

Some notes on the foundations of universal computation and the decidability/universality frontier.

Hector Zenil

hectorz@andrew.cmu.edu

hector.zenil@lifl.fr

Tuesday, March 25, 2008

Summary

The 2,3 Turing machine that Stephen Wolfram suspected to be universal since the publication of his book *A New Kind of Science* (NKS) in 2002, turned out to be universal after the Alex Smith proof. This was made by means of allowing an infinite non-periodic but computationally simple enough background supporting but not performing the actual computation carried out by the machine. The aim of this notebook is to survey the necessary and sufficient conditions of computational universality and the consequences after the proof in question taking the seminal concept in computer science of computational universality to its limits.

Introduction

Various tools and methods have been devised for proving computational universality. Most of them involve the emulation of another system already known to be universal. A proof following that way can show emulation either of a specific universal system or of a whole class of systems that contain universal examples, for instance, tag systems.

Alexander Smith, a student from the University of Birmingham in the UK, has proved by means of emulating 2-color tag systems through a sequence of translations, that the 2,3 Turing machine that Stephen Wolfram suspected to be universal since the publication of his *New Kind of Science* (NKS) in 2002, is in fact a universal Turing machine. Tag systems, invented by Emil Post in 1920, are sets of rules that specify a fixed number of elements to be removed from the beginning of a sequence and a set of elements to be appended onto the end based on the elements that were removed. An m -color tag system denote a tag system in which its rules always delete m elements from the beginning of the sequence. For each $m > 1$, the set of m -color tag systems is Turing-complete. In particular, a 2-color tag system can be constructed to emulate a universal Turing machine. A Turing machine can be shown to be a universal Turing machine by proving that it can emulate a complete class of m -color tag systems with $m > 1$.

A Turing machine M is an abstract device that performs operations over a tape. The tape is unbounded in both directions and divided into single squares. Each square of the tape has a color from a fixed finite list of colors. Usually a color often called "blank", supposed to represent the "emptiness" of the tape, is used to fill the tape in both directions, this color can be regarded as the background over which the Turing machine operates. There is a "head" that can read a color at a time, choose to write a new color in place, and then move to the left or to the right. The Turing machine is always provided with a finite input usually called initial condition or initial configuration. The machine also has a non-empty and finite set of states and a partial function that, according to the current state and the color of the tape under the head, determines the next state, what should be written and the movement of the head. Usually denoted by δ it fully describes the behavior of a Turing machine. This description constitutes the standard model of a Turing machine, namely a one bi-infinite tape, one head, deterministic, with a blank color and a halting state machine. A whole kind of variations turned out to be equivalent in terms of computational power, such as adding several tapes or heads, constraining the tape to a single direction or allowing non-deterministic transitions, none of them provide any additional computational power.

Universal computation is the central concept in computer science since Alan Turing published his paper in 1936. It can be regarded in fact as the birth of computer science itself. A universal Turing machine is a Turing machine able to perform, via a suitable encoding, the computation of any other possible Turing machine. Turing model plays also a variety of explanatory roles in science, for instance, Turing's model is an example of how a computation by a human being or a mechanical device could be performed step by step and it is also used as the fundamental unit in many fields such as computational complexity and algorithmic information theory. Beginning in the early sixties Marvin Minsky built a 7,4 universal Turing machine and a race with Watanabe, Rogozhin and others, in order to find smaller universal Turing machines began. Minsky's machine simulates Turing machines via 2-color tag systems, which were proved by himself and Cooke to be Turing-complete. Recently, Neary and Woods gave small universal machines that simulate Turing machines via a new variant of tag systems, what they called bi-tag systems. It turned out that tailoring variations of tag systems was an extremely powerful tool to find and construct smaller Turing machines. Six months ago a research prize to find the final the smallest universal Turing machine was announced. A large fraction of Wolfram's NKS deals with presenting minimal systems. The set of this prize was to motivate the research on Wolfram's minimal program. In his NKS, Stephen Wolfram found a universal 2,5 Turing machine and suggested that a particular 2,3 machine might be universal. If so, because it is known that no universal Turing machine is possible with 2 states and 2 colors, that 2,3 Turing machine would be the smallest. Today the answer to the question has come in an exciting way. That particular Turing machine is universal and its proof didn't come alone, we have learned a lot and Alex Smith has done a remarkable contribution by shaking the foundations of the concept of computational universality.

One traditional aim in math is to look at the most simple model or theory in a way that no axiom be redundant or derivable from the others. In computer science one can also ask for the minimal description of a computational model. One can wonder in general what are those essential elements necessary for universal computation. An essential element would be essential as far as it is necessary for the description of a universal Turing machine for which without it, the description would no longer be that of a universal Turing machine. That would happen, as it is well-known, if one does not allow a set of states, which

play the role of a kind of memory that supplies a universal Turing machine, among with the unbounded tape, with its characteristic power. As for the colors since without them there would be nothing to compute with.

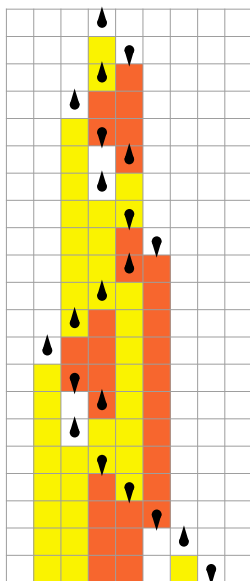
The rules for the Turing machine proved to be universal are :

- {1, 2} → {1, 1, -1}
- {1, 1} → {1, 2, -1}
- {1, 0} → {2, 1, 1}
- {2, 2} → {1, 0, 1}
- {2, 1} → {2, 2, 1}
- {2, 0} → {1, 2, -1}

where this means {state, color} → {state, color, offset}. These rules can be represented pictorially by :



■ **A visual representation of the evolution of the 2,3 universal Turing machine after 20 steps**



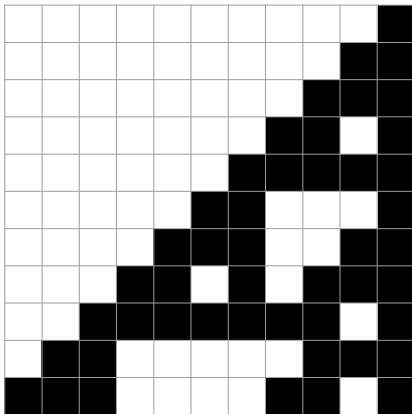
On the necessary and sufficient conditions for universal computation

One could probably think that it would be possible to dispose even of the whole physical description of a Turing machine, namely the head and the tape, and remain with the transition function that characterize its behavior in every sense. And that is totally fair, such descriptions in those terms exist and have been studied for the same time but probably not to the same extent as for Turing machines. Those computational models avoiding physical descriptions continue to be broadly studied as, for instance, functions going from \mathbb{N} to \mathbb{N} , where \mathbb{N} is the set of natural numbers. And one can think of Turing machines in those terms. However, most of the success of the Turing model is that it did not only successfully capture the seminal concept of computation as the others also did, but it does it in more explanatory terms, as history has shown by favoring it in a great extent over the others. Introducing elements such as a head and a tape has allowed people to better understand what is underneath, and finally accept what the Turing model explains, what mechanically and intuitively we call computation (modulo the Church-Turing thesis). Moreover, the intuitive relation with a physical device with pedagogical qualities that would be lost in behalf of a higher abstraction if one disposes of them.

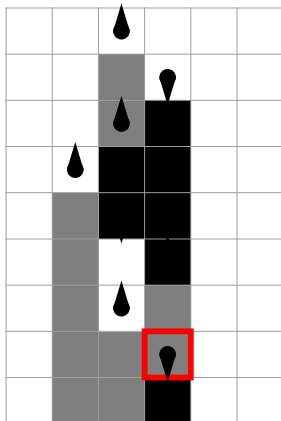
Halting state vs. unbounded computation

It has been agreed, not without some initial reluctance, that halting states as a special state in the set Q of states of a Turing machine that tells the machine to halt, is an arbitrary element in the minimal description of a Turing machine that can be taken away so the Turing machine would remain a Turing machine that potentially would compute up to reach a special halting configuration. That was pushed and evidenced through other studied systems behaving in a similar fashion such as tag systems itself that can meet other termination criteria, or for cellular automata for which, in general, no clear termination exists other than an arbitrary termination step. Since the universality proof of rule 110 cellular automata, the concept of universality was no longer necessarily associated to a proof of undecidability of its halting state. Other concepts have come since, such as partial halting problems and unbounded computation as in the study of cellular automata. In the same way, one would be able to wonder where a computation begins and if it is not just a continuation of another. A consequence of the arbitrariness of the halting state is therefore the arbitrariness of the starting state since the starting state could be any other step between, such as in the case for the halting.

- **Cellular automaton rule 110 arbitrary halted after 10 steps**



- **Turing machine halted after reaching the down-arrow state and a gray cell (an arbitrary terminal configuration)**



■ Unbounded vs. infinite tape

Being that 2,3 Turing machine universal one would fairly ask to perform and show an actual computation, let's say that we want to feed the machine with some encoding of the arithmetic operation 2^2 and get the correct answer 4. One has to bear in mind that these devices are already so small that any encoding of even very simple operations require a lot of effort under the description of the proof, but that doesn't mean that there is no other way to compute that operation with the same machine in a simpler way.

A suitable encoding is initially required because one would need to be able to encode by hand such an initial condition, recalling that the difficulty resides in the lack of elements and rules--the product of the colors and states--to describe something intuitively meaningful for us. Even if you find a simple intuitive encoding for that operation, if you want to perform any other you would require to find or build another different encoding unless you come up with a general suitable encoding just as the one used for the 2,3 Turing machine proof letting it to behave universal. Then you probably could conceive a compiler, either with your own encoding or the encoding in the current proof including a translation between the tag system and any expression in regular mathematical language or any other high-level programming language. A fair question to make is of course whether or not such a compiler or the Turing machine itself would be computationally efficient. And the relationship between universality and complexity classes at the border of decidability has begun to be recently discussed in this same context and we will say more, regarding the 2,3 Turing machine, below.

For descriptions of a Turing machine, it is often found that they are provided with unbounded tapes, whether to one or both directions. This is somehow tricky and unleashes an old debate with philosophical and practical content. If one allows an actual infinity one can easily wonder if the model is longer feasible since nothing seems to suggest that we can use or take advantage of any non-finite element in physical terms. Even if one is not intending to construct a Turing machine but to think of such kind of computation around, one could still wonder if such a "natural" Turing machine would have access to an infinite source, if so, then the Turing machine would also easily reach capabilities beyond the characteristic power of the Turing machine model. One way to overcome these troubles (neither being able to provide a finite tape nor an actual infinite tape) is by allowing the tape to grow indefinitely in length. In favor of that term, one can put all digital computers into that class by thinking that it is always possible to provide the machine with some extra memory as long as the computation need it. But the concept in question, if the tape is actually infinite or not, has a lot to do with the actual content of the tape. When the tape is filled with a "blank" color, one is actually filling it. How is that would be compatible with the notion of a finite configuration or initial condition? Well, the way to round the problem is by thinking that the supposed color is actually the physical background that supports the Turing machine computation and that it does not play any particular role other than supporting it. And that is also compatible with both cases, either constructing such a device, like a digital computer, or finding one around as a representation of a physical phenomena.

■ "blank" symbol vs. arbitrary background

Over the years, small universal machines were given for a number of variants on the standard model that in principle preserve the essence and most minimal technical requirements on computation and universality. One variation on the standard model is, for instance, to allow the "blank" symbol of the Turing machine's tape to be an infinitely repeated word to one or both sides of the tape.

One can imagine feeding this machine with finite inputs while the background over which the machine operates looks computationally simple supporting the computation but neither doing it nor contributing to the overall computational power as we said in the section above.

So one immediate question that one can draw is why changing an infinite background of period $n=1$ --the "blank"--for $n>1$ would matter by also using more colors as long as the content of the tape remains computationally simple. Let's dig further into that matter:

Let's define a couple (M,A) of 2 automata M and A with $M=\{S,C\}$ where S is the set of states and C the set of colors, and $A=\{s,c\}$ where s is a subset of S and c a subset of C . Then we will say that M and A are compatible since they have compatible states and colors, i.e. states and colors that both can syntactically manipulate. And let's assume that the couple (M,A) is proved to be universal, and that A is proved to be not, does that imply that M is universal? Not necessarily, that depends on how important is A in the operation of (M,A) . If M is universal by its own, it shouldn't be necessary to keep A . But if A interacts with M at every step t , then M cannot reach universality by itself but with external help. On the other hand, if A is used to feed M at $t=1$, A is a translator of M from the output of another system into the vocabulary of M in terms of S and C . Therefore, if A is not universal and does not intervene in any other step but $t=1$ then M stands for the computational power of the couple (M,A) , i.e. it is universal.

An auxiliary machine like A could be, for instance, a finite automaton generating a tape with a background for M . The role of A in Alex Smith's proof for the Wolfram's 2,3 Turing machine consists in translating rather than computing. However, even when the background is computationally simple, the simplicity of the 2,3 Turing machine is impaired by its extensive operation-dependent processing of the sequential cyclic tag-system that emulates and a direct consequence of the lack of elements $S \times C$. A cyclic tag system is a tag system in which the list of rules is applied in sequential order.

The useful part can be therefore recognized by the bounds determined by the background and the background actually works as a support for the computation itself. Minsky required initially finite-length initial conditions over a tape filled with a repetitive "blank" character. Later, others allowed and proposed infinite but repetitive backgrounds which was already a common situation in the study of cellular automata from which rule 110 took advantage for its own universality proof. Alex Smith is pushing this generalization further up to its border by allowing non-repetitive backgrounds. By emulating a sequence of computational systems Smith proves that the 2,3 Turing machine by emulating any 2-color tag systems for an arbitrary number of steps, using a finite-length initial condition defining an initial condition in which only finitely many cells are relevant, and no other cells are visited during the evolution of that initial condition.

Basically the possible scenarios consist in adding a condition of background with an unbounded simple pattern going all along the tape surrounding the "meaningful" part (the actual input for M). This special pattern or (also called word) is constituted by a subset of the set of colors C filling the unbounded tape as follows: a repeated pattern to the left of the actual input and another pattern repeated to the right of the input word, i.e. at the initial configuration the tape looks like $\dots.LLLLLwRRRRR\dots$ where w is the input for M and L and R are arbitrary patterns or "blank" words over the tape, all w , R and L belonging, when decomposed, to the set of colors C . Furthermore, the configuration of the tape can look as LwR with L and R non-repetitive backgrounds but produced by A , the non-universal device.

In accordance to Martin Davis definition of universality from the paper cited in the bibliography, a condition that must be added is that the encoding itself be computationally simple. For there would not be much point of claiming universality for a Turing machine for which the encoding would require another universal machine to carry it out. The problem is then to encode the computation in terms of M such that the encoding itself does not require another universal machine. So even when the state of the tape could look rather sophisticated since the beginning, it is computationally speaking very simple.

This proof extends what is done in systems such as cellular automata and by doing so one can even go further asking which

other systems could benefit from this in a reasonable and sound way. Alex Smith's use of non-periodic backgrounds is a natural generalization of the sort of ideas in Wolfram's New Kind of Science, and somebody was bound to make that generalization eventually. As I suggested above, this considerations approaches a more natural and closer connection with physical reality and physical constraints which is by itself a rich field of further analysis.

■ **On the feasibility of non-periodic configurations**

As explained above these particular kind of backgrounds not playing a main role in the computation can be defined as generated by a computationally simple program so one can provide the Turing machine with a tape filled with the suitable pattern without having to perform any further computation. One can think that the process is tantamount to filling an unbounded tape with a blank, it doesn't matter how you color the tape, as far that that color doesn't encode the actual computation nor performs it by itself. Therefore, one can think about it as the support over which the machine operates, just as the same role the blank plays in the traditional model. This approach seems even more closer to possible physical constraints since one can think in computations having place around over some kind of background/noise and not over "blank" fixed tapes. In other terms, one can think on the process as providing a filled tape with a sequence of symbols previously calculated or calculated just before the actual computation simple enough to be able to provide with more tape as the computation requires, just as it does for the "blank tape".

Computationally simple backgrounds (initial fragments of one-directional tapes) :

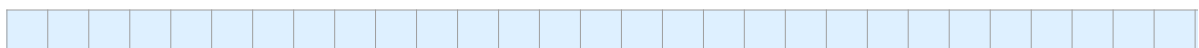
■ **The traditional "blank" tape, with "blank" as an actual white color.**

```
Row[{ArrayPlot[{Table[White, {30}]}], Mesh -> All, ImageSize -> 600}, ". . ."]]
```



■ **A blue-colored "blank" tape.**

```
Row[{ArrayPlot[{Table[LightBlue, {30}]}], Mesh -> All, ImageSize -> 600}, ". . ."]]
```



■ **Other possible repetitive backgrounds.**

The easiest way to make a recurrent sequence is to form a periodic sequence, one where the sequence repeats entirely after a given number m of steps.

```
Row[{ArrayPlot[{Flatten[Table[{0,1},{15}]}], Mesh->All, ImageSize->600}, ". . ."]]
```



```
Row[{ArrayPlot[{Flatten[Table[{0,1,1},{10}]}], Mesh->All, ImageSize->600}, ". . ."]]
```



■ **Non-repetitive but computationally simple.**

Based on the recurrent construction of a non-rational number by appending a black to each group after putting a white.

■ On the abundance of universal systems

After the universality of rule 110, this discovery establishes once again a new remarkably low threshold for universality, and this time the ultimate boundary for the most important of the abstract models of computation : the Turing machine. This scientific discovery was made possible by the fact the computational universe is even richer than we had imagined. Not only because there are plenty of these systems around but because, as proved again, even the most simple conceivable systems are capable of universal computation. Wolfram's Principle of Computational Equivalence (PCE) claims that all kinds of simple systems support universal computation.

A relaxed definition of universality would make any sense only as far as it is able to distinguish between what it is from what it is not, under the definition. Suppose that under this relaxation it turns out that any Turing machine could be set for universal computation. So, let's explore the only two conceivable possibilities : (a) that it turns out that any Turing machine can become a universal Turing machine under some suitable encoding with the encoding itself not being universal; or (b) it turns out that the set of universal Turing machines is bigger than what was initially expected but still not big enough so that there are non-trivial Turing machines for which, under all possible encodings, they still do not and cannot reach computational universality. Concerning (a), it seems very unlikely since one can still consider completely silly machines always moving their head to the right or to the left or simply repeating an instruction disregarding the content of the tape. But both (a) and (b) strongly strength Wolfram's PCE by placing the notion of computation itself as an ubiquitous phenomena.

References and bibliography

- Davis, M.D. "A note on universal Turing machines." Automata Studies, Shannon C.E. and McCarthy, J, editors, Annals of Mathematics Studies, Princeton university Press : 1956.
- Margenstern, M. "Frontier between Decidability and Undecidability: A Survey." Theoretical Computer Science 231, no.2 (2000) : 217 - 251.
- Margenstern, M. and Pavlotskaya, L. "On the optimal number of instructions for universality of Turing machines connected with a finite automaton." International Journal of Algebra and Computation, 13 (2) (2003): 133 –202.
- Minsky, M. Computation : Finite and Infinite Machines. Prentice-Hall, Inc., 1967.
- Post, E. "Finite Combinatory Processes--Formulation 1." Journal of Symbolic Logic 1, no.3 (1936) : 103 - 105.
- Rogozhin, Yu. "Small Universal Turing Machines." Theoretical Computer Science 168, no.2 (1996) : 215 - 240.
- Shannon, C. "A Universal Turing Machine with Two Internal States." Automata Studies.Princeton University Press (1956) : 157 - 165
- Turing, A. "On Computable Numbers, with an Application to the Entscheidungsproblem." Proceedings of the London Mathematical Society Series 2, 42 (1937) : 230 - 265.
- Turing, A. "On Computable Numbers, with an Application to the Entscheidungsproblem. A Correction. "Proceedings of the London Mathematical Society Series 2, 43 (1938) : 544 - 546
- Woods, D and N. Turlough, "The Complexity of Small Universal Turing Machines", Springer, 2007
- Woods, D and N. Turlough, "Small weakly universal Turing machines.",
- Wolfram, S. A New Kind of Science. Wolfram Media, 2002