Computation in a Universe with Time Travel Looks Very Quantum-ish

Greg Baker (gregb@ifost.org.au)

February 20, 2016

Abstract

In this paper I propose an alternate consistent way of reasoning about computation in the presence of time machines by using cellular automata. Interestingly, some behaviour spontaneously arises out of a such a system that looks very much like the reality we live in. It is of course co-incidence, but interesting nonetheless.

List of Figures

Contents

1 Background and Motivation

My goal for some time now has been to prove that BQP (the set of bounded error, probabilistic problems which can be solved in polynomial time on quantum computers) is the same as BTP (the set of problems which can be solved by a computer which has access to a time travel facility).

The longer-term goal is to prove that if there are any serious and sensible constraints with what can be done using a time machine, that NP_C (the set of NP-hard problems on a classical computer) is equal to NP_T, which is of course a much more extraordinary claim.

1.1 A bad starting point

I have completely failed to do this, naturally enough. One of the most significant impediments to this is the problem of reasoning about temporal computing.

A naive approach (which I briefly proposed in my honours' thesis) was to extend a classical computer with a special function: "set variable x to be y at time t". Unfortunately, this suffers the computational equivalent of the grandfather paradox, such as the program in figure 1.

```
T=1 X := 1
T=2 SLEEP
T=3 X := (1 - X) at T=2
T=4 PRINT X
```

Figure 1: The grandfather paradox expressed in temporal computing pseudo-code

What output should it give? No possible output makes sense with classical reasoning.

1.2 At rock-bottom, still digging

If one considers the PRINT statement in listing 1 to be an observation of a quantum system the grandfather paradox could be resolved. The state at T=2 and T=3 would simply be:

$$\frac{1}{\sqrt{2}}|X=1\rangle + \frac{1}{\sqrt{2}}|X=0\rangle$$

The PRINT statement causes the wave function to collapse into one of two possible values with equal probability.

A small variation on the program in figure 1 is in figure 2 where almost any X, $-1 \leq X \leq 1$ is a possible output. Further modifications to the program could create entangled variables, and presumably further modifications still could produce complex-valued "probabilities" associated with each state.

But is this logically consistent? How did we get from a program to a pseudo-wavefunction? What are the precise semantics of the "set variable x to y at time t" statement? In particular, a temporal computer should surely be able to run two temporal operations

```
T=1 X := 1
T=2 DECLARE Z AS COMPLEX
T=3 Z := E^(i * X)
T=5 X := REAL(Z) AT T=2
T=6 PRINT X
```

Figure 2: A smear expressed in temporal computing pseudo-code

.

```
T=1 X := 1
T=2 X := 2 at T=1
T=2 X := 3 at T=1
```

Figure 3: Lucky dip universe – what is the value of X?

at the same time (parallel processing of time statements), so what would the output of the program in figure 3 be?

2 The strong chronology protection approach

One consistent way of approaching temporal computing is to assume that a time machine cannot alter history in any way at all. In human terms, a time traveller going backwards in time is merely acting out a role which occurred in the time traveller's own historical time-line[1].

This approach does allow for the creation of information from nothing, since it is consistent for a time traveller returning through history to provide the plans for a time machine to the time traveller's younger self and for the time machine to be built from these plans.

There are two problems with this approach to temporal computing, though:

1. In order to preserve consistency, any information created from nothing in a strong chronology-protection universe would have to be either:
   - copied from its original source with a flawless copy mechanism so that the copy could be sent back in time, or
   - on a medium which does not decay or degrade in any way as it ages – if the original is sent back in time – since the original would have an undefined age.

   Thus a temporal computer operating under the strong chronology protection principle would be intolerant of single-noise of any kind.
2. This still does not support "set variable x to y at time t" properly – either the set occurred once, or it did not. There is nothing in the model which can give

[1] This would itself have implications for whether the time traveller had any free will whilst in history, and might indicate superdeterminism.

```
POSITRONIC_VARIABLE x
CLASSICAL_VARIABLE y

y := x
if (y < 2 or y > sqrt(102241))
then
  y := 2
end if

if (102241 mod y == 0)
then
  x := y
else
  x := y + 1
end if
```

Figure 4: Constant-time Factoring 102241 using positronic variables

a meaningful way of reasoning about the behaviour of the system if "set" occurred twice at the same time on the same variable.

More significantly, it's a very boring model of time travel.

3 Obligatory summary of prior work in this area

Talking about "prior work" in relation to time-travel equipped computers begs a great many jokes. Damian Conway[2] has an oft-presented talk "Temporally Quaquaversal Virtual Nanomachine Programming In Multiple Topologically Connected Quantum-Relativistic Parallel Spacetimes...Made Easy!" which has perhaps the highest joke-density of any work on the relationship between temporal computing and quantum theory.

The Damian Conway model of temporal computing uses "positronic" variables. A positronic variable can interact with a classical variable in two different ways.

- A positronic variable can be assigned the value of a classical variable.
- A classical variable can be assigned the value of a positronic variable.

The trick is that the value read out of the positronic variable is the value that is put into it in the future, not the value from the most recent past as with a classical variable. An example program is given in figure 4.

However, there is nothing to stop the program of figure 5 being written. Or, more significantly, nothing to stop that program being run, with possibly catastrophic effects on the universe's time-line.

[2] Who is – as far as I am aware – unrelated to John Horton Conway

```
POSITRONIC_VARIABLE x
CLASSICAL_VARIABLE y

y := x
x := y + 1
```

Figure 5: An inconsistent program

4 A logically consistent temporal computing model using cellular automata

1. **The game is played out on a finite board.** Without this constraint, every cell depends on an infinite number of other cells.
2. **The first generation is not constrained.** This is equivalent to someone setting up the computation initially.
3. **For the second and subsequent generations, if a cell has 5 or 6 neighbours alive among its neighbour cells one generation in the past or one generation in the future, then it is alive.** That is, if a cell had two alive neighbours in the previous generation, and three alive in the following generation, it is alive.

Figure 6: Rules for the evolution of Temporal Life

Conway's Game of Life and its family of related cellular automata need no introduction. Various authors have written extensions to the game to produce a quantum version, where the cells are in a superposition of states. They all share a common feature: each generation is computed from the one before.

In order to permit communication backwards in time, **temporal cellular automata cannot be uniquely computed from preceeding generations**.

The rules of a temporal automaton can only provide rules that can determine whether a particular game is consistent.

An example definition of a temporal automaton which will be used throughout this paper is given in figure 6 which – for want of a better name – is referred to as "Temporal Life" from here onwards.

4.1 Reasoning

A Temporal Life board is a function $B(t,x,y) \rightarrow \quad alive \quad | \quad dead$. It can also be considered as a sequence of generations $G_1, G_2, \ldots$ where each generation G_n is a set

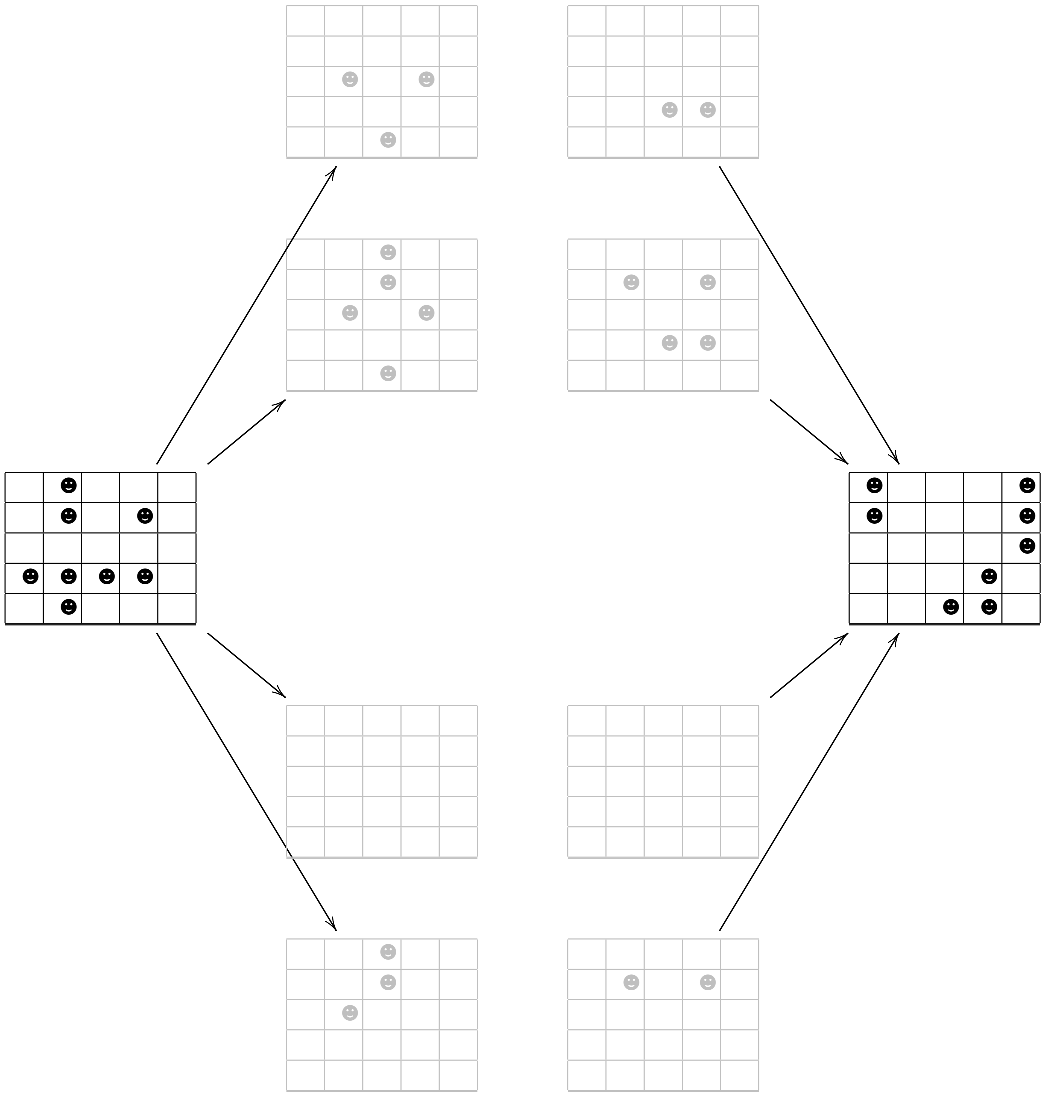


Figure 7: An observer has measured a grid of 25 cells from generation 1 and generation 4. There are four possibilities for generations 2 and 3, although they pose considerable constraints on the generation before 1 and after 4. Mid-left and mid-right on the second-to-top row of generation 3 are 100% correlated despite being separated by a 0% correlated cell.

of alive points $(x_{in}, y_{in})\ldots$.

Some questions which can be asked about a game of Temporal Life are:

- Given a set of observations $t_i, x_i, y_i, B(t_i, x_i, y_i)$, is there an internally consistent B and G_1 which would yield these observations? Is there more than one?
- Given an initial starting condition of G_1 what is the probability across all possible $B(t, x, y)$ functions consistent with G_1 that for some point (t_0, x_0, y_0), $B(t_0, x_0, y_0)$ will be observed to be alive?

The first question asks "have we created a grandfather paradox?". The second question asks "what is the useful computational power of a temporal computer?".

Figure 7 on page 7 shows an example Temporal Life problem – a first and fourth generation are known, and we wish to reason about generation three. Since generation two and three influence each other, there are many possible consistent states.

4.2 Universe Termination

It is possible – in fact, quite likely – that two generations G_{n-1} and G_n might be inconsistent – there is no G_{n+1} which would mean that every cell in G_n satisfied the neighbour relations rule.

This would mean that G_n should never follow G_{n-1}, and unless there is an alternative G'_n which does have a consistent G'_{n+1}, would mean that G_{n-1} should not follow G_{n-2} either.

Thus the probability of a cell being observed alive or dead in the present can depend on very distant future states.

Surprisingly few pairs of generations are consistent. I have been running simulations, looking at pairs of randomly generated boards, where each cell had a $\frac{5}{16}$ chance of being alive. It appears that the probability of two typical-but-dense generations being consistent is less than 5%, with the few generations which are consistent having around 20 alternative followers.

4.3 Complexity Theory

I have not yet proven that a Temporal Life game is Turing-complete. However, it seems likely given the underlying similarity to Conway's Life.

Almost all computations in a Temporal Life game would be probabilistic in nature – set one (or two) initial generations G_1 (and possibly G_2) and then observe some or all elements G_n some time later. Since there will almost always be several alternatives for G_n, the answer will generally be read off probabilistically.

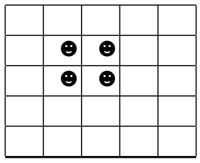


Figure 8: The only non-null stable structures are 2x2 blocks

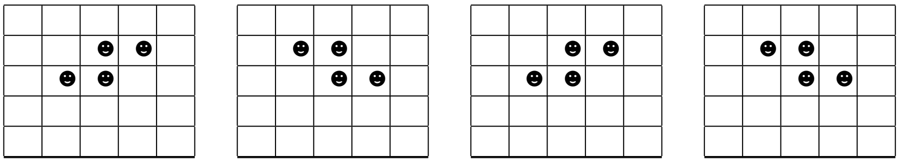


Figure 9: A very simple period 2 structure

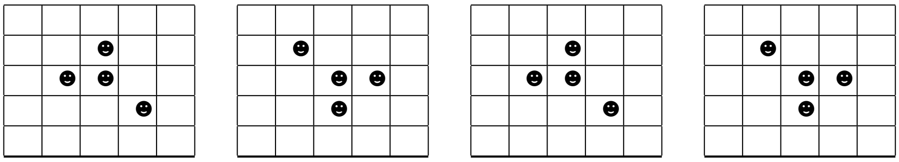


Figure 10: Period 2 structure with diagonal flip line

5 Temporal Life Zoo

5.1 Stable (period 1) structures

A brute-force attack on all 5x5 Temporal Life game boards found only two stable structures (games where for some n, $G_{n-1} = G_n = G_{n+1}$ and is consistent):

- A completely empty generation
- A 2x2 square as shown in figure 8.

Such a stable structure can persist indefinitely through a game. Of course, because two consecutive generations do not uniquely identify a third, a stable structure can launch a completely different generation at any time.

5.2 Period 2 structures

Period 2 structures are surprisingly rare. Figures 9, 10 and 11 (and their rotational and translational variants) appear to be the only examples.

6 Interesting Behaviour in the Game of Temporal Life

There are few real-world analogues to Temporal Life at a macro-scale. An economist might make an analogy between the cells in Temporal Life and regions in a large

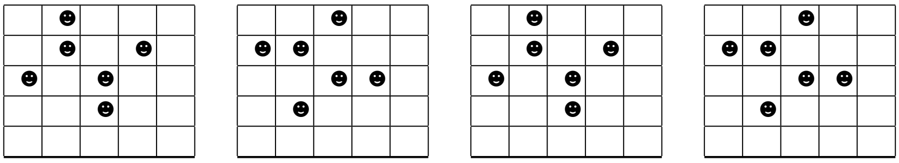


Figure 11: More complex period-2 structure

country, where the future expectation of economic activity in some neighbouring region might permit borrowing against its future revenue stream and maintain the viability of the first region, which in turn causes the neighbouring region to be developed later.

There are however, some intriguing parallels between the the behaviour of Temporal Life games and some basic physics in our universe, which are discussed in the following sections.

- Is the complexity of simulating a game of temporal life using classical computation in the same order as the complexity of simulating quantum computation?
- Is there an arrow of time?
- How similar are time and space?
- Is there a maximum speed?
- Can we get correlation, but not causation across space-like intervals?
- Are virtual particles flitting in and out of existence?
- Do we have a renormalisation problem?

6.1 Simulation Complexity

Suppose it is possible to calculate $B(t_0 + 1, x_0, y_0)$ quickly on a classical computer given $B(t, x, y)$ for all $t \langle= t_0$ and all (x, y) in the vicinity of (x_0, y_0). Then Temporal Life would have no more computational power than a classical computer[3].

However, this seems unlikely. Deducing what values $B(t\_0 + 1, x_0, y_0)$ could take is essentially a boolean satisfiability problem.

The probabilistic approach (given a generation G_{n-1} and G_n, randomly create G_{n+1} until G_n is consistent) falls foul of the universe termination problem of section 4.2 as it can lead to very expensive back-tracking to find an alternate consistent sequence.

Thus there should be an exponential slow-down in simulating a game of Temporal Life on a classical computer, just as there is in attempting to simulate a quantum computer on a classical computer.

[3]And since no-one has yet shown that a Temporal Computer is Turing-complete, it could show that Temporal Life has less computational power than a classical computer

6.2 The Arrow of Time

Notice that the rules of the game in figure 6 (page 6) are completely time-symmetric. If the role of past and future generations are swapped, the rules are unchanged.

And yet, there can be clear directions to the evolution of the system, as shown in figure 12 on page 12.

6.3 Time is similar, but not quite the same to a spatial dimension

Stacking generations on top of each other[4] produces a three-dimensional elongated cube-like structure with various cubic cells alive or dead.

The rules of the game (figure 6) are almost invariant if time is swapped with a spatial dimension. Of the 26 neighbouring cubes in a three-dimensional structure, 16 of them are special and count towards a cell's alive-ness and it is the positioning of these 16 which give time its special preference.

6.4 What can travel at the speed of light?

In Conway's life no structure can step across an empty area of board at the local speed of light[5].

Temporal Life allows infinite waves to travel at the speed of light, but so far I have not been able to come up with any simple, finite structure that does.

Interestingly, nor have I been able to come up with any finite structure which evolves into a structure moving at the speed of light. It is as though accelerating up to the speed of light is impossible.

6.5 EPR paradox

While there is a local speed of light, it is possible for two cells to be separated by a space-like interval and yet still have "entangled" properties.

Consider figure 7 on page 7. If an observer has the state of a grid of 25 cells in the first and fourth generations then there are only four possible consistent games.

Notice the pairing in the third generation. The only differences in each case are two cells in the second row – which are 100% correlated despite being separated by another cell with 0% correlation to either of them.

This reminds me for some reason of the spooky action-at-a-distance of the EPR paradox.

6.6 Something from nothing

Figure 12 shows a phenomenon that has no equivalent in any classical cellular automata. Cells can come alive purely because of the presence of alive cells in the future, which in turn could be alive because of cells in their future.

[4] Which is very hard to do in LaTeX and even harder to print out without a 3D printer.

[5] Other very similar cellular automata do support this.

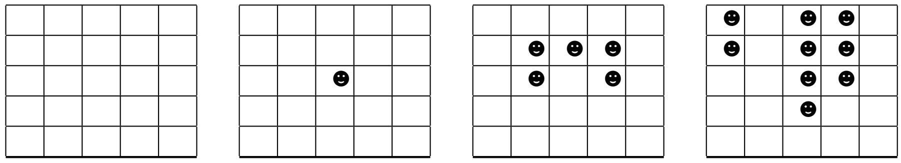


Figure 12: Alive cells can spontaneously create themselves from entirely-dead regions

I have not yet been able to create a complete life-cycle of a "virtual particle" such as the one in figure 12, because of the sheer computational complexity involved.

So while I can show that there is a set of generations[6] G_0, G_1, G_2, G_3, G_4 where G_0 is completely dead, G_1 has one alive cell and in which G_n is consistent with G_{n-1} and G_{n+1} for all $n\langle 4$ I haven't yet been able to extend this sequence out further into a large number of generations.

At the moment, all examples of such sequences have the interesting property that the number of alive cells in G_{n+1} is greater than the number of alive cells in G_n. However, this is not true in general for consistent Temporal Life generations – the sequence G_4, G_3, G_2, G_1, G_0 for example, is consistent and decreases in size.

So it is possible – but highly unlikely – that every possible extension of the series $G_0, G_1, G_2, \ldots$ eventually yields a pair of generations G_{n-1} and G_n for which no consistent G_{n+1} exists.

But if consistent sequences of this nature do exist, they share an interesting similarity to virtual particles flitting in and out of existence in our universe.

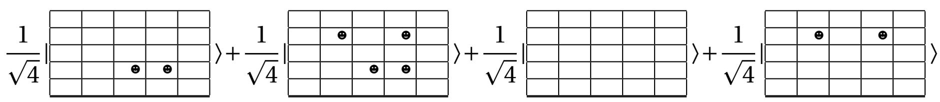
6.7 Renormalisation, Holography, Mach's Principal?

Because of the issues discussed in section 6.6, larger boards will have more and more free space in which a "virtual particle" could appear. These could interact with any other structure on the board eventually. So even a tiny structure in an empty board should be uncomputable if the board were infinite in extent.

Thus, given G_{h0} – a generation on a board of size h – the task of producing the consistent futures $G_{h1}, G_{h2} \ldots$ there is still more computation to do to calculate the consistent futures of G_{k0} where $k\rangle h$. In other words, the boundary of the universe has an impact on the evolution of the universe, even though it may be a space-like interval to get to it.

7 Real valued wave-functions

Again looking at figure 7 on page 7, generation 3 almost begs to be written as:



[6]Generation 5 of figure 12 did not fit into the 5x5 grids I have been using in this paper and so is not shown here.

And if we do this, we can observe that the evolution of the system is unitary.

But of course, the associated "amplitudes" and unitary evolution matrices are all real-valued. Complex valued "amplitudes" would seem to be a necessary requirement in order to reproduce quantum-like computation, along with some mechanism equivalent to interference.

I don't know how to fix this, my ideas so far have been:

- If instead of having a binary state of "alive" or "dead" we have unit vectors (for "alive") and zero vectors (for "dead"), the universe becomes much richer, and computationally almost completely intractable. This doesn't immediately create complex amplitudes, but I'm hoping it might be equivalent to it somehow.
- What if the rules for whether a cell is alive or dead depended were malleable, and depended on the state of the game board?
- What if the neighbour count algorithm took a weighted sum over all the potential boards in the previous generation? This would mean that some board layouts could be part of a generation sequence only if they appeared in parallel with certain other boards, giving several sets of possible histories.

8 Summary and Future Directions

As far as the author is aware, no-one else has proposed a logically-consistent model of computation in the presence of time travel which does not suffer from grandfather paradox problems.

This paper has improved on this just a little, by developing a restricted (but logically sound) model of temporal computation. This model has some (probably coincidental) unexpected similarities with our universe.