The complexity of small universal Turing machines: a survey ⋆

Damien Woods

Department of Computer Science, University College Cork, Ireland.

Turlough Neary

Boole Centre for Research in Informatics, Department of Mathematics, University College Cork, Ireland.

Abstract

We survey some work concerned with small universal Turing machines, cellular automata, tag systems, and other simple models of computation. For example it has been an open question for some time as to whether the smallest known universal Turing machines of Minsky, Rogozhin, Baiocchi and Kudlek are efficient (polynomial time) simulators of Turing machines. These are some of the most intuitively simple computational devices and previously the best known simulations were exponentially slow. We discuss recent work that shows that these machines are indeed efficient simulators. As a related result we also find that Rule 110, a well-known elementary cellular automaton, is also efficiently universal. We also mention some old and new universal program-size results, including new small universal Turing machines and new weakly, and semi-weakly, universal Turing machines. We then discuss some ideas for future work arising out of these, and other, results.

Key words: small universal Turing machines, computational complexity, polynomial time, simulation, tag systems, cellular automata
PACS:

⋆ D. Woods is funded by Science Foundation Ireland grant number 04/IN3/1524. T. Neary is funded by the Irish Research Council for Science, Engineering and Technology, and by Science Foundation Ireland Research Frontiers Programme.

Email addresses: `d.woods@cs.ucc.ie` (Damien Woods), `tneary@cs.nuim.ie` (Turlough Neary).
URLs: `http://www.cs.ucc.ie/~dw5/` (Damien Woods), `http://www.cs.nuim.ie/~tneary/` (Turlough Neary).

1 Introduction

In this short survey we explore results related to the time and size complexity of universal Turing machines, and some related models. We also discuss results for variants on the Turing machine model to give an idea of the many strands of work in the area. Of course the choice of topics is incomplete and reflects the authors' interests, and there are other interesting surveys that may interest the reader [29,21,27].

In 1956 Shannon [71] considered the question of finding the smallest possible universal Turing machine, where size is the number of states and symbols. In the early Sixties, Minsky and Watanabe had a running competition to see who could find the smallest universal Turing machine [36,39,77,78]. Early attempts [15,78] gave small universal Turing machines that efficiently (in polynomial time) simulated Turing machines. In 1962, Minsky [39] found a small 7-state, 4-symbol universal machine. Minsky's machine worked by simulating 2-tag systems, which where shown to be universal by Cocke and Minsky [6]. Rogozhin [65] extended Minsky's technique of 2-tag simulation and found small machines with a number of state-symbol pairs. Subsequently, some of Rogozhin's machines were reduced in size or improved by Robinson [63], Rogozhin [68], Kudlek and Rogozhin [19], and Baiocchi [4]. All of the smallest known 2-tag simulators are plotted as circles in Figure 1. Also, Table 1 lists a number of these machines.

Unfortunately, Cocke and Minsky's 2-tag simulation of Turing machines was exponentially slow. The exponential slowdown was essentially caused by the use of a unary encoding of Turing machine tape contents. Therefore, for many years it was entirely plausible that there was an exponential trade-off between program size complexity on the one hand, and time/space complexity on the other; the smallest universal Turing machines seemed to be exponentially slow.

Figure 1 shows a non-universal curve. This curve is a lower bound that gives the state-symbol pairs for which it is known that the halting problem is decidable [45]. The 1-symbol case is trivial, and the 1-state case was shown by Shannon [71] and, by using another method, Hermann [12]. Pavlotskaya [56] and, via another method, Kudlek [18] have shown that there are no universal 2-state, 2-symbol machines, where one transition rule is reserved for halting. Pavlotskaya [57] has also shown that there are no universal 3-state, 2-symbol machines, and also claimed [56], without proof, there are no universal machines for the 2-state, 3-symbol case. Again, both of these cases assume that a transition rule is reserved for halting.

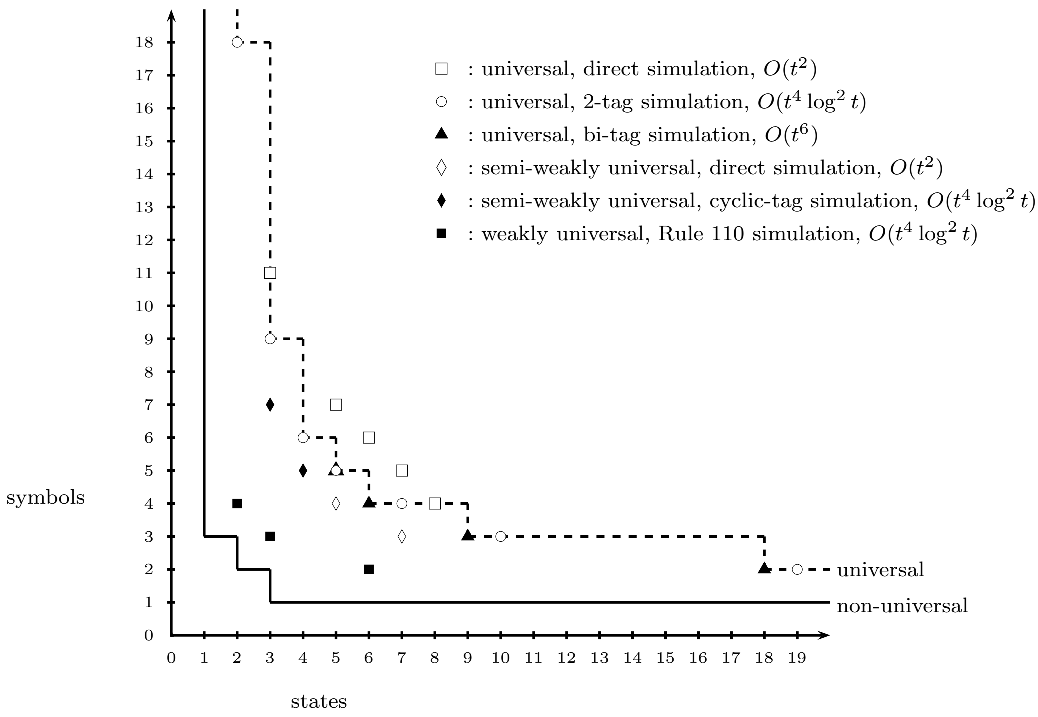

Fig. 1. State-symbol plot of small universal Turing machines. The type of simulation is given for each group of machines. Also we give the simulation overheads in terms of simulating a single tape, deterministic Turing machine that runs in time t.

2 Time and size efficiency of universal machines

As mentioned above, some of the very earliest small Turing machines were polynomial time simulators. Subsequently attention turned to the smaller, but exponentially slower, 2-tag simulators given by Minsky, Rogozhin and others.

Recently [49] we have given small machines that are efficient polynomial time simulators. More precisely, if M is a deterministic single-tape Turing machine that runs in time t, then there are machines, with state-symbol pairs given by the squares in Figure 1, that directly simulate M in polynomial time $O(t^2)$. These machines define a $O(t^2)$ curve. They are currently the smallest known universal Turing machines that simulate Turing machines in $O(t^2)$ time

Given these efficient $O(t^2)$ simulators it still remained the case that the smallest machines were exponentially slow. However we have recently shown [82] that 2-tag systems are in fact efficient simulators of Turing machines. More precisely, if M is a deterministic single-tape Turing machine that runs in time t then there is a 2-tag system that simulates M and runs in polynomial time $O(t^4 \log^2 t)$. The small machines of Minsky, Rogozhin, and others have a quadratic time overhead when simulating 2-tag systems, hence by the result in [82] they simulate Turing machines in time $O(t^8 \log^4 t)$. It turns out that the time overhead can be improved [45] to $O(t^4 \log^2 t)$, giving the $O(t^4 \log^2 t)$ machines in Figure 1. Thus, there is currently little evidence for the claim of an exponential trade-off between program size complexity, and time/space complexity.

From the point of view of program size, Neary and Woods [45,46,50] have recently given four Turing machines that are presently the smallest known (standard) machines with 2, 3, 4 and 5 symbols. The 5-symbol machine improves on the 5-symbol machine of Rogozhin [68] by one transition rule. The remainder of these machines improve on the 2- and 4-symbol machines of Baiocchi [4], and the 3-symbol machine of Rogozhin [68], by one state each. They simulate our universal variant of tag systems called *bi-tag systems* [47]. These small machines simulate Turing machines in polynomial time $O(t^6)$ and are illustrated as triangles in Figure 1. Bi-tag systems are essentially 1-tag systems (and so they read and delete one symbol per timestep) augmented with additional context sensitive rules that read, and delete, two symbols per timestep. On the one hand bi-tag systems are universal, while on the other hand they are sufficiently 'simple' to be simulated by such small machines.

Exponentially improving the time efficiency of 2-tag systems has implications for a number of models of computation, besides small universal Turing machines. Following our result, the simulation efficiency of many biologically inspired models of computation, including neural networks, H systems and P systems, has been improved from exponential to polynomial. For example, Siegelmann and Margenstern [72] give a neural network that uses only nine high-order neurons to simulate 2-tag systems. Taking each synchronous update of the nine neurons as a single parallel timestep, their neural network simulates 2-tag systems in linear time. They note that "tag systems suffer a significant slow-down ... and thus our result proves only Turing universality and should not be interpreted complexity-wise as a Turing equivalent." Our work shows that their neural network is in fact efficiently universal. Rogozhin and Verlan [70] give a tissue P system with eight rules that simulates 2-tag systems in linear time, and thus we have improved its simulation time overhead from exponential to polynomial. This system uses splicing rules (from H systems) with membranes (from P systems) and is non-deterministic. Harju and Margenstern [11] gave an extended H-system with 280 rules that generates recursively enumerable sets using Rogozhin's 7-state, 4-symbol universal Turing machine. Using our result from 2-tag systems, the time efficiency of their construction is improved from exponential to polynomial, with a possible small constant increase in the number of rules. The efficiency of Hooper's [14] small 2-tape universal Turing machine is also improved from exponential to polynomial. The technique of simulation via 2-tag systems is at the core of many of the universality proofs in Margenstern's survey [29]. Our work exponentially improves the time overheads in these simulations, such as Lindgren and Nordahl's cellular automata [20], Margenstern's non-erasing Turing machines [23,25], and Robinson's tiling [62].

3 Non-standard universal Turing machines; time efficiency and program size

So far we have been discussing results for universal Turing machines that have one tape, one tape head, and are deterministic (we often refer to this setup as the *standard* model). Of course one can consider results for other variants of the model. There are many generalised models, for example allowing multiple tapes, multiple dimensions, or even coupling the Turing machine with a finite automaton. Restricted models include non-erasing and reversible Turing machines, and machines with restricted instructions. In this section we explore program size and time complexity results for a number of generalised and restricted models. Table 2 contains program size results for a number of such non-standard machines.

3.1 Weak universality and Rule 110

An interesting generalisation occurs when we stick to the standard conventions, but we allow the blank portion of the tape to contain a word, that is constant (independent of the input), and is repeated infinitely often in one direction, say to the left of the input. We say that such Turing machines are *semi-weakly universal*. Some of the earliest small universal Turing machines were semi-weak [78,79]. Sometimes another word is also repeated infinitely often to the right. Universal machines that use this setup are called *weakly universal* [31].

It is not difficult to see how this generalisation can help to reduce program size. For example, it is typical of small universal Turing machine computations that the program being simulated is stored on the tape. When reading an instruction we often mark certain symbols. At a later time we then restore marked symbols to their original values. If the simulated program is repeated infinitely often, say to the left of the input, things may be much easier as we can simply skip the 'restore' phase of our algorithm and access a new copy of the program when simulating the next instruction, thus reducing the universal program's size.

This was the strategy used by Watanabe [78,79] to find the semi-weak, direct Turing machine simulators shown in Figure 1 as hollow diamonds. Recently [84] we have given two new semi-weakly universal machines and these are shown as solid diamonds in Figure 1. These machines simulate cyclic tag systems [7,81]. It is interesting to note that our machines are symmetric with those of Watanabe, despite the fact that we use a different simulation technique. Our 4-state, 5-symbol machine has only 17 transition rules, making it the smallest known semi-weakly universal machine (Watanabe's 5-state, 4-symbol machine has 18 transition rules). The time overhead for these machines is polynomial. More precisely, if M is a single-tape deterministic Turing machine that runs in time t, then M is simulated by either of our semi-weak machines in time $O(t^4 \log^2 t)$. Watanabe's semi-weak machines also ran in polynomial time, with a very efficient overhead of $O(t^2)$.

Cook, Eppstein, and Wolfram [7,81] gave weakly universal Turing machines that were significantly smaller than the existing semi-weak machines. These were improved upon by Neary and Woods [51] to give the smallest known weakly universal machines. In (states, symbols) notation their sizes are $(2,4)$, $(3,3)$ and $(6,2)$, and they are illustrated in Figure 1. These machines work by simulating Rule 110, a very simple kind of cellular automaton. Rule 110 is an elementary cellular automaton, which means that it is a one-dimensional, nearest neighbour, binary cellular automaton [80]. More precisely, it is composed of a sequence of cells $\ldots p_{-1}p_0p_1\ldots$ where each cell has a binary state $p_i \in \{0,1\}$. At timestep $t+1$ the value of cell $p_{i,t+1} = F(p_{i-1,t}, p_{i,t}, p_{i+1,t})$ is given by the synchronous local update function F

$$\begin{array}{ll} F(0,0,0)=0 & F(1,0,0)=0 \\ F(0,0,1)=1 & F(1,0,1)=1 \\ F(0,1,0)=1 & F(1,1,0)=1 \\ F(0,1,1)=1 & F(1,1,1)=0 \end{array}$$

Rule 110 was shown to be universal via an impressive and detailed simulation of cyclic tag systems, the result is stated and described in [81] and the full proof is given in [7]. In the proof, the Rule 110 instance has a special (constant) word repeated infinitely to the left of the input, and another to the right. Rule 110 has a very simple update rule which facilitates the writing of very small weak Turing machines that simulate it.

As noted, Rule 110 was shown to be universal by simulating cyclic tag systems, which in turn simulate 2-tag systems. The chain of simulations included the exponentially slow 2-tag algorithm of Cocke and Minsky, thus Rule 110, and the weakly universal machines that simulate it, were exponentially slow. In a recent paper [48] we have improved their simulation time overhead to polynomial by showing that cyclic tag systems are efficient simulators of Turing machines. This result has interesting implications for Rule 110. For example, given an initial configuration of Rule 110, and a value t in unary, predicting t timesteps of a Rule 110 computation is P-complete. Therefore, unless P = NC, which is widely believed to be false, we cannot hope to quickly (in polylogarithmic time) predict the evolution of this simple cellular automaton even if we have a polynomial amount of parallel hardware. Rule 110 is the simplest (one-dimensional, nearest neighbour) cellular automaton that has been shown to have a P-complete prediction problem. In particular Ollinger's [53] intrinsic universality result already shows that prediction for one dimensional nearest neighbour cellular automata is P-complete for 6 states (later improved to 4 states by Richard [61]), and our result improves this to 2 states. The question of whether Rule 110 prediction is P-complete has been, directly and indirectly, asked in a number of previous works (for example [2,40,41]).

It is currently unknown whether all of the lower bounds in Figure 1 hold for weak machines. For example, the non-universality results of Pavlotskaya were proven for the case where one transition rule is reserved for halting, however the smallest weak machines do not halt.

3.2 Other non-standard universal Turing machines

Weakness has not been the only generalisation on the standard model in the search for ever smaller universal machines. We give some notable examples here, many others are to be found in Table 2.

Before Shannon's famous paper, Moore [42] observed that 2-symbol machines were universal as any Turing machine could be converted into a 2-symbol machine by the (now) usual encoding. In the same paper Moore used this observation to give a universal 3-tape machine with 15 states and 2 symbols. Moore's machine uses only 27 instructions, each instruction being a sextuple that either moves one of its tape heads or prints a single symbol to one of its tapes. One of the tapes in Moore's 3-tape machine is circular and contains the simulated program, therefore his machine also operates correctly if the circular tape is replaced with a semi-weak tape. Moore's result has been largely ignored in the literature despite being the first published small universal Turing machine. Interestingly, Moore's paper cites unpublished work by Shannon on the universality of non-erasing machines.

Hooper [13,14] gave universal machines with 2 states, 3 symbols and 2 tapes, and with 1 state, 2 symbols and 4 tapes. One of the tapes in Hooper's 4-tape machine is circular and contains the simulated program, and so could be replaced by a semi-weak tape. Priese [60] gave a 2-state, 4-symbol machine with a 2-dimensional tape, and a 2-state, 2-symbol machine with a pair of 2-dimensional tapes. Margenstern and Pavlotskaya [32,33] gave a 2-state, 3-symbol Turing machine that uses only 5 instructions and is universal when coupled with a finite automaton. They also showed that the halting problem is decidable for such machines with 4 instructions [33].

3.3 Restricted universal Turing machines

If we restrict the standard Turing machine model the problem of finding very small universal machines becomes more difficult. Over the years, a number of authors have looked at non-erasing Turing machines which are permitted to overwrite blank symbols only. Moore [42] mentions that Shannon had proved that such non-erasing Turing machines simulate arbitrary Turing machines, however Shannon's work was never published. Shortly after, Shannon proved that 2-symbol Turing machines are universal, and Wang [75] proved that 2-symbol non-erasing Turing machines are universal. Later, Minsky proved the same result as Wang, but using the technique of simulation via non-writing Turing machines, yet another (universal) restriction [38]. More recently, Margenstern [22–26,30] has constructed a number of small non-erasing universal machines with further restrictions.

Fischer [9] gives a number of universality results for Turing machines that use restricted forms of transition rules. In one result he proves that 3-state Post machines are universal (Post machines are like Turing machines, but they can not write and move in the same timestep). Interestingly, Aanderaa and Fischer [1] show that the halting problem for 2-state Post machines is decidable.

Bennett [5] has shown that 3-tape reversible Turing machines are universal. Morita and others have since shown universality results for reversible Turing machines with 1 tape and 2 symbols [43], and 17 states and 5 symbols [44].

4 Further work

There are many avenues for further work in this area, here we highlight a few examples.

Applying computational complexity theory to the area of small universal Turing machines allows us to ask a number of questions that are more subtle than the usual questions about program size. As we move towards the origin in Figure 1, the universal machines have larger (but polynomial) time overheads. Can the time overheads in Figure 1 be further improved (lowered)? Can we prove lower bounds on the simulation time of machines with a given state-symbol pair? Proving non-trivial simulation time lower bounds seems like a difficult problem. Such results could be used to prove that there is a polynomial trade-off between simulation time and universal program size.

As we move away from the origin, the non-universal machines seem to have more power. For example Kudlek's classification of 2-state, 2-symbol machines shows that the sets accepted by these machines are regular, with the exception of one context free language ($a^n b^n$). Can we hope to fully characterise the sets accepted by non-universal machines (e.g. in terms of complexity or automata theoretic classes) with given state-symbol pairs or other program restrictions?

When discussing the complexity of small machines the issue of encodings becomes very important. For example, when proving that the prediction problem for a small machine is P-complete [10], the relevant encodings should be in logspace, and this is the case for all of the polynomial time machines in Figure 1.

Of course there are many models of computation that we have not mentioned where researchers have focused on finding small universal programs. Post's [59] tag systems are an interesting example. Minsky [37,38] showed that tag systems are universal with deletion number 6. Cocke and Minsky lowered the deletion number to 2, by showing that 2-tag systems were universal. They used productions (appendants) of length at most 4. Wang [76] further lowered the production length to 3. Recently, De Mol [8] has given a lowerbound by showing that the reachability (and thus halting) problems are decidable for 2-tag systems with production length 2. It would be interesting to find the smallest universal tag systems in terms of number of symbols, deletion length, and production length.

The space between the non-universal curve and the smallest non-weakly universal machines in Figure 1 contains some complicated beasts. These lend weight to the feeling that finding new lower bounds on universal program size is tricky. Most noteworthy are the weakly and semi-weakly universal machines discussed above. Table 2 highlights that the existence of general models that have provably less states and symbols than the standard universal machines (for example the machines with (state, symbol, dimensions, tapes) of (2,2,2,2) [60], (1,7,3,1) [17], and (1,2,1,4) [14]). Also of importance are the small machines of Margenstern [28,29], Baiocchi [3], and Michel [34,35] that live in this region and simulate iterations of the $3x + 1$ problem. So it seems that there are plenty of animals yet to be tamed.

Acknowledgements

We thank Olivier Bournez, Paola Bonizzoni and the organisers of *Computability in Europe (CiE) 2007* for the invitation to this special issue. An earlier version [83] of this work was presented at *CiE 2007*.

References

[1] S. Aanderaa and P. C. Fischer. The solvability of the halting problem for 2-state post machines. *Journal of the Association for Computing Machinery*, 14(4):677–682, Oct. 1967.

[2] S. Aaronson. Book review: A new kind of science. *Quantum Information and Computation*, 2(5):410–423, 2002.

[3] C. Baiocchi. 3N+1, UTM e tag-system. Technical Report Pubblicazione 98/38, Dipartimento di Matematico, Università di Roma, 1998. (In Italian).

[4] C. Baiocchi. Three small universal Turing machines. In M. Margenstern and Y. Rogozhin, editors, *Machines, Computations, and Universality*, volume 2055 of *Lecture Notes in Computer Science*, pages 1–10, Chişinău, Moldova, May 2001. MCU, Springer.

[5] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, 1973.

[6] J. Cocke and M. Minsky. Universality of tag systems with $P = 2$. *Journal of the Association for Computing Machinery*, 11(1):15–20, Jan. 1964.

[7] M. Cook. Universality in elementary cellular automata. *Complex Systems*, 15(1):1–40, 2004.

[8] L. De Mol. Study of limits of solvability in tag systems. In J. Durand-Lose and M. Margenstern, editors, *Machines, Computations and Universality (MCU)*, volume 4664 of *LNCS*, pages 170–181, Orléans, France, Sept. 2007. Springer.

[9] P. C. Fischer. On formalisms for Turing machines. *Journal of the Association for Computing Machinery*, 12(4):570–580, Oct. 1965.

[10] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to parallel computation: P-completeness theory.* Oxford university Press, Oxford, 1995.

[11] T. Harju and M. Margenstern. Splicing systems for universal Turing machines. *DNA Computing: Lecture Notes in Computer Science*, 3384:149–158, June 2005.

[12] G. T. Hermann. The uniform halting problem for generalized one state Turing machines. In *Proceedings of the ninth annual Symposium on Switching and Automata Theory (FOCS)*, pages 368–372, Schenectady, New York, Oct. 1968. IEEE Computer Society Press.

[13] P. Hooper. Some small, multitape universal Turing machines. Technical report, Computation Labortory, Harvard University, Cambridge, Massachusetts, 1963.

[14] P. Hooper. Some small, multitape universal Turing machines. *Information Sciences*, 1(2):205–215, 1969.

[15] N. Ikeno. A 6-symbol 10-state universal Turing machine. In *Proceedings, Institute of Electrical Communications*, Tokyo, 1958.

[16] H. Kleine-Büning. *Über probleme bei homogener Parkettierung von $\mathbb{Z} \times \mathbb{Z}$ durch Mealy-automaten bei normierter verwendung*. PhD thesis, Institut für Mathematische Logik, Münster, 1977.

[17] H. Kleine-Büning and T. Ottmann. Kleine universelle mehrdimensionale Turingmaschinen. *Elektronische Informationsverarbeitung und Kybernetik*, 13(4-5):179–201, 1977. (In German.).

[18] M. Kudlek. Small deterministic Turing machines. *Theoretical Computer Science*, 168(2):241–255, 1996.

[19] M. Kudlek and Y. Rogozhin. A universal Turing machine with 3 states and 9 symbols. In W. Kuich, G. Rozenberg, and A. Salomaa, editors, *Developments in Language Theory (DLT) 2001*, volume 2295 of *LNCS*, pages 311–318, Vienna, May 2002. Springer.

[20] K. Lindgren and M. G. Nordahl. Universal computation in simple one-dimensional cellular automata. *Complex Systems*, 4(3):299–318, 1990.

[21] M. Margenstern. Suprising areas in the quest for small universal devices. *Electronic Notes in Theoretical computer Science*. (To appear.).

[22] M. Margenstern. Sur la frontière entre machines de Turing á arrêt décidable et machines de Turing universelles. Technical Report 92-83, LITP Institut Blaise Pascal, 1992.

[23] M. Margenstern. Non-erasing Turing machines: A frontier between a decidable halting problem and universality. In *Fundamentals of Computation Theory (FCT)*, volume 710 of *LNCS*, pages 375–385, Szeged, Hungry, May 1993. Springer.

[24] M. Margenstern. Une machine de Turing universelle sur {0,1}, non-effaçante et à trois instructions gauches. Technical Report 94-08, LITP Institut Blaise Pascal, 1994.

[25] M. Margenstern. Non-erasing Turing machines: A new frontier between a decidable halting problem and universality. In *LATIN*, volume 911 of *LNCS*, pages 386–397, Valparaíso, Chile, Apr. 1995. Springer.

[26] M. Margenstern. Une machine de Turing universelle non-effaçante à exactement trois instructions gauches. *CRAS, Paris*, 320(I):101–106, 1995.

[27] M. Margenstern. Decidability and undecidability of the halting problem on Turing machines, a survey. In S. Adian and A. Nerode, editors, *Logical Foundations of Computer Science (LFCS)*, volume 1234 of *LNCS*, pages 226–236, Yaroslav, Russia, July 1997. Springer.

[28] M. Margenstern. Frontier between decidablity and undecidablity: a survey. In M. Margenstern, editor, *Machines, Computations, and Universality (MCU) volume 1*, pages 141–177, France, 1998. IUT, Metz.

[29] M. Margenstern. Frontier between decidability and undecidability: a survey. *Theoretical Computer Science*, 231(2):217–251, Jan. 2000.

[30] M. Margenstern. On quasi-unilateral universal Turing machines. *Theoretical Computer Science*, 257(1–2):153–166, Apr. 2001.

[31] M. Margenstern. An algorithm for building intrinsically universal cellular automata in hyperbolic spaces. In *Proceedings of the 2006 International Conference on Foundations of Computer Science (FCS)*, pages 3–9, Las Vegas, NV, June 2006. CSREA Press.

[32] M. Margenstern and L. Pavlotskaya. Vers ue nouvelle approche de l'universalité concernant les machines de Turing. Technical Report 95-58, LITP Institut Blaise Pascal, 1995.

[33] M. Margenstern and L. Pavlotskaya. On the optimal number of instructions for universality of Turing machines connected with a finite automaton. *International Journal of Algebra and Computation*, 13(2):133–202, Apr. 2003.

[34] P. Michel. Busy beaver competition and Collatz-like problems. *Archive Mathematical Logic*, 32(5):351–367, 1993.

[35] P. Michel. Small Turing machines and generalized busy beaver competition. *Theoretical Computer Science*, 326:45–56, Oct. 2004.

[36] M. Minsky. A 6-symbol 7-state universal Turing machines. Technical Report 54-G-027, MIT, Aug. 1960.

[37] M. Minsky. Recursive unsolvability of Post's tag problem. Technical Report 54-G-023, Massachusetts Institute of Technology, June 1960.

[38] M. Minsky. Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines. *Annals of Mathematics*, 74(3):437–455, Nov. 1961.

[39] M. Minsky. Size and structure of universal Turing machines using tag systems. In *Recursive Function Theory: Proceedings, Symposium in Pure Mathematics*, volume 5, pages 229–238, Provelence, 1962. AMS.

[40] C. Moore. Quasi-linear cellular automata. *Physica D*, 103:100–132, 1997.

[41] C. Moore. Predicting non-linear cellular automata quickly by decomposing them into linear ones. *Physica D*, 111:27–41, 1998.

[42] E. F. Moore. A simpilfied universal Turing machine. In *ACM national meeting*, pages 50–54, Toronto, Canada, 1952. ACM Press.

[43] K. Morita, A. Shirasaki, and Y. Gono. A 1-tape 2-symbol reversible Turing machine. *The Transactions of the IEICE Japan*, E72(3):223–228, Mar. 1989.

[44] K. Morita and Y. Yamaguchi. A universal reversible Turing machine. In J. Durand-Lose and M. Margenstern, editors, *Machines, Computations and Universality (MCU)*, volume 4664 of *LNCS*, pages 90–98, Orléans, France, Sept. 2007. Springer.

[45] T. Neary. *Small universal Turing machines.* PhD thesis, National University of Ireland, Maynooth. In submission.

[46] T. Neary. Small polynomial time universal Turing machines. In *Fourth Irish Conference on the Mathematical Foundations of Computer Science and Information Technology (MFCSIT'06)*, pages 325–329, Ireland, 2006. University College Cork.

[47] T. Neary and D. Woods. A small fast universal Turing machine. Technical Report NUIM-CS-TR-2005-12, Department of Computer Science, NUI Maynooth, 2005.

[48] T. Neary and D. Woods. P-completeness of cellular automaton Rule 110. In M. Bugliesi et al., editor, *International Colloquium on Automata Languages and Programming (ICALP)*, volume 4051 (Part I) of *Lecture Notes in Computer Science*, pages 132–143. Springer, July 2006.

[49] T. Neary and D. Woods. Small fast universal Turing machines. *Theoretical Computer Science*, 362(1–3):171–195, Oct. 2006.

[50] T. Neary and D. Woods. Four small universal Turing machines. In J. Durand-Lose and M. Margenstern, editors, *Machines, Computations and Universality (MCU)*, volume 4664 of *LNCS*, pages 242–254, Orléans, France, Sept. 2007. Springer.

[51] T. Neary and D. Woods. Small weakly universal Turing machines, July 2007. arXiv:0707.4489v1 [cs.CC].

[52] A. Nozaki. On the notion of universality of Turing machine. *Kybernetika Academia Praha*, 5(1):29–43, 1969. (English translated version).

[53] N. Ollinger. The quest for small universal cellular automata. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2380 of *LNCS*, pages 318–329, Malaga, Spain, July 2002. Springer.

[54] T. Ottmann. Eine universelle Turingmaschine mit zweidimensionalem band. *Elektronische Informationsverarbeitung und Kybernetik*, 11(1-2):27–38, 1975. (In German).

[55] T. Ottmann. Einfache universelle mehrdimensionale Turingmaschinen. Habilitationsschrift, Karlsruhe, 1975.

[56] L. Pavlotskaya. Solvability of the halting problem for certain classes of Turing machines. *Mathematical Notes (Springer)*, 13(6):537–541, June 1973. (Translated from Matematicheskie Zametki, Vol. 13, No. 6, pp. 899–909, June, 1973).

[57] L. Pavlotskaya. Dostatochnye uslovija razreshimosti problemy ostanovki dlja mashin T'juring. *Avtomaty i Mashiny*, pages 91–118, 1978. (Sufficient conditions for the halting problem decidability of Turing machines) (in Russian).

[58] L. Pavlotskaya. On machines, universal by extensions. *Theoretical Computer Science*, 168(2):257–266, Nov. 1996.

[59] E. Post. Formal reductions of the general combinatorial decision problem. *American Journal of Mathmatics*, 65(2):197–215, Apr. 1943.

[60] L. Priese. Towards a precise characterization of the complexity of universal and nonuniversal Turing machines. *SIAM J. Comput.*, 8(4):508–523, 1979.

[61] G. Richard. A particular universal cellular automaton, Sept. 2006. HAL research report (oai:hal.archives-ouvertes.fr:hal-00095821_v1).

[62] R. M. Robinson. Undecidability and nonperiodicity for tilings of the plane. *Inventiones Mathematicae*, 12(3):177–209, 1971.

[63] R. M. Robinson. Minsky's small universal Turing machine. *International Journal of Mathematics*, 2(5):551–562, 1991.

[64] Y. Rogozhin. Sem' universal'nykh mashin T'juringa. In *Fifth all union conferece on Mathematical Logic, Akad. Naul SSSR. Otdel. Inst. Mat., Novosibirsk*, page 27, 1979. (Seven universal Turing machines. In Russian).

[65] Y. Rogozhin. Sem' universal'nykh mashin T'juringa. *Systems and theoretical programming, Mat. Issled.*, 69:76–90, 1982. (Seven universal Turing machines. In Russian).

[66] Y. Rogozhin. Universal'naja mashina T'juringa s 10 sostojanijami i 3 simvolami. *Izvestiya Akademii Nauk Respubliki Moldova, Matematika*, 4(10):80–82, 1992. (A universal Turing machine with 10 states and 3 symbols. In Russian).

[67] Y. Rogozhin. About Shannon's problem for Turing machines. *Computer Science Journal of Moldova*, 1(3):108–111, 1993.

[68] Y. Rogozhin. Small universal Turing machines. *Theoretical Computer Science*, 168(2):215–240, Nov. 1996.

[69] Y. Rogozhin. A universal Turing machine with 22 states and 2 symbols. *Romanian Journal of Information Science and Technology*, 1(3):259–265, 1998. (In Russian).

[70] Y. Rogozhin and S. Verlan. On the rule complexity of universal tissue P systems. In R. Freund et al., editor, *Sixth international Workshop on Membrane Computing*, volume 3850 of *LNCS*, pages 356–362, Vienna, July 2005. Springer.

[71] C. E. Shannon. A universal Turing machine with two internal states. *Automata Studies, Annals of Mathematics Studies*, 34:157–165, 1956.

[72] H. T. Siegelmann and M. Margenstern. Nine switch-affine neurons suffice for Turing universality. *Neural Networks*, 12(4–5):593–600, Feb. 1999.

[73] H. Takahashi. Keisankikai II. *Iwanami, Tokyo*, 1958. (Computing machine II. In Japanese.).

[74] K. Wagner. Universelle Turingmaschinen mit n-dimensionale band. *Elektronische Informationsverarbeitung und Kybernetik*, 9(7-8):423–431, 1973. (Universal Turing machines with n-dimensional tapes. In German).

[75] H. Wang. A variant to Turing's theory of computing machines. *Journal of the Association for Computing Machinery*, 4(1):63–92, Jan. 1957.

[76] H. Wang. Tag sytems and lag systems. *Mathematical Annals*, 152(4):65–74, Oct. 1963.

[77] S. Watanabe. On a minimal universal Turing machine. Technical report, MCB Report, Tokyo, Aug. 1960.

[78] S. Watanabe. 5-symbol 8-state and 5-symbol 6-state universal Turing machines. *Journal of the ACM*, 8(4):476–483, Oct. 1961.

[79] S. Watanabe. 4-symbol 5-state universal Turing machine. *Information Processing Society of Japan Magazine*, 13(9):588–592, 1972.

[80] S. Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601–644, July 1983.

[81] S. Wolfram. *A new kind of science*. Wolfram Media, Inc., 2002.

[82] D. Woods and T. Neary. On the time complexity of 2-tag systems and small universal Turing machines. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 439–446, Berkeley, California, Oct. 2006. IEEE.

[83] D. Woods and T. Neary. The complexity of small universal Turing machines. In *Computation and Logic in the Real World: Third Conference of Computability in Europe, CiE 2007*, volume 4497 of *LNCS*, pages 791–798, Siena, Italy, June 2007. Springer. Invited.

[84] D. Woods and T. Neary. Small semi-weakly universal Turing machines. In J. Durand-Lose and M. Margenstern, editors, *Machines, Computations and Universality (MCU)*, volume 4664 of *LNCS*, pages 303–315, Orléans, France, Sept. 2007. Springer.

states	symbols	state-symbol product	author
m	2	2m	Shannon [71]
2	n	2n	Shannon [71]
12	6	72	Takahashi [73] (mentioned in [78])
10	6	60	Ikeno [15] (also appears in [36])
8	6	48	Watanabe [77] (mentioned in [39])
7	6	42	Minsky [36]
8	5	40	Watanabe [78]
6	6	36	Minsky [39]
7	4	28	Minsky [39]
24	2	48	Rogozhin [64,65,68]
2	21	42	Rogozhin [64,65]
11	3	33	Rogozhin [64,65]
3	10	30	Rogozhin [64,65]
7	4	28	Rogozhin [64,65,68]
5	5	25	Rogozhin [64,65,68]
4	6	24	Rogozhin [64,65,68]
2	18	36	Rogozhin [68]
10	3	30	Rogozhin [66,68]
3	10	30	Rogozhin [67,68]*
22	2	44	Rogozhin [69]
19	2	38	Baiocchi [4]
7	4	28	Baiocchi [4]*
3	9	27	Kudlek & Rogozhin [19]
5	5	25	Neary & Woods [50]*
6	4	24	Neary & Woods [50]
9	3	27	Neary & Woods [50]
18	2	36	Neary & Woods [50]

Table 1
Small standard universal Turing machines. If there are multiple machines with the same state-symbol pair, the machine with the smallest number of instructions is denoted *.

states	symbols	dimensions	tape	author
15	2	1	3	Moore [42]†
6	5	1	1	Watanabe [78]†
1	2	1	4	Hooper [13,14]†
2	3	1	2	Hooper [13,14]
7	3	1	1	Watanabe (mentioned in [79,52])†
5	4	1	1	Watanabe [79]†
8	4	2	1	Wagner [74]
2	7	2	1	Ottmann [54]‡
10	2	2	1	Ottmann [55,17]‡
6	3	2	1	Ottmann [55,17]‡
4	4	2	1	Ottmann [55,17]‡
2	6	2	1	Kleine-Büning & Ottmann [17]‡
1	7	3	1	Kleine-Büning & Ottmann [17]‡
2	5	2	1	Kleine-Büning & Ottmann [17]‡
2	3	2	1	Kleine-Büning & Ottmann [17]‡
4	5	2	1	Kleine-Büning & Ottmann [17]
3	6	2	1	Kleine-Büning & Ottmann [17]
10	2	2	1	Kleine-Büning [16]
2	5	2	1	Kleine-Büning [16]
2	4	2	1	Priese [60]
2	2	2	2	Priese [60]
4	7	1	1	Pavlotskaya [58]⋆
2	5	1	1	Margenstern & Pavlotskaya [32]⋆
2	3	1	1	Margenstern & Pavlotskaya [33]⋆
4	3	1	1	Cook [7] & Wolfram [81]‡
3	4	1	1	Cook [7] & Wolfram [81]‡
2	5	1	1	Cook [7] & Wolfram [81]‡
7	2	1	1	Eppstein (published by Cook [7])‡
3	7	1	1	Woods & Neary [84]†
4	5	1	1	Woods & Neary [84]†
6	2	1	1	Neary & Woods [51]‡
3	3	1	1	Neary & Woods [51]‡
2	4	1	1	Neary & Woods [51]‡

Table 2
Small non-standard universal Turing machines. Semi-weak machines are denoted by †, weak machines by ‡, and machines coupled with a finite automaton by ⋆.