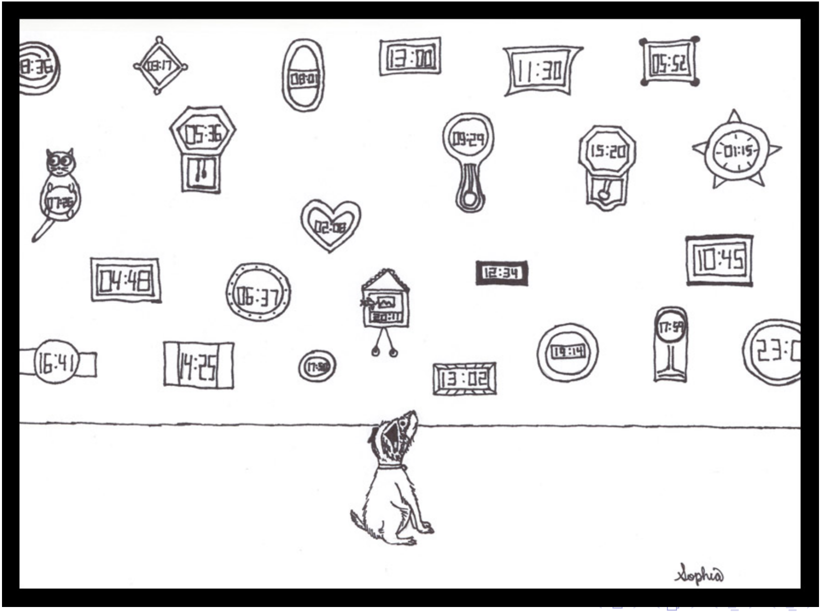
TIME TRAVEL: A NEW HYPERCOMPUTATIONAL PARADIGM

Selim G. Akl

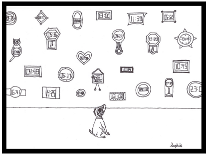
School of Computing
Queen's University
Kingston, Ontario, Canada

SUM ERGO COMPUTO

A computational challenge



A computational challenge

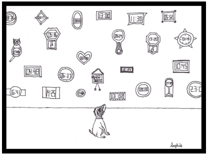


Problem A: For an arbitrary number of clocks N, it is required to compute the average of the N times displayed at a given moment T.

Now, on to the challenge. A universal computer is a computer that is capable of computing any computable function.

Computational Challenge:
Design a universal computer that solves Problem A.

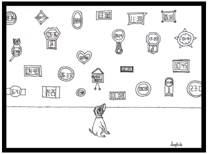
A computational challenge



A valid solution must satisfy the following conditions:

- The computer must be universal in the sense that, once defined, its specifications are fixed once and for all.
- The universal computer should be able to solve Problem A for all values of N (a finite but unbounded positive integer) and at a random moment T selected by the problem poser.
- There is no limit on the size of the memory your universal computer can have.
- There is no limit on the amount of time your universal computer takes to solve Problem A.

A computational challenge



The only limit is on how many elementary operations (read, write, add, etc.) your universal computer can perform per tick: this must be a finite (and fixed) number.

Overview

1. Basic assumptions
2. Non-universality in computation
3. Unconventional computational problems
4. Implications of non-universality in computation
5. Time travel to the rescue
   - Proposals and difficulties
   - What if time travel were possible?
6. Conclusion

Basic assumptions

1. Every computer, be it theoretical or practical, performs a finite and fixed number of elementary operations per computational step.

2. The number of elementary operations per computational step

   - may be a constant (as on your laptop),
   - or it may be a variable (as with accelerating machines),

   but it is always finite and fixed once and for all (even if it is a function of time).

3. Each computational step takes one time unit.

Non-universality in computation

For $n \geq 1$

Π_n = set of all computational problems with n scalar inputs (each of finite size)

C_n = a model of computation capable of at most n elementary operations per time unit (each operating on a finite number of finite-size scalars)

Theorem There exists at least one computational problem $\pi \in \Pi_{n+1}$ solvable on C_{n+1} but not on C_n.

Unconventional computational problems

Problem 1: Time-varying variables

$$F_0(x_0),\ F_1(x_1),\ \ldots,\ F_{n-1}(x_{n-1})$$

The x_i are themselves functions that vary with time:

$$x_0(t),\ x_1(t),\ \ldots,\ x_{n-1}(t)$$

$$F_0(x_0(t_0)), F_1(x_1(t_0)), \ldots, F_{n-1}(x_{n-1}(t_0))$$

Each $x_i(t_0)$ is a physical variable, available in its natural environment, **ready to be operated on**.

$F_i(x_i(t_0))$ can be computed in one time unit **if there is a computer to perform the calculation**.

IMPORTANT

$x_i(t)$ **changes with the passage of time:**

If $x_i(t)$ is not operated on at time $t = t_0$, then after one time unit

$$x_i(t_0)$$

becomes

$$x_i(t_0 + 1)$$

and after two time units it is

$$x_i(t_0 + 2)$$

and so on. Furthermore $x_i(t_0)$ cannot be recovered from $x_i(t_0 + k)$

Problem 2: Time-varying computational complexity

An algorithm consists
of a number of stages:

A stage executed at time t
requires $C(t)$ constant-time operations.

Problem 2: Time-varying computational complexity

$$f_0(x_0), f_1(x_1), \ldots, f_{n-1}(x_{n-1})$$

1. All n functions are entirely independent.
2. Computing $f_i(x_i)$ at time t requires
$$C(t) = 3^t \text{ operations, } t \geq 0.$$

ACCELERATING MACHINE

Suppose an **accelerating machine** is available which can **double** the number of operations that it can perform at each step:

Step 1: 1 operation
Step 2: 2 operations
Step 3: 4 operations
and so on.

Cannot solve the problem for $n > 1$.

Problem 3: Rank-varying computational complexity

An algorithm consists
of a number of independent stages

$$S_0, S_1, \ldots, S_{n-1}$$

Rank of stage S_j = order of execution of S_j

**A stage whose rank is i
requires $C(i)$ constant-time operations.**

For example, $C(i) = 3^i$, where $i \geq 0$.

Problem 4: Interacting variables

$$x_0,\ x_1,\ \ldots,\ x_{n-1},$$

are the variables of a **physical system**.

They need to be **measured** in order to compute

$$F_0(x_0),\ F_1(x_1),\ \ldots,\ F_{n-1}(x_{n-1}),$$

The physical system has the property that

- measuring one variable disturbs any number of the remaining variables unpredictably and irreversibly
  - meaning that we cannot tell which variables have changed value, and by how much.

Mathematical model

$$
\begin{aligned}
x_0(t+1) &= \mathbf{g_0}(x_0(t), x_1(t), \ldots, x_{n-1}(t)) \\
x_1(t+1) &= \mathbf{g_1}(x_0(t), x_1(t), \ldots, x_{n-1}(t)) \\
&\vdots \\
x_{n-1}(t+1) &= \mathbf{g_{n-1}}(x_0(t), x_1(t), \ldots, x_{n-1}(t))
\end{aligned}
$$

Note:

1. When the system is in a state of equilibrium, its variables do not change with the passage of time.
2. The functions $\mathbf{g_0}, \mathbf{g_1}, \ldots, \mathbf{g_{n-1}}$ are too complex or unknown to capture mathematically.

Problem 4: Interacting variables

Suppose we wish to measure $x_0(t_0), x_1(t_0), \ldots, x_{n-1}(t_0)$ at moment $\mathbf{t_0}$, when the system is in a stable state, in order to compute

$$F_i(x_i(t_0)), 0 \leq i \leq n-1.$$

It is easy to measure $\mathbf{x_0}(\mathbf{t_0})$.

This measurement operation will change the state of the system from

$$(x_0(t_0), x_1(t_0), \ldots, x_{n-1}(t_0))$$

to

$$(x_0'(t_0), x_1(t_0), \ldots, x_{n-1}(t_0)),$$

where $x_0'(t_0)$ denotes the value of variable $\mathbf{x_0}$ after measurement.

Problem 4: Interacting variables

Therefore at time $t_0 + 1$,

$$\begin{aligned}
x_0(t_0+1) &= \mathbf{g_0}(x_0'(t_0), x_1(t_0), \ldots, x_{n-1}(t_0)) \\
x_1(t_0+1) &= \mathbf{g_1}(x_0'(t_0), x_1(t_0), \ldots, x_{n-1}(t_0)) \\
&\vdots \\
x_{n-1}(t_0+1) &= \mathbf{g_{n-1}}(x_0'(t_0), x_1(t_0), \ldots, x_{n-1}(t_0))
\end{aligned}$$

Problem 5: Mathematical constraints

$$x_0,\ x_1,\ \ldots,\ x_{n-1},$$

are all available

- they even **already reside in memory**.

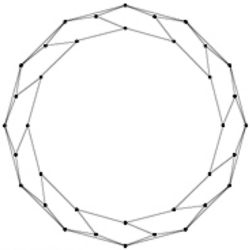
The present computation has three distinguishing properties:

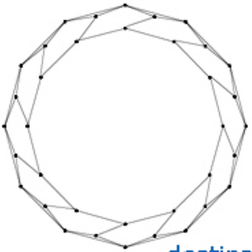
1. All the x_i obey a certain *global condition* (a mathematical property).
   - *This condition must hold throughout the computation.*
   - *Otherwise, the computation fails.*
2. Applying F_i to x_i produces a new value for x_i:

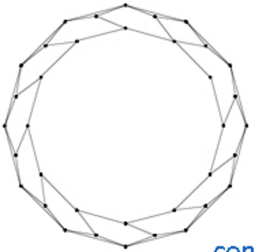
$$x_i^{\text{new}} = F_i(x_i), \quad 0 \leq i \leq n-1.$$

3. If $F_i(x_i)$ is computed for any *one* of the variables **the global condition is no longer satisfied.**

Problem 5: Mathematical constraints







Non-universality in computation

The concept of a Universal Computer cannot be realized.

Computable functions F exist that cannot be computed on any machine U that is capable of only a finite and fixed number of operations per step.

True even if, when attempting to compute function F, computer U:

1. is endowed with an infinite memory,
2. is able to communicate with the outside world, and
3. is given an infinite amount of time.

General statement of the result

Non-universality in computation

Given n spatially and temporally connected physical variables,

$$X_1, X_2, \ldots, X_n,$$

where n is a positive integer,

and a function

$$F_n(X_1, X_2, \ldots, X_n)$$

of these variables,

no computer can evaluate F_n for any arbitrary n unless it is capable of an infinite number of operations per time unit.

Computability does not imply universality

- F_n is readily computed by a machine M_n capable of exactly n operations per time unit (for example, M_n may be equipped with n processors)
- M_n cannot compute F_{n+1}
- F_{n+1} can be computed by a machine M_{n+1}
- M_{n+1} is in turn defeated by F_{n+2}
- and so on ad infinitum.

Even if given an infinite amount of time and space M_{n+i-1} cannot simulate M_{n+i} for $n \geq 1$.

No finite computer is universal

- No computer is universal if it is capable of exactly $T(m)$ operations during time unit m, and $T(m)$ is finite and fixed once and for all.
- Each of the unconventional computations presented so far can be F_n.
  - Time-varying variables paradigm:

    Compute $F_n(X_1(t), X_2(t), \ldots, X_n(t))$ at $t = t_0$.
- While necessarily finite, the problem size n is unbounded.
- It is this unboundedness that defeats any supposed universal computer when faced with problems involving temporal and spatial variables that require computational ubiquity.

Consequences to theory and practice

- The conjectured Church-Turing Thesis is false
    - It is not true that given enough time and space any single general-purpose computer defined a priori can perform all computations that are possible on all other computers.
    - The only possible universal computer would be one capable of an infinite number of operations per time unit.

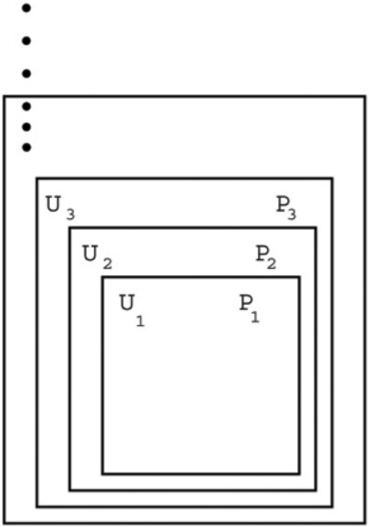
- With M. Nagy I have shown that there exist computations that can be performed on a quantum computer but not on any classical machine (not even in principle by a machine with infinite resources)

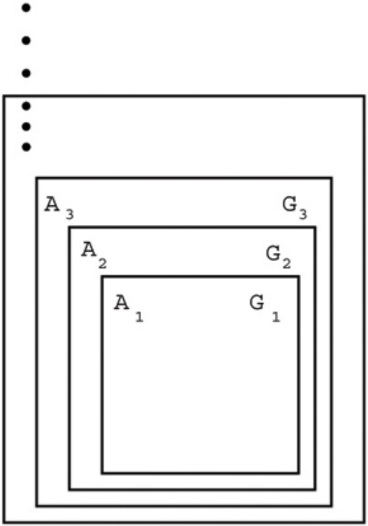
  Consequently, the only possible universal computer would have to be

    - a quantum computer
    - capable of an infinite number of operations per time unit

Non-universality and incompleteness

Non-universality in computation parallels incompleteness in mathematics.





Time travel to the rescue

For the 'time-varying variables' paradigm that disproved universality,

$$x_0(t_0), x_1(t_0), x_2(t_0), \ldots, x_{n-1}(t_0)$$

could we go back in time and pick up the readings that we missed?

Time travel to the rescue

Time travel to the future is exciting, but
What about time travel to the past?

À la recherche du temps perdu

Marcel Proust

In search of information lost
In search of opportunity missed

Profound effect on: Philosophy, Information Theory, History, Anthropology, Physics, Cosmology, Biology, ...

Is there a time arrow?

Standard arguments against time travel

1. Thermodynamic arrow of time
   - Entropy
2. Cosmological arrow of time
   - Expanding universe
3. Electromagnetic arrow of time
   - Unidirectional propagation
4. Quantum arrow of time
   - Decoherence

"Of all the problems which lie on the borderline of philosophy and science, perhaps none has caused more spilled ink, more controversy and more emotion than the problem of the direction of time ... [T]he main problem with 'the problem of the direction of time' is to figure out exactly what the problem is or is supposed to be!"

J. Earman, An attempt to add a little direction to 'The problem of the direction of time', *Philosophy of Science*, Vol. 41, March 1974, pp. 15–47.

Time machines in physics

Einstein's Field Equations of General Relativity are given by

$$G_{\mu v} = 8\pi T_{\mu v}$$

where

$T_{\mu v}$ is the stress-energy tensor,

$G_{\mu v} = R_{\mu v} - \frac{1}{2} g_{\mu v} R$ is the Einstein tensor,

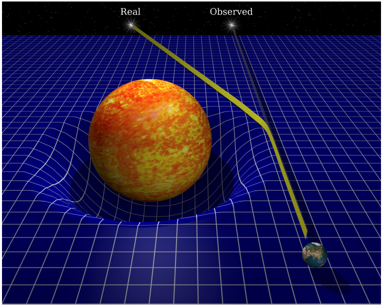
$R_{\mu v}$ the Ricci curvature tensor,

$g_{\mu v}$ is the metric tensor,

R the scalar curvature.

Time machines in physics

The equations $G_{\mu\nu} = 8\pi T_{\mu\nu}$ relate the curvature of spacetime to the matter and energy content of the universe.

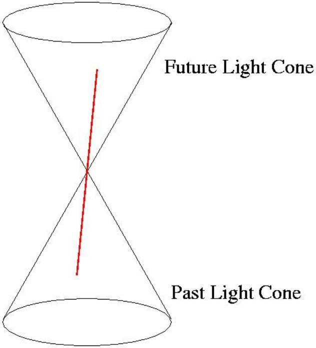


Masses curve the geometry of spacetime
The geometry of spacetime tells masses how to move

Closed timelike curves and space travel

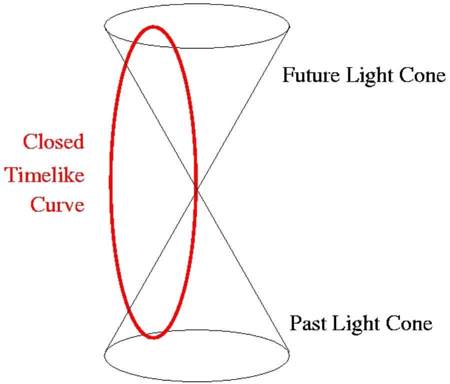
Some solutions to $\mathbf{G}_{\mu\mathbf{v}} = \mathbf{8}\pi\mathbf{T}_{\mu\mathbf{v}}$
lead to so-called closed timelike curves or CTCs defined as follows:

- A light cone represents all possible future positions of an object from its current position



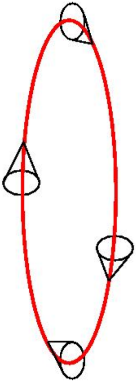
Closed timelike curves and space travel

- If space were curved, a series of light cones that loops back on itself would form a CTC



Closed timelike curves and space travel

A CTC would allow for time travel.



How can we build a time machine in theory?

1. Van Stockum (1937)
   - Infinitely long rotating cylinder
2. Gödel (1949)
   - Rotating universe
3. Kerr-Newman (1963-1965)
   - Rotating black holes
4. Tipler (1974)
   - Sufficiently long but finite rotating cylinder
5. Thorne (1988)
   - Two spheres connected by a wormhole, using negative matter and negative energy
6. Gott (1991)
   - Travel around colliding cosmic strings
7. Alcubierre (1994)
   - Warp drive machines

Recommended reading

Hajnal Andréka, István Németi, Christian Wüthrich
A twist in the geometry of rotating black holes: seeking the cause of acausality
General Relativity and Gravitation, Vol. 40, 2008, 1809–1823.

Difficulties with time travel

Engineering obstacles

- Infinitely long cylinders
- Rotating black holes
- Wormholes
- Negative matter
- Negative energy
- Cosmic strings
- Superluminal warp drives

One more difficulty: Cannot travel to a date earlier than the date when the machine itself was built.

More difficulties with time travel

Paradoxes

1. Grandfather paradox
    - Changing the past to make the present impossible
2. Information paradox
    - Information comes from the future to make same information possible in the future
3. Bilker's paradox
    - Changing the present to make the future impossible
4. Sexual paradox
    - Fathering oneself, a biological impossibility

Resolving time travel paradoxes

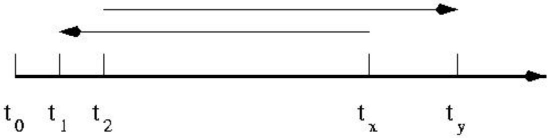
1. Events occur only once
   - For example: If you did not do something at a certain moment in the past, then you cannot go back and do it through time travel, because that moment occurred only once with you in it.
   - **There is no second time**.
2. Novikov's self-consistency principle
   - Invisible law of nature prevents actions leading to paradox
     - Free will?
3. Parallel universes
   - For every choice a new world is created
     - The **Boat of Million Years** of the Ancient Egyptians.

Objections specific to unconventional problems

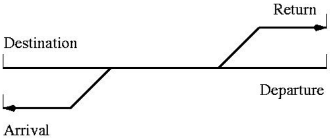
Let us assume our 'universal computer' is allowed unlimited time travel to the past to recover what was missed.

Non-standard arguments against the usefulness of time travel

1. Duration of the trip: It takes time to get there and back!



2. Parallel worlds: You may end up in a different world both times!



3. Philosophical question: Can information be recovered from the past?

Putting all objections aside

Would time travel help restore universality in computation?

1. Time-varying variables
2. Time-varying computational complexity
3. Rank-varying computational complexity
4. Interacting variables
5. Mathematical constraints

Putting all objections aside

Basic Assumptions

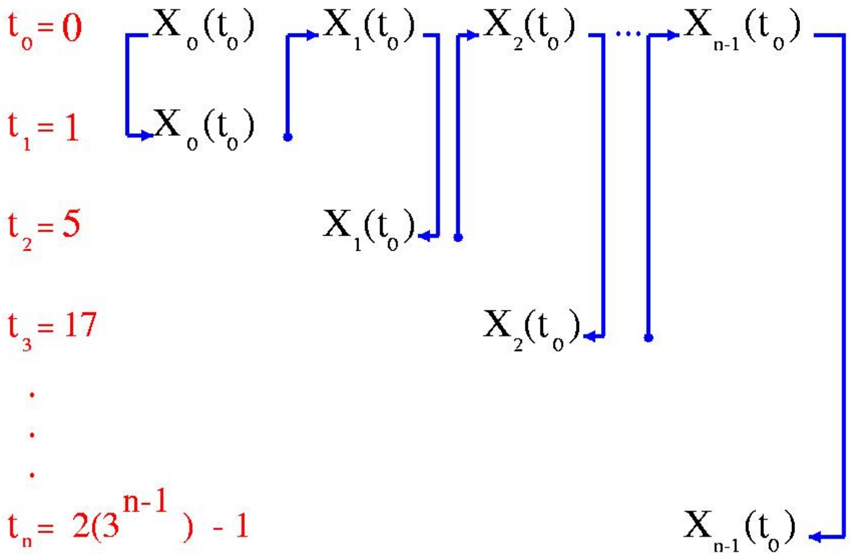
1. The past cannot be changed through time travel
2. Time travel to the past operates on independent entities, possibly in parallel universes

Because of assumption 2, assumption 1 is not violated:

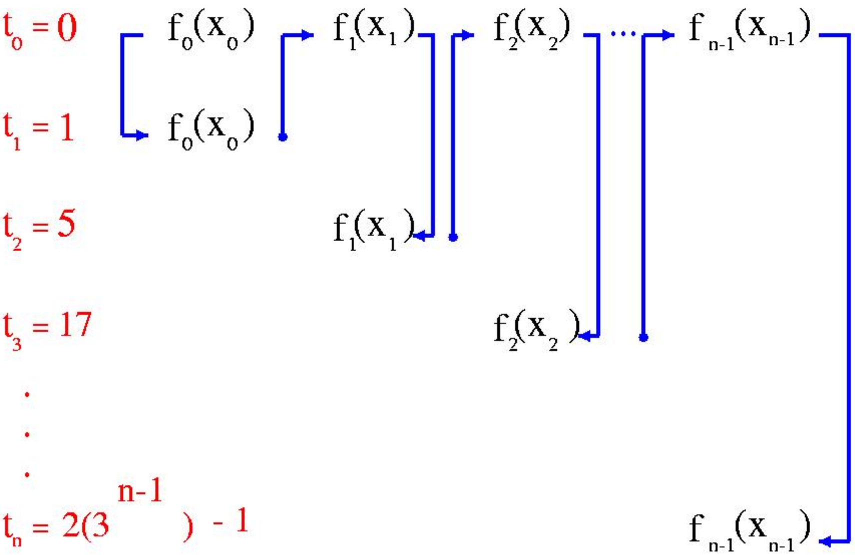
- Obtaining information (e.g. reading a variable), or
- Performing an action (e.g. computing a function)

do not represent alterations of the past.

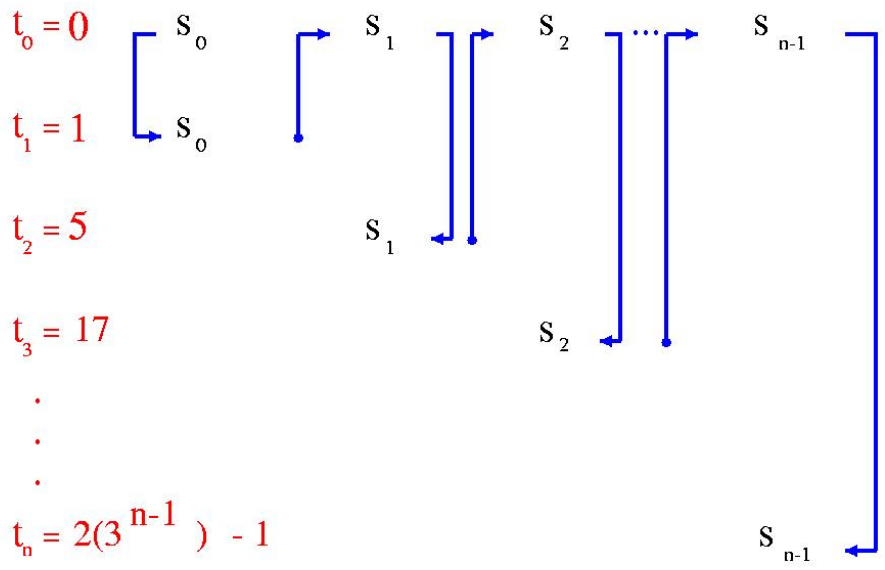
Time travel and time-varying variables



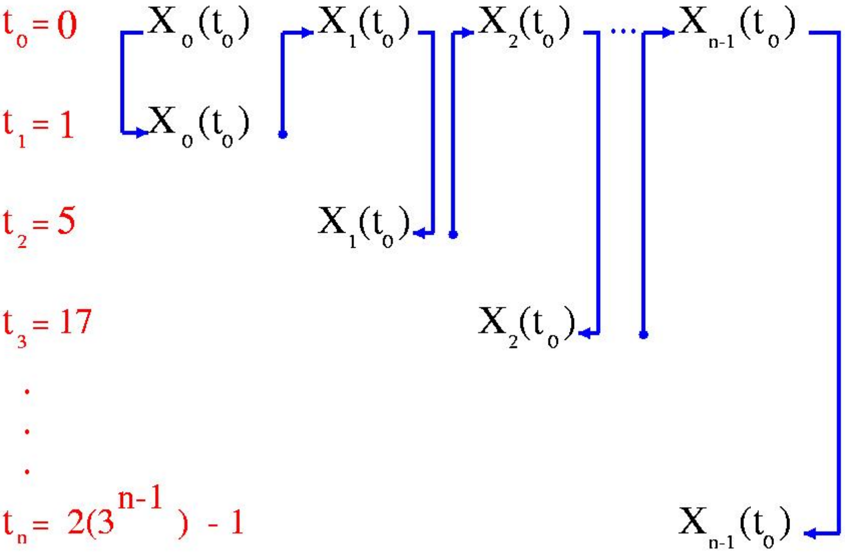
Time travel and time-varying complexity



Time travel and rank-varying complexity

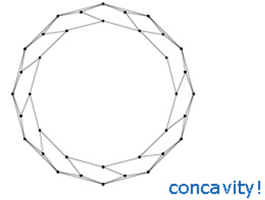


Time travel and interacting variables



Time travel and mathematical constraints

Time travel does not help here:
A concavity is created after the first replacement!



Since we are operating on a unique undecomposable entity, no amount of time travel can help: The past cannot be changed.

Is time the solver of all problems?

"t'as un problème insurmontable?
tiens, pose-le là, sur la table

Laisse passer, laisse passer
et le temps et le temps et le temps et le temps et le temps
te le réglera
okay, okay ....bien"

Gilbert Bécaud

What about time travel to the future?

A computer equipped with a CTC may be able to travel to the future and send itself solutions to problems that are otherwise computationally intractable.

This idea was proposed by Todd Brun in 2008.

It is not clear, however, how travel to the future would solve problems with mathematical constraints that travel to the past failed to solve.

(... and I would rather learn how the stock market will do!)

Time travel is a powerful hypercomputational paradigm

Time travel, assuming all objections are ignored (or addressed), allows us to solve problems that are otherwise only solvable on a universal computer capable of an infinite number of operations per time unit.

These problems include the following ones presented in this talk and involving:

1. Time-varying variables
2. Time-varying computational complexity
3. Rank-varying computational complexity
4. Interacting variables

However ...

The non-universality result is still valid!

Time travel, despite being an extraordinary assumption in itself, still fails to solve the counterexample to universality which involves variables obeying a mathematical constraint.

Conclusion

The requirement of computational ubiquity implies non-universality in computation

For every putative universal computer capable of at most

$V(t)$ elementary operations during time unit t,

it is possible to define a computable function requiring

$W(t)$ elementary operations during time unit t,

where

$$W(t) > V(t) \text{ for all } t.$$

Conclusion

Non-universality poses a serious challenge to some fundamental beliefs in computer science.

Time travel, assuming it is possible, appears to mitigate this inconvenience.

BUT NOT ALWAYS!

Open Question

Can those computational problems amenable to solution by time travel (and those that are not) be fully identified?

Conclusion

A conjecture

Let H be a branch of human thought, such as mathematics, science, philosophy, linguistics, and so on.

A system U is said to be universal over H if it encompasses all the constituents of H, through explanation, generation, or solution, as the case may be.

If U is closed (that is, finite and fixed), then it cannot be universal over H.