polynomial space and time coincide both for synchronized alternating (cf. [7] and [4]) and wireless parallel Turing machines. Thus, w.r.t. these classes both models are equivalent. Especially note that similarly as synchronized alternating Turing machines, WPTMs use their space in an optimal manner—e.g., "wireless" polynomial time equals "wireless" logarithmic space.

6 Wireless computing and uniform circuits

Now we will continue our quest toward descriptive complexity issues in wireless computing. To that end we establish the connection between the WPTM computations and the circuit computations.

Let $\mathsf{WTISP}(\mathsf{T}(\mathsf{n}), \mathsf{S}(\mathsf{n}))$ denote the class of WPTM computations (with the index tape) of a simultaneous time complexity $T(n)$ and space complexity $S(n)$, and $\mathsf{ATISP}(\mathsf{T}(\mathsf{n}), \mathsf{S}(\mathsf{n}))$ denote an analogous class for ATMs.

Let $\mathsf{USIDE}(\mathsf{S}(\mathsf{n}), \mathsf{T}(\mathsf{n}))$ be the uniform family of bounded fan-in circuits of size $S(n)$ and depth $T(n)$ (where $\mathsf{U} \in \{\mathsf{U_E}, \mathsf{U_E^*}\}$, using the notation from [6]). It is known that $\mathsf{ATISP}(\mathsf{T}(\mathsf{n}), \mathsf{S}(\mathsf{n})) = \mathsf{USIDE}(2^{\mathsf{O}(\mathsf{S}(\mathsf{n}))}, \mathsf{T}(\mathsf{n}))$, for $S(n), T(n) \geq \log n$ deterministically computable in space $S(n)$ [6].

Let $\mathsf{NC}^{\mathsf{i}} =_{\mathsf{def}} \mathsf{USIDE}(\mathsf{n}^{\mathsf{O}(1)}, (\log \mathsf{n})^{\mathsf{i}})$ be the class of all sets $A \subseteq \{0,1\}^*$ for which there is a uniform circuit family of polynomial size and of depth $(\log n)^i$, let $\mathsf{NC} =_{\mathsf{def}} \bigcup_{\mathsf{i} \geq 0} \mathsf{NC}^{\mathsf{i}}$.

By these relations we get further consequences of the previous results:

Corollary 8.

(i) for $\log n \leq S(n) \leq T(n) \leq c^{S(n)}$, *with* $S(n)$ *and* $T(n)$ *deterministically computable in space* $S(n)$ *we have*
$\mathsf{WTISP}(\mathsf{T}(\mathsf{n}), \mathsf{S}(\mathsf{n})) = \mathsf{ATISP}(\mathsf{T}(\mathsf{n}), \mathsf{S}(\mathsf{n})) = \mathsf{USIDE}(2^{\mathsf{O}(\mathsf{S}(\mathsf{n}))}, \mathsf{T}(\mathsf{n}))$;
(ii) $\mathsf{WTISP}((\log \mathsf{n})^{\mathsf{i}}, \log \mathsf{n}) = \mathsf{NC}^{\mathsf{i}}$
(iii) $\mathsf{WTISP}((\log \mathsf{n})^{\mathsf{O}(1)}, \log \mathsf{n}) = \mathsf{NC}$

Thus, simultaneously bounded WPTM computations characterize exactly uniformly bounded circuit families.

7 Number of alternations and listening complexity

It appears that there is a neat relationship between the number of alternations of an ATM and the listening complexity of a simulating WPTM.

Theorem 9. *Let* $T(n), A(n)$ *be time constructible functions and* $S(n)$ *be a space constructible function. Let* A *be an ATM of time complexity* $T(n)$ *and space complexity* $S(n)$ *making* $A(n)$ *alternations. Then* A *can be simulated by a WPTM* W *of time complexity* $O(T(n))$, *space complexity* $O(S(n) + \log T(n))$ *and of listening complexity* $LC(n) = O(A(n))$.

Sketch of the proof: The starting idea is that one used in the simulation described in the proof of Theorem 5. From the viewpoint of the statement we are after, the simulation from Theorem 5 makes too many listenings: when evaluating the computational tree T of A, the quality of configurations is collected with the help of broadcasting proceeding level by level, i.e., the number of listenings needed gets proportional to $T(n)$.

To decrease the listening complexity we will evaluate the tree in a bottom–up manner, proceeding only through alternating nodes (i.e., all other nodes will be skipped from consideration); the root and leaves of T are considered alternating nodes too. First of all, we will assume that all paths in T are of length $T(n)$. This can be achieved by keeping a counter of size $\sim \log T(n)$ in each process. Moreover, while descending T, W remembers in each configuration its own depth as well as the alternating configuration in the nearest higher level in T and the number of that level.

Evaluation of a subtree S whose all leaves are alternating nodes in the nearest depth, rooted in an alternating configuration a, is done in three steps: in the first, second, and third step, respectively, all leaves with accepting, rejecting, and undefined quality, respectively, send their evaluation to a on channel whose number is given by a concatenated with the level number of a. Based on this information A can compute its own quality.

Note that in order the previous evaluation procedure to work correctly the communication in S requires good timing. Moreover, to save listenings, the roots of alternating subtrees must be switched on to the listening mode only at times when their leaves are about to report to the respective roots. This is solved by implementing a triggering mechanism which makes use of the before mentioned counter. The mechanism works as follows: each process continues counting until $T(n)$. Reaching the maximal value all processes start count down towards zero leaving enough time between individual counts for performing the three–step evaluation procedure described above. Now the simulation preserves the following invariant: in the count–down time i, $T(n) \geq i \geq 0$, the alternating nodes whose nearest alternating root is at level i will start the three–step evaluation procedure and simultaneously, the alternating nodes at level i (i.e., the roots of the corresponding subtrees) switch to the listening mode.

Note that the number of listening nodes along each spawning path in G is proportional to the number of alternations of A. □

The existence of similar result for an ATM simulating a WPTM is an open problem.

8 Channel complexity hierarchy

Channel complexity is a practically motivated complexity measure in wireless computation which reflects the number of different channels on which processes of a parallel wireless computation can communicate simultaneously. Intuitively,

more channels can contribute to a better use of parallelism. This motivates the study of classes of wireless computations of bounded channel complexity. We will restrict our attention to logarithmically space–bounded and polynomially time–bounded WPTM computations of bounded channel complexity. In what follows, poly and exp will denote, as usually, the classes of functions of form n^k or c^{n^k}, respectively, for any $k, c > 0$.

Definition 10. $\Lambda\mathsf{S}^{\mathsf{log}}_{\mathsf{c(n)}}$ denotes the class of logarithmically space bounded WPTM computations of channel complexity $c(n)$, for any $0 \leq c(n) \leq \mathsf{poly}$. Similarly, $\Lambda\mathsf{T}^{\mathsf{poly}}_{\mathsf{c(n)}}$ denotes the class of polynomially time–bounded WPTM computations of channel complexity $c(n)$, for any $0 \leq c(n) \leq \mathsf{exp}$.

Obviously, $\Lambda\mathsf{S}^{\mathsf{log}}_{0} = \mathsf{DLOG}$ since without the wireless communication the WPTM computations reduce to deterministic Turing machine computations. On the other hand, $\Lambda\mathsf{S}^{\mathsf{log}}_{\mathsf{poly}} = \mathsf{WLOGSPACE} = \mathsf{PSPACE}$ by virtue of Theorem 7. Nevertheless, there is no wireless logarithmic–space hierarchy based on channel complexity:

Theorem 11. $\Lambda\mathsf{S}^{\mathsf{log}}_{1} = \Lambda\mathsf{S}^{\mathsf{log}}_{\mathsf{poly}} = \mathsf{PSPACE}$

Sketch of the proof: To simulate logarithmically space–bounded WPTM W_1 using a polynomial number of channels by a single channel machine W_2 we use the so–called *channel multiplexing.* The idea is to simulate one parallel step of W_1 by one cycle of polynomial number of parallel steps of W_2. More precisely, the communication on channel i on W_1 in time j is in W_2 simulated by $i-$th step of the $j-$th cycle. □

As far as the wireless polynomial–time hierarchy is concerned, observe that $\Lambda\mathsf{T}^{\mathsf{poly}}_{0} = \mathsf{P}$ and $\Lambda\mathsf{T}^{\mathsf{poly}}_{\mathsf{exp}} = \mathsf{WPTIME} = \mathsf{PSPACE}$ (the latter equality is by Theorem 7). For a single channel we have the following relationship:

Theorem 12. $\mathsf{NP} \cup \mathsf{coNP} \subseteq \Lambda\mathsf{T}^{\mathsf{poly}}_{1} \subseteq \Sigma^{\mathsf{poly}}_{2}$

Sketch of the proof: The first inclusion is almost obvious—a single–channel WPTM M_2 constructs the respective computational tree and the leaves report to the root similarly as in the proof of Theorem 9. The simulating Σ_2 machine M_1 in the second inclusion first guesses a "broadcasting schedule" for M_2, i.e., information what symbol is broadcasted at what time. The corresponding table is of polynomial size. Then M_1, in universal mode simulates M_2 and instead of broadcasting/listening each processor of M_1 consults the table for the necessary entry to learn what symbol has been broadcasted (if any). □

The polynomial–time wireless hierarchy collapses for channel complexity between 1 and poly, the rest remains open:

Theorem 13. $\mathsf{P} \subseteq \Lambda\mathsf{T}^{\mathsf{poly}}_{1} = \Lambda\mathsf{T}^{\mathsf{poly}}_{2} = \ldots \Lambda\mathsf{T}^{\mathsf{poly}}_{\mathsf{poly}} \subseteq \ldots \subseteq \Lambda\mathsf{T}^{\mathsf{poly}}_{\mathsf{exp}} = \mathsf{PSPACE}$

Sketch of the proof: The identities among the complexity classes are proved with the help of channel multiplexing as in Theorem 11. □

By virtue of this theorem, the P vs. PSPACE problem now translates in the question, whether we can separate polynomially time–bounded wireless parallel computations with channel complexities 0 and 1, and between poly and exp.

It is interesting to compare the hierarchy from Theorem 13 with the standard alternating polynomial–time hierarchy. While both ends of these hierarchies coincide, there does not seem to be a simple relationships between the rest of the hierarchies.

9 Conclusion

Our investigations have revealed a tight relationship between the basic complexity measures on ATMs and WPTMs. It appears that within the range of an exponential dependency between time and space complexity functions, simultaneous time and space bounded computations of a WPTM are linearly equivalent to the corresponding simultaneous measures on an equivalent ATM. Thanks to this equivalency the relationship of ATMs to the quantified Boolean formulae and uniform Boolean circuits translates directly to WPTMs. Thus, the WPTMs offer an alternative tool for characterizing the alternating computations or, in general, for characterizing any computation described by an equivalent formal tool from the domain of descriptive complexity. The WPTM computations have interesting structural properties induced by the nature of wireless computing. First of all, it is the listening complexity capturing the time a processor must be "on air" during a wireless computations. We have shown that this time is related to the number of alternations. The second structural measure we have considered was the channel complexity capturing the broadband of a wireless computation. We have seen that this measure has lead to a new interesting hierarchy of polynomially time–bounded complexity classes positioned between P and PSPACE (providing the disjointness of both classes).

A further potential of wireless parallel computing in computational and descriptive complexity studies remains to be seen in the future.

References

[1] Chandra, A., Kozen, D., Stockmeyer, L.: Alternation. Journal of the ACM, Vol. 28, pp. 114–133, 1981.

[2] van Emde Boas, P.: Machine Models and Simulations. In: J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science*, Vol. A, Elsevier Science Publishers, Amsterdam, 1990, pp. 1–66.

[3] Geffert, V.: A Communication Hierarchy of Parallel Computations. Theor. Comput. Sci., Vol 198, No. 1-2, pp. 99–130, 1998.

[4] Hromkovič, J., Karhumäki, J., Rovan, B., Slobodová, A.: On the power of synchronization in parallel computations. Discrete Appl. Math. 32, 1991, 155-182.

[5] Immerman, N.: Descriptive Complexity. Springer-Verlag, New York, 1999, 268 p.

[6] Vollmer, H.: Introduction to Circuit Complexity. Springer-Verlag, Berlin, Heidelberg, 1999, 270 p.

[7] Wiedermann, J.: On the Power of Synchronization. Elektronische Informationsverarbeitung und Kybernetik (EIK), Vol. 25, No. 10, pp. 499–506, 1989.

[8] Wiedermann, J., Pardubská, D.: On the Power of Broadcasting in Mobile Computing. In: B. Cooper, B. Löwe, A. Sorbi (eds.), New Computational Paradigms: Changing Conceptions of What Is Computable. To appear in Springer-Verlag, New York Inc., 2007. See also Technical Report V-944, ICS, October 2005, Prague (accessible via http://www.cs.cas.cz).

[9] Wiedermann, J., Pardubská, D.: Computing by Broadcasting: Alternation in Disguise (Extended Version). Technical Report V-975, ICS, Prague, September 2006 (accessible via http://www.cs.cas.cz).