THEORETICAL FOUNDATIONS: MODELS OF COMPUTATION

The Sirius Machine and Limitations of Turing-like Machines

When uncountably infinite maps to countably infinite

Sirius Chan
Y12 CSSOC member

Turing computability and computability are essentially the same, the Church-Turing thesis states anything that can be computed intuitively can also be computed by a Turing machine. This text showcases a formal way for comparing infinite tape machines, and explains why all Turing-like machines are identical in behaviour, implying they can only compute Turing computable problems regardless of of their tape size.

The Turing machine is an abstract machine consisting of a read/write head and an infinitely long tape of cells - more specifically a tape of countably infinite cells. For satirical purposes we devise a *Sirius machine* with uncountably infinite cells. By inspection it should be able to compute a larger set of computable problems than the Turing machine for it has more tape, thus showing that the set of Turing computable problems is merely a subset of a larger set of computable problems. Or is it?

We will begin by comparing the two machines by inspection, then moving on to provide an intuitive explanation on the unobvious conclusion. Finally providing a formal way of proving the equivalence between the Turing machine and *Sirius machine*.

You will certainly be disappointed.

TURING-LIKE MACHINES

Turing-like machines can be emulated by a Turing machine and vice versa, implying they are identical characteristic-wise. The exact steps of determining whether a machine is Turing-like will be covered in later sections, for now we focus on the *Sirius machine*.

Storing data

Cells in a countably infinite tape can be counted off with the set of natural numbers $\mathbb{N}$, whereas that of an uncountably infinite tape cannot. Therefore there is data representable on the *Sirius machine* but not on the Turing machine.

More formally we say the countably infinite tape of a Turing machine can be indexed by the set of natural numbers $\mathbb{N}$, while the uncountably infinite tape of a *Sirius machine* can only be indexed by the set of real numbers $\mathbb{R}$.

Assume there exists a problem only representable by uncountably infinite cells, and computable by the *Sirius machine*. So by definition it is a computable problem to the *Sirius machine*, but not representable, therefore an uncomputable problem to the Turing machine.

Carrying out instructions

There is no difference between the *Sirius machine* and the Turing machine other than their tape size. They both include an FSM representing state, and follow the same idea of state transitions.

Such machines carry out transitions in sequence, implying there is a transition 1, transition 2, transition 3, $\ldots$. In other words, a Turing-like machine can only carry out countably infinite transitions.

Accessible storage size

Since a Turing-like machine can only access one cell per transition, the maximum number of cells it can access $|C|$ cannot be larger than countably infinity $\aleph_0$.

$$|C| \leq \aleph_0$$

Consider the cases

A. The machine terminates after a finite number of transitions.

B. The machine does not terminate after a finite number of transitions, but it repeatedly access a finite set of cells.

C. The machine does not terminate after a finite number of transitions, and it does not repeatedly access a finite set of cells.

Machine A and B have a finite accessible storage $|C| < \aleph_0$, whereas machine C has a countably infinite accessible storage $|C| = \aleph_0$. For any given tape of length l, only machine C will be able to determine whether $l < \aleph_0$ or $l \geq \aleph_0$.

Determining accessible storage size

More generally, for any machine to determine whether it is supplied with a tape of length $l < m$ or $l \geq m$, it must have accessible storage $|C| \geq m$. Where m is an arbitrary value to compare to l.

Therefore, for a machine to determine whether it is supplied with a tape of uncountably infinite $\aleph_1$ cells, it has to have an accessible storage $|C| \geq \aleph_1$. Any Turing-like machine, when supplied with a tape of uncountably infinite cells, will still behave as if it is supplied with a countably infinite tape. In which case there is no difference between the *Sirius machine* and the Turing machine.

The hypercomputer is a family of hypothetical models of computation that can solve non Turing-computable problems. However, a model of computation which utilises an uncountably infinite memory is yet to be devised.

DEFINING THE TURING-LIKE MACHINE

Even though there are uncountably infinite cells in a *Sirius machine*, only countably finite of which are visible to the machine. As a result it still cannot solve problems represented by uncountably infinite cells, and can be emulated with a Turing machine of countably infinite tape.

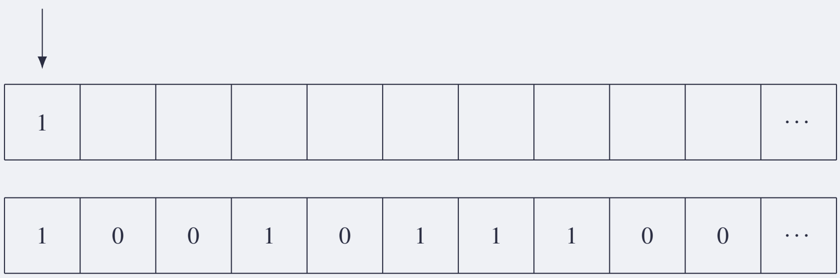
Tape mapping

To demonstrate there is no difference between the *Sirius machine* and the Turing machine, we can map the accessible tape of a *Sirius machine* to that of a Turing machine - which is a one-dimensional, infinitely long list.

A transition in a Turing machine is written as (S, W, P), where S is the next state, W is the content to write to the current cell, and P is the position of cursor after the transition.

The movement of cursor in a Turing machine can be complex and hard to visualise, another way to demonstrate the tape length of a Turing machine can be done by viewing the machine differently.

1. This representation of a Turing machine consists of the original tape with countably infinite cells, and a second empty countably infinite write tape. By properties of infinities, we know this machine also have countably infinite cells in total.
2. At its initial state, the content in its starting cell is copied to the first cell of the write tape.



3. To carry out transition (S, W, P)
   - Update the current state to S.
   - Write content W to current cell.
   - Move cursor right by one cell.
   - Copy the content in position P from the original tape to current cell.
4. If cell P in the original tape has already been accessed, its copied value may be modified. So instead of copying the unmodified value from the original tape, we copy the potentially modified value from the write tape. How the access record is stored is beyond the scope of this text, but it can be done using a third infinite tape.

This representation helps visualise properties of a Turing machine.

Determining accessible storage by tape mapping

- The position of the cursor $i - 1$ is the number of transitions the machine has gone through.
- The number of accesses to the original tape is the size of the accessible storage $|C|$.

$$i = |C| + |C_w|$$

Where $|C_w|$ is the number of copies from the write tape.

Again consider the cases

A. The machine terminates after a finite number of transitions:
$|C|$ is finite because i is finite.

B. The machine repeatedly access a finite set of cells:
$|C_w|$ grows at the same rate as i, where i is infinite but $|C|$ is finite.

C. The machine does not repeatedly access a finite set of cells:
$|C_w|$ does not grow at the same rate as i, where i is infinite and $|C|$ is infinite.

For any case, we know that $i \leq \aleph_0$, and $|C| \leq i$.

$$|C| \leq i \leq \aleph_0$$

Tape mappable implies Turing-like

By mapping the original tape to a one-dimensional write tape, we know that the accessible storage $|C|$ of a Turing machine is less than or equal to countably infinite $\aleph_0$, and the machine can therefore be emulated by a Turing machine. The same process can be repeated for a *Sirius machine* by replacing the original tape with a tape of uncountably infinite cells to show its maximum accessible storage $|C|$ is also countably infinite, which is left as an exercise for the reader.

Conclusion

Any infinite tape machine can be emulated with a Turing machine, and because $\aleph_0$ is the smallest infinity, all tape machines must be able to emulate a Turing machine. In other words, all Turing-like machines have identical properties despite the difference in their tape size.