Motivation

Absolute computability

‘The great importance of the concept of general recursiveness (or Turing’s computability)…is largely due to the fact that with this concept one has for the first time succeeded in giving an absolute definition of an interesting epistemological notion, i.e., one not depending on the formalism chosen.’ [Gödel (1946) 1990, p. 150]
Motivation

Absolute computability

‘For how can we ever exclude the possibility of our being presented, some day (perhaps by some extraterrestrial visitor), with a (perhaps extremely complex) device or “oracle” that “computes” a noncomputable function? However, there are fairly convincing reasons for believing that this will never happen.’ [Davis 1958, p. 11]
Motivation

Relative computability

‘Troubles with absolutism are deeper and more extensive than these cracks (analogue procedures and newer physics) reveal. For one thing, compute-
ability is relative not simply to physics, but more generally to systems of
frameworks, which include or contain underlying logics.’ [Sylvan and Cope-
land 2000, p. 190]
Hypercomputers

Definition

A **hypercomputer** is any machine (theoretical or real) that compute functions or numbers, or more generally solve problems or carry out tasks, that **cannot be computed or solved** by a Turing machine [Copeland 2002b].
Hypercomputers

Definition

A hypercomputer is any machine (theoretical or real) that compute functions or numbers, or more generally solve problems or carry out tasks, that cannot be computed or solved by a Turing machine [Copeland 2002b].

Super Turing Machines and Non Turing Machines

\[ L \subseteq \Sigma^* \]

Super-TM

\[ L \subseteq \Sigma^* \]

TM

non-TM
Possible Sources of Hypercomputation

- Computability
- Mathematics
- Logic
- Physics
- Biology
- ?

Hypercomputation Model (HM)
Definition

A **oracle Turing machine** (OTM) is a Turing machine equipped with an **oracle** that is capable of answering questions about the membership of a specific set of natural numbers [Turing 1939].
First Hypercomputation Model: Oracle Turing Machines

Definition

A oracle Turing machine (OTM) is a Turing machine equipped with an oracle that is capable of answering questions about the membership of a specific set of natural numbers [Turing 1939].

Hypercomputability features

- If the oracle is a recursive set then $\text{OTM} \equiv \text{TM}$.
- If the oracle is a non-recursive set then $\text{OTM} \equiv \text{HM}$. 
On the ‘Hypercomputation’ Term

Copeland and Proudfoot [1999]:

<table>
<thead>
<tr>
<th>Right</th>
<th>Wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypercomputation</td>
<td>Super-Turing computation</td>
</tr>
<tr>
<td></td>
<td>Computing beyond Turing’s limit</td>
</tr>
<tr>
<td></td>
<td>Breaking the Turing barrier</td>
</tr>
<tr>
<td></td>
<td>Etc.</td>
</tr>
</tbody>
</table>
Definition

An **accelerated Turing machine** (ATM) is a Turing machine that performs its first step in one unit of time and each subsequent step in half the time of the step before [Copeland 1998, 2002a].
Hypercomputation Model: Accelerated Turing Machines

Definition

An **accelerated Turing machine** (ATM) is a Turing machine that performs its first step in one unit of time and each subsequent step in half the time of the step before [Copeland 1998, 2002a].

Hypercomputability features

Since

\[ 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots = \sum_{i=0}^{\infty} \frac{1}{2^i} = 2, \]

an ATM could complete an **infinity** of steps in two time units.
Hypercomputation Model: Analog Recurrent Neural Network (ARNN)

Description [Siegelmann 1999]

\[ \vec{X}(t + 1) = \sigma(\vec{A} \cdot \vec{X}(t) + \vec{B} \cdot \vec{U}(t) + \vec{C}) \]
Hypercomputation Model: Analog Recurrent Neural Network (ARNN)

Hypercomputability features

\[ a_{ij} \in \{\mathbb{N}, \mathbb{Q}, \mathbb{R}\} \Rightarrow \text{ARNN} \equiv \{\text{DFA}, \text{TM}, \text{HM}\}. \]
Standard Quantum Computation (SQC)

Models

Quantum Turing machines (QTM) [Deutsch 1985] and quantum circuits [Deutsch 1989].
Standard Quantum Computation (SQC)

Models
Quantum Turing machines (QTM) [Deutsch 1985] and quantum circuits [Deutsch 1989].

Relation between the models

\[
\begin{align*}
\text{TMs} & \equiv \text{Probabilistic TMs} \\
\text{Reversible TMs} & \equiv \text{QTMs}
\end{align*}
\]
‘Weak’ Hypercomputation Based on SQC

Generation of truly random numbers

\[ U_H |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow \text{measure} \]
‘Weak’ Hypercomputation Based on SQC

Generation of truly random numbers

\[ U_H |0\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \rightarrow \text{measure} \]

1. We observe the superposition state: ‘The act of observation causes the superposition to collapse into either \(|0\rangle\) or the \(|1\rangle\) state with equal probability. Hence you can exploit quantum mechanical superposition and indeterminism to simulate a \textit{perfectly fair} coin toss.’ [Williams and Clearwater 1997, p. 160]
'Weak' Hypercomputation Based on SQC

Generation of truly random numbers

\[ U_H |0\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \rightarrow \text{measure} \]

1. We observe the superposition state: ‘The act of observation causes the superposition to collapse into either \(|0\rangle\) or the \(|1\rangle\) state with equal probability. Hence you can exploit quantum mechanical superposition and indeterminism to simulate a perfectly fair coin toss.’ [Williams and Clearwater 1997, p. 160]

2. The problem: It is not clear how to use this property to solve a non-computable Turing machine problem [Ord and Kieu 2009].
Common misunderstanding

quantum computation $\equiv$ SQC

$\equiv$ adiabatic quantum computation (AQC)
Common misunderstanding

quantum computation $\equiv$ SQC
$\equiv$ adiabatic quantum computation (AQC)

The real situation

Kieu’s hypercomputational quantum algorithm (KHQA) [Kieu 2003]:

finite AQC $\equiv$ SQC
infinite AQC $\equiv$ KHQA
Hypercomputational Quantum Algorithm à la Kieu

Sicard, Ospina and Vélez [2006]:

Classically non-computable $P$ problem (Hilbert’s 10th problem)

Hypercomputational quantum algorithm

Simulation

Partial solution to $P$

Physical referent (Infinite square well)

Dynamical algebra ($\mathfrak{su}(1, 1)$)
Definition

An **infinite time Turing machine** is a Turing machine working on a time clocked by **transfinite ordinals** [Hamkins and Lewis 2000; Hamkins 2002, 2007].

†Figure from Hamkins [2002, Fig. 1].
Hypercomputation Model: Infinite Time Turing Machines

Definition

An **infinite time Turing machine** is a Turing machine working on a time clocked by **transfinite ordinals** [Hamkins and Lewis 2000; Hamkins 2002, 2007].

Description

For convenience, the machines have three tapes:

![Table showing input, scratch, and output tapes](Hamkins_2002_Fig_1)

† Figure from Hamkins [2002, Fig. 1].
Description (continuation)

In stage $\alpha + 1$ the machine works as usual.
Description (continuation)

In stage $\alpha + 1$ the machine works as usual.

In limit ordinal stages the machine works as follows [Hamkins and Lewis 2000, p. 569–570]:

‘To set up such a limit ordinal configuration, the head is plucked from wherever it might have been racing towards, and placed on top of the first cell. And it is placed in a special distinguished limit state.’

‘Now we need to take a limit of the cell values on the tape. And we will do this cell by cell according to the following rule: if the values appearing in a cell have converged, that is, if they are either eventually 0 or eventually 1 before the limit stage, then the cell retains the limiting value at the limit stage. Otherwise, in the case that the cell values have alternated from 0 to 1 and back again unboundedly often, we make the limit cell value 1.’
Description (continuation)

‘This completely describes the configuration of the machine at any limit ordinal stage $\beta$, and the machine can go on computing, $\beta + 1$, $\beta + 2$, and so on, eventually taking another limit at $\beta + \omega$ and so on through the ordinals.’
Hypercomputation Model: Infinite Time Turing Machines

Description (continuation)
‘This completely describes the configuration of the machine at any limit ordinal stage $\beta$, and the machine can go on computing, $\beta + 1$, $\beta + 2$, and so on, eventually taking another limit at $\beta + \omega$ co and so on through the ordinals.’

Hypercomputability features
The halting problem is decidable in $\omega$ many steps by infinite time Turing machines [Hamkins and Lewis 2000].
Hypercomputation Model: Infinite Time Turing Machines

Description (continuation)

‘This completely describes the configuration of the machine at any limit ordinal stage $\beta$, and the machine can go on computing, $\beta + 1$, $\beta + 2$, and so on, eventually taking another limit at $\beta + \omega$ co and so on through the ordinals.’

Hypercomputability features

The halting problem is decidable in $\omega$ many steps by infinite time Turing machines [Hamkins and Lewis 2000].

Remark

Although not related with computability but algorithmic complexity, $P \neq NP$ for infinite time Turing machines [Schindler 2003].
Hypercomputation Model: Infinite Time Turing Machines

Theorem (Hamkins and Lewis [2000, Theorem 1.1])
Every halting infinite time computation is countable.
Theorem (Hamkins and Lewis [2000, Theorem 1.1])
Every halting infinite time computation is countable.

Remark
The infinite time Turing machines have been generalised by the ordinal computability models, which are models based on ordinal numbers. These models include infinite time Turing machines or Turing machines working on tapes of transfinite ‘length’. Seyfferth [2013] shows a brief overview of these models.
The Church-Turing thesis

‘Any procedure than can be carried out by an idealised human clerk working mechanically with paper and pencil can also be carried out by a Turing machine.’ [Copeland and Sylvan 1999]
The Church-Turing thesis

‘Any procedure than can be carried out by an idealised human clerk working mechanically with paper and pencil can also be carried out by a Turing machine.’ [Copeland and Sylvan 1999]

Thesis M

‘What can be calculated by a machine is Turing machine computable.’ [Gandy 1980]
Physical Hypercomputation?

Open problem

The refutation of a general/physical version of Gandy’s Thesis M.
Physical Hypercomputation?

Open problem

The refutation of a general/physical version of Gandy’s Thesis M.

- Based on quantum physics (infinite adiabatic quantum computation)?

Physical Hypercomputation?

Open problem

The refutation of a general/physical version of Gandy’s Thesis M.

- Based on quantum physics (infinite adiabatic quantum computation)?
An Interesting Project: Formal Verification of Hypercomputation in Relativistic Physics

Stannett and Németi [2014] and Stannett [2015]:

Goals

- Implement first-order axiomatisations of theories of the relativity using the proof assistant Isabelle;
An Interesting Project: Formal Verification of Hypercomputation in Relativistic Physics

Stannett and Németi [2014] and Stannett [2015]:

Goals

- Implement first-order axiomatisations of theories of the relativity using the proof assistant Isabelle;
- Add a model of computation carried out by machines travelling along specific spacetime trajectories;

Continued on next slide
Goals (continuation)

- Consider how the power of these computational systems changes according to the underlying topology of spacetime;
An Interesting Project: Formal Verification of Hypercomputation in Relativistic Physics

Goals (continuation)

- Consider how the power of these computational systems changes according to the underlying topology of spacetime;
- Select a recursively uncomputable problem $P$ (for example, the Halting Problem) and machine-verify the following claims:
  - in simpler relativistic settings, $P$ remains uncomputable;
  - in some spacetimes, $P$ can be solved.
Is Hypercomputation a Myth?

Davis’ refutations

Is Hypercomputation a Myth?

Davis’ refutations

Is Hypercomputation a Myth?

Davis’ refutations


Refutation to Davis

Academic Community

Communities

- Hypercomputation Research Network
- Computability in Europe (CiE)
Academic Community

Communities

- Hypercomputation Research Network
- Computability in Europe (CiE)

Books and dedicated journal issues

Final Remarks

‘Once upon on time, back in the golden age of the recursive function theory, computability was an absolute.’ [Sylvan and Copeland 2000, p. 189]

Final Remarks

‘Once upon on time, back in the golden age of the recursive function theory, computability was an absolute.’ [Sylvan and Copeland 2000, p. 189]

‘Via the great pioneers of electronic computing...the 1930s analysis of computation led to the modern computing era. Who knows where a 21st-century overhaul of that classical analysis might lead.’ [Copeland, Dresner, Proudfoot and Shagrir 2016, p. 38]

Final Remarks

‘Once upon on time, back in the golden age of the recursive function theory, computability was an absolute.’ [Sylvan and Copeland 2000, p. 189]

‘Via the great pioneers of electronic computing…the 1930s analysis of computation led to the modern computing era. Who knows where a 21st-century overhaul of that classical analysis might lead.’ [Copeland, Dresner, Proudfoot and Shagrir 2016, p. 38]

‘Is there any limit to discrete computation, and more generally, to scientific knowledge?’ [Calude and Dinneen 1998, p. 1]

Final Remarks

‘Once upon on time, back in the golden age of the recursive function theory, computability was an absolute.’ [Sylvan and Copeland 2000, p. 189]

‘Via the great pioneers of electronic computing...the 1930s analysis of computation led to the modern computing era. Who knows where a 21st-century overhaul of that classical analysis might lead.’ [Copeland, Dresner, Proudfoot and Shagrir 2016, p. 38]

‘Is there any limit to discrete computation, and more generally, to scientific knowledge?’ [Calude and Dinneen 1998, p. 1]

‘“In breaking the Turing barrier, our knowledge of the world, and therefore our control of it, would be altered forever,” Professor Cooper added.’†

References


References