

Tag Systems and the Complexity of Simple Programs

Turlough Neary¹(✉) and Damien Woods²

¹ Institute of Neuroinformatics, University of Zürich and ETH Zürich,
Zürich, Switzerland

`tneary@ini.phys.ethz.ch`

² Division of Engineering and Applied Science,
California Institute of Technology, Pasadena, CA 91125, USA
`woods@caltech.edu`

Abstract. In this mini-survey we discuss time complexity and program size results for universal Turing machines, tag systems, cellular automata, and other simple models of computation. We discuss results that show that many of the simplest known models of computation including the smallest known universal Turing machines and the elementary cellular automaton Rule 110 are efficient simulators of Turing machines. We also recall a recent result where the halting problem for tag systems with only 2 symbols (the minimum possible) is proved undecidable. This result has already yielded applications including a significant improvement on previous undecidability bounds for the Post correspondence problem and the matrix mortality problem.

1 Introduction

This brief survey is concerned with time complexity and program size results for universal Turing machines, tag systems, cellular automata and other simple models of computation. We pay particular attention to tag systems as they are at the center of many of the results we discuss. Here we provide only a brief glimpse at the above mentioned topics and we direct the reader who wishes to learn more to other related surveys [17, 27].

2 Program Size of Small Universal Machines

In 1956 Shannon [35] considered the question of finding the smallest possible universal Turing machine, where size is the number of states and symbols. Figure 1 summarises the state of the art for the smallest known standard, weakly and semi-weakly universal Turing machines. Here we say a machine is standard if it

Turlough Neary is supported by Swiss National Science Foundation grants 200021-141029 and 200021-153295. Damien Woods is supported by NASA grant NNX13AJ56G, and National Science Foundation grants 0832824 & 1317694 (The Molecular Programming Project), CCF-1219274 and CCF-1162589.

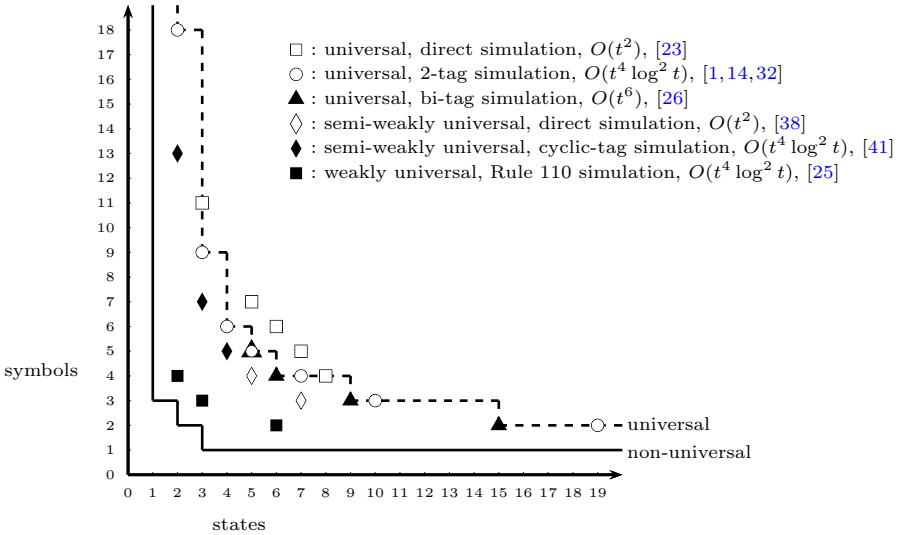


Fig. 1. State-symbol plot of small universal Turing machines. The type of simulation is given for each group of machines. For example, each machine plotted as a hollow circle simulates 2-tag systems, a technique introduced by Minsky [20]. The technique of Rule 110 simulation was devised by Cook [4] and later improved upon in [26] to give the weakly universal machines plotted as solid squares. Direct simulation indicates that a machine was proved universal by simulating Turing machines directly rather than via some other system. Simulation time overheads are given in terms of simulating a single-tape deterministic Turing machine that runs in time t .

is deterministic and has a single-tape. Semi-weakly universal machines generalise the standard model by allowing an infinitely repeated word on one side of the input, and the (standard) infinitely repeated blank symbol on the other. Weakly universal machines are a further generalisation on the standard model as they allow an infinitely repeated word to the left of the input, and another infinitely repeated word to the right. It is often the case that generalising the model allows us to find smaller universal programs. This notion is borne out when we compare the standard, weak and semi-weak machines in Figure 1.

In Figure 1 the machines with the state-symbol pairs (2, 18), (3, 9), (4, 6), (5, 5), (6, 4), (9, 3), and (15, 2) are the smallest known standard machines and these machines define the universal curve (dashed line). Figure 1 also gives 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 [11, 13, 28, 29]. It is currently unknown whether all of the lower bounds in Figure 1 hold for weak and semi-weak machines. For example, the non-universality results of Pavlotskaya [28, 29] were proven under the assumption that the (standard) blank symbol is infinitely repeated to the left and right of the input.

3 Time Efficiency of Small Universal Machines

Cocke and Minsky [3] proved 2-tag systems universal via an exponentially slow simulation of Turing machines. So for many years the smallest known universal Turing machines were also exponentially slow. However, following the introduction of a new efficient 2-tag system algorithm for simulating Turing machines [40], the smallest known universal machines were found to be efficient simulators of Turing machines. To be more exact, given a single tape Turing machine M that runs in time t the small machines given by hollow circles in Figure 1 simulate M in time $O(t^8 \log^4 t)$ (this overhead was later improved [21] to $O(t^4 \log^2 t)$). Another consequence of this result is that many other systems [10, 12, 15, 16, 33, 34, 36] that simulate 2-tag systems, either directly or via a chain of simulations, are polynomial (instead of exponential) time simulators of Turing machines.

Rule 110 was proved universal by Matthew Cook via an impressive and intricate simulation of cyclic tag systems, the result is described in [39] and a full proof is given in [4]. Unfortunately, cyclic tag systems simulated Turing machines via Cocke and Minsky’s exponentially slow 2-tag algorithm and so Rule 110 and the small weakly universal machines mentioned earlier were exponentially slow simulator of Turing machines. However, in [22] it was shown that cyclic tag systems simulate Turing machines in polynomial time. As a result, the following problem is now P-complete for Rule 110: Given a number t in unary, an initial configuration of Rule 110 and a cell c_i , predict the state of c_i after t timesteps. Rule 110 is the simplest (one-dimensional, nearest neighbour) cellular automaton that has been shown to have a P-complete prediction problem.

4 Universality of Binary Tag Systems

Tag systems are a type of rewriting system that use a very simple form of rule. A tag system acts on a dataword, which is a string of symbols taken from a finite alphabet Σ . There is a fixed set of rules $R : \Sigma \rightarrow \Sigma^*$ and a deletion number $\beta \in \mathbb{N}$. In a single timestep, the leftmost symbol σ_j of the dataword is read, if there is a rule $\sigma_j \rightarrow \alpha_j$ then the string α_j is appended to the right of the dataword and the leftmost β symbols are deleted. As an example we give the first 4 steps of Post’s [31] binary tag system with deletion number 3 and the rules $0 \rightarrow 00$ and $1 \rightarrow 1101$ on the input 0101110.

0101110 † 111000 † 0001101 † 110100 † 1001101 † ...

A tag system computation halts if its dataword is shorter than its deletion number. Surprisingly, the halting problem for the simple tag system given above (which Post discovered in the 1920s) remains open to this day [6]. Another remarkably simple tag system with an open halting problem was found by De Mol [5] when she reduced the well known Collatz problem to the halting problem for a tag system with only 3 rules and deletion number 2.

Tag systems were introduced by Post [30,31] and proved universal by Minsky [19]. Soon after this Cocke and Minsky [3] proved 2-tag systems (tag systems with deletion number 2) universal. As mentioned earlier, 2-tag systems have been used to prove universality for many of the smallest known universal Turing machines [1,14,20,32] and are central to many other universality results [10,12,15,16,33,34,36]. Given that tag systems have been so useful in the search for new simple universal systems, it is surprising that there have been no attempts to simplify tag systems since the 1960s. All known universal tag systems [3,4,37] have a large number of symbols and thus a large number of rules. Recently [24] it was shown that tag systems with only 2 symbols (the minimum possible) are universal by showing that they simulate cyclic tag systems. Applications have already been found for this result. The undecidable halting problem for binary tag systems reduces to the Post correspondence problem for 5 pairs of words. The previous bound for undecidability in this problem, which is due to Matiyasevich and Sénizergues [18], was 7 pairs. Following this new result, only the cases for 3 and 4 pairs of words remain open, as the problem is known to be decidable for 2 pairs [7]. Applying the reductions of Halava and Harju [8], and Cassaigne and Karhumäki [2] to the Post correspondence problem for 5 pairs of words shows that the matrix mortality problem is undecidable for sets with six 3×3 matrices and for sets with two 18×18 matrices. The previous bounds for the undecidability in this problem was seven 3×3 matrices and two 21×21 matrices [9].

Looking to the future, we expect that binary tag systems will have an important role in proving further undecidability results and in the search for new simple models of computation.

References

1. Baiocchi, C.: Three small universal turing machines. In: Margenstern, M., Rogozhin, Y. (eds.) MCU 2001. LNCS, vol. 2055, pp. 1–10. Springer, Heidelberg (2001)
2. Cassaigne, J., Karhumäki, J.: Examples of undecidable problems for 2-generator matrix semigroups. *Theoretical Computer Science* **204**(1–2), 29–34 (1998)
3. Cocke, J., Minsky, M.: Universality of tag systems with $P = 2$. *Journal of the Association for Computing Machinery* **11**(1), 15–20 (1964)
4. Cook, M.: Universality in elementary cellular automata. *Complex Systems* **15**(1), 1–40 (2004)
5. De Mol, L.: Tag systems and Collatz-like functions. *Theoretical Computer Science* **390**(1), 92–101 (2008)
6. De Mol, L.: On the complex behavior of simple tag systems - an experimental approach. *Theoretical Computer Science* **412**(1–2), 97–112 (2011)
7. Ehrenfeucht, A., Karhumäki, J., Rozenberg, G.: The (generalized) Post correspondence problem with lists consisting of two words is decidable. *Theoretical Computer Science* **21**(2), 119–144 (1982)
8. Halava, V., Harju, T.: Mortality in matrix semigroups. *American Mathematical Monthly* **108**(7), 649–653 (2001)

9. Halava, V., Harju, T., Hirvensalo, M.: Undecidability bounds for integer matrices using Claus instances. *International Journal of Foundations of Computer Science* **18**(5), 931–948 (2007)
10. Harju, T., Margenstern, M.: Splicing systems for universal Turing machines. In: Ferretti, C., Mauri, G., Zandron, C. (eds.) *DNA 2004*. LNCS, vol. 3384, pp. 149–158. Springer, Heidelberg (2005)
11. Hermann, G.T.: The uniform halting problem for generalized one state Turing machines. In: *Proceedings of the Ninth Annual Symposium on Switching and Automata Theory (FOCS)*, pp. 368–372, IEEE Computer Society Press, Schenectady, New York, Oct. 1968
12. Hooper, P.: Some small, multitape universal Turing machines. *Information Sciences* **1**(2), 205–215 (1969)
13. Kudlek, M.: Small deterministic Turing machines. *Theoretical Computer Science* **168**(2), 241–255 (1996)
14. Kudlek, M., Rogozhin, Y.: A universal turing machine with 3 states and 9 symbols. In: Kuich, W., Rozenberg, G., Salomaa, A. (eds.) *DLT 2001*. LNCS, vol. 2295, pp. 149–158. Springer, Heidelberg (2002)
15. Lindgren, K., Nordahl, M.G.: Universal computation in simple one-dimensional cellular automata. *Complex Systems* **4**(3), 299–318 (1990)
16. Margenstern, M.: Non-erasing Turing machines: A new frontier between a decidable halting problem and universality. In: Baeza-Yates, R.A., Poblete, P.V., Goles, E. (eds.) *LATIN*. LNCS, vol. 911, pp. 386–397. Springer, Heidelberg (1995)
17. Margenstern, M.: Frontier between decidability and undecidability: a survey. *Theoretical Computer Science* **231**(2), 217–251 (2000)
18. Matiyasevich, Y., Sénizergues, G.: Decision problems for semi-Thue systems with a few rules. *Theoretical Computer Science* **330**(1), 145–169 (2005)
19. Minsky, M.: Recursive unsolvability of Post’s problem of “tag” and other topics in theory of Turing machines. *Annals of Mathematics* **74**(3), 437–455 (1961)
20. Minsky, M.: Size and structure of universal Turing machines using tag systems. In: *Recursive Function Theory: Proceedings, Symposium in Pure Mathematics*, vol. 5, pp. 229–238, AMS, Provelence (1962)
21. Neary, T.: Small universal Turing machines. Ph.D thesis, National University of Ireland, Maynooth (2008)
22. Neary, T., Woods, D.: P-completeness of Cellular Automaton Rule 110. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) *ICALP 2006*. LNCS, vol. 4051, pp. 132–143. Springer, Heidelberg (2006)
23. Neary, T., Woods, D.: Small fast universal Turing machines. *Theoretical Computer Science* **362**(1–3), 171–195 (2006)
24. Neary, T.: Undecidability in binary tag systems and the Post correspondence problem for five pairs of words. In: Mayr, Ernst W., Ollinger, Nicolas (eds.) *32nd International Symposium on Theoretical Aspects of Computer Science, (STACS 2015)*, vol. 30 of LIPIcs, pp. 649–661 (2015)
25. Neary, T., Woods, D.: Small weakly universal turing machines. In: Kutylowski, M., Charatonik, W., Gębala, M. (eds.) *FCT 2009*. LNCS, vol. 5699, pp. 262–273. Springer, Heidelberg (2009)
26. Neary, T., Woods, D.: Four small universal Turing machines. *Fundamenta Informaticae* **91**(1), 123–144 (2009)
27. Neary, T., Woods, D.: The complexity of small universal turing machines: a survey. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) *SOFSEM 2012*. LNCS, vol. 7147, pp. 385–405. Springer, Heidelberg (2012)

28. Pavlotskaya, L.: Solvability of the halting problem for certain classes of Turing machines. *Mathematical Notes* (Springer) **13**(6), 537–541 (1973). (Translated from *Matematicheskije Zametki*, Vol. 13, No. 6, pp. 899–909, June, 1973)
29. Pavlotskaya, L.: Dostatochnye uslovija razreshimosti problemy ostanovki dlja mashin T'juring. *Avtomaty i Mashiny*, pp. 91–118 (1978). (Sufficient conditions for the halting problem decidability of Turing machines. In Russian)
30. Post, E.L.: Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics* **65**(2), 197–215 (1943)
31. Post, E.L.: Absolutely unsolvable problems and relatively undecidable propositions - account of an anticipation. In: Davis, M. (ed.) *The undecidable: basic papers on undecidable propositions, unsolvable problems and computable functions*, pages 340–406. Raven Press, New York (1965). (Corrected republication, Dover publications, New York, 2004)
32. Rogozhin, Y.: Small universal Turing machines. *Theoretical Computer Science* **168**(2), 215–240 (1996)
33. Rogozhin, Y., Verlan, S.: On the rule complexity of universal tissue P systems. In: Freund, R., Păun, G., Rozenberg, G., Salomaa, A. (eds.) *WMC 2005*. LNCS, vol. 3850, pp. 356–362. Springer, Heidelberg (2006)
34. Rothmund, P.W.K.: A DNA and restriction enzyme implementation of Turing Machines, In: Lipton, R.J., Baum, E.B. (eds.) *DNA Based Computers: Proceeding of a DIMACS Workshop*, vol. 2055 of DIMACS, pp. 75–119. AMS, Princeton University (1996)
35. Shannon, C.E.: A universal Turing machine with two internal states. *Automata Studies*, *Annals of Mathematics Studies* **34**, 157–165 (1956)
36. Siegelmann, H.T., Margenstern, M.: Nine switch-affine neurons suffice for Turing universality. *Neural Networks* **12**(4–5), 593–600 (1999)
37. Wang, H.: Tag systems and lag systems. *Mathematical Annals* **152**(4), 65–74 (1963)
38. Watanabe, S.: 4-symbol 5-state universal Turing machine. *Information Processing Society of Japan Magazine* **13**(9), 588–592 (1972)
39. Wolfram, S.: *A new kind of science*. Wolfram Media Inc (2002)
40. Woods, D., Neary, T.: On the time complexity of 2-tag systems and small universal Turing machines. In: 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 439–446. IEEE, Berkeley, California, Oct. 2006
41. Woods, D., Neary, T.: Small semi-weakly universal Turing machines. *Fundamenta Informaticae* **91**(1), 179–195 (2009)