Remark 1. The 2D cellular automata we constructed in the many-one reductions to prove undecidabilities of

- 2D nilpotency
- 2D reversibility
- 2D surjectivity

were all with the **von Neumann -neighborhood**. So these decision problems are undecidable among automata with the von Neumann -neighborhood.

Remark 2. Similarly in the one-dimensional case, the cellular automaton we constructed in the proof of undecidability of 1D nilpotency

- has radius-$\frac{1}{2}$ neighborhood $(0, 1)$, and
- has a **spreading state** q: any cell that has q in its neighborhood turns into state q.

So the decision problem 1D nilpotency is undecidable among radius-$\frac{1}{2}$ CA that have a spreading state.

Remark 3. Our undecidability results apply also among **higher dimensional CA**:

For any $d > 2$ a given two-dimensional CA A can be converted into a d-dimensional CA consisting of a $(d-2)$-dimensional grid of independent two-dimensional layers, each of which operates as A.

This d-dimensional CA is nilpotent, surjective or reversible if and only if A is nilpotent, surjective or reversible, respectively.

Remark 4. Each of the decision problems yields a Busy beaver -like, very rapidly growing function.

Define three functions

$$n, s, r \ : \ \mathbb{N} \longrightarrow \mathbb{N}$$

as follows:

For every $k \in \mathbb{N}$

- $n(k)$ is the largest t such that there is a nilpotent, one-dimensional, radius-$\frac{1}{2}$ cellular automaton G with k states, and a configuration c such that $G^t(c)$ is not the quiescent configuration.

Remark 4. Each of the decision problems yields a Busy beaver -like, very rapidly growing function.

Define three functions

$$n, s, r \ : \ \mathbb{N} \longrightarrow \mathbb{N}$$

as follows:

For every $k \in \mathbb{N}$

- $n(k)$ is the largest t such that there is a nilpotent, one-dimensional, radius-$\frac{1}{2}$ cellular automaton G with k states, and a configuration c such that $G^t(c)$ is not the quiescent configuration.
- $s(k)$ is largest t such that there is a non-surjective, two-dimensional CA that uses the von Neumann neighborhood and has k states, such that there is no orphan pattern of size $t \times t$.

Remark 4. Each of the decision problems yields a Busy beaver -like, very rapidly growing function.

Define three functions

$$n, s, r \ : \ \mathbb{N} \longrightarrow \mathbb{N}$$

as follows:

For every $k \in \mathbb{N}$

- $n(k)$ is the largest t such that there is a nilpotent, one-dimensional, radius-$\frac{1}{2}$ cellular automaton G with k states, and a configuration c such that $G^t(c)$ is not the quiescent configuration.
- $s(k)$ is largest t such that there is a non-surjective, two-dimensional CA that uses the von Neumann neighborhood and has k states, such that there is no orphan pattern of size $t \times t$.
- $r(k)$ is the largest t such that there is a reversible, two-dimensional CA G with k states and the von Neumann neighborhood such that the inverse of G is not obtained using the radius-t neighborhood.

Each of the functions n, s, r is well defined since its value for any k is the maximum among a finite number of positive integers.

Yet **no algorithm can compute an upper bound** for any of the functions: If some algorithm could produce for every given k a number t such that $n(k) \leq t$, $s(k) \leq t$ or $r(k) \leq t$ then this algorithm could be used to solve the associated decision problem 1D Nilpotency, 2D Surjectivity or 2D Reversibility.

(In each case such t would provide a bound on the size of instances that the corresponding semi-algorithm needs to check.)

We see that

- there are nilpotent CA that have configurations that survive very long time before becoming quiescent,
- there are non-surjective CA whose smallest orphan is very large, and
- there are reversible CA whose inverse CA have very large neighborhoods.

We see that

- there are nilpotent CA that have configurations that survive very long time before becoming quiescent,
- there are non-surjective CA whose smallest orphan is very large, and
- there are reversible CA whose inverse CA have very large neighborhoods.

While topology provides that every non-surjective CA has an orphan, computation theory provides that the size of a smallest orphan can be huge.

While topology provides that every injective CA has an inverse CA, computation theory provides that the size of the neighborhood of the inverse CA can be enormous.

Computational universality

General purpose computers are computationally universal: they can simulate any semi-algorithm if the semi-algorithm is encoded properly and is given to the computer as part of its input.

Note: It is very useful that computational universality is possible – otherwise we would have to build a new computer for each computational task!

For any languages $L \subseteq \Sigma^*$ and $K \subseteq \Delta^*$, let us denote

$$K \leq_m L$$

if K can be **many-one reduced** to L.

(In other words, there exists an algorithm that takes as input an arbitrary word $u \in \Delta^*$ and outputs a word $v \in \Sigma^*$ with the property that $u \in K$ if and only if $v \in L$.)

Language K is "at most as complex" as L:

- if L is recursive then K is also recursive.
- if L is recursively enumerable then K is also recursively enumerable.

Similarly for decision problems P and Q: we denote

$$Q \leq_m P$$

if Q can be many-one-reduced to P. Again,

- if P is decidable then Q is also decidable.
- if P is semi-decidable then Q is also semi-decidable.

A recursively enumerable language $L \subseteq \Sigma^*$ is **r.e.-complete** if

$$K \leq_m L$$

holds for **every** recursively enumerable language K.

Then the language L is "maximally complex" among recursively enumerable languages.

In the recursion theory terminology r.e-complete languages are called Σ_1^0**-complete** languages (w.r.t. many-one reductions).

Similarly, a semi-decidable decision problem P is **r.e.-complete** if

$$Q \leq_m P$$

holds for **every** semi-decidable decision problem Q.

Similarly, a semi-decidable decision problem P is **r.e.-complete** if

$$Q \leq_m P$$

holds for **every** semi-decidable decision problem Q.

Semi-algorithms for r.e.-complete decision problems are **computationally universal**. This means that a semi-algorithm A is universal iff for every semi-algorithm B there exists a conversion algorithm that converts an arbitrary input of B into an equivalent input of A.

Similarly, a semi-decidable decision problem P is **r.e.-complete** if

$$Q \leq_m P$$

holds for **every** semi-decidable decision problem Q.

Semi-algorithms for r.e.-complete decision problems are **computationally universal**. This means that a semi-algorithm A is universal iff for every semi-algorithm B there exists a conversion algorithm that converts an arbitrary input of B into an equivalent input of A.

There exist r.e.-complete decision problems (and languages).

Proposition. Decision problem Semi-algorithm halting is r.e.-complete.

Proof.

Many-one reducibility is a **transitive** relation:

$$R \leq_m P \text{ and } P \leq_m Q \quad \Longrightarrow \quad R \leq_m Q.$$

It follows for semi-decidable problems P and Q that if $P \leq_m Q$ and P is r.e.-complete then also Q is r.e.-complete.

Many-one reducibility is a **transitive** relation:

$$R \leq_m P \text{ and } P \leq_m Q \implies R \leq_m Q.$$

It follows for semi-decidable problems P and Q that if $P \leq_m Q$ and P is r.e.-complete then also Q is r.e.-complete.

All our undecidability proofs for tilings and cellular automata (also the proofs that were skipped) were obtained by a chain of many-one reductions from Semi-algorithm halting.

Corollary. The following decision problems are r.e.-complete: TM halting on blank tape, Finite tiling problem, Periodic tiling problem, 1D nilpotency and 2D Reversibility. The complements of the following are r.e.-complete: Tiling problem with the seed tile, Tiling problem, NW-deterministic tiling problem and 2D Surjectivity.

Any semi-algorithm for any of the problems in the corollary is computationally universal.

Let us call a Turing machine M **computationally universal** if the language it recognizes is r.e.-complete. (In other words, if the problem of determining whether M accepts a given input word w is r.e.-complete.)

Such machines exist because r.e.-complete languages exist, and every such language is recognized by some Turing machine.

Let us call a Turing machine M **computationally universal** if the language it recognizes is r.e.-complete. (In other words, if the problem of determining whether M accepts a given input word w is r.e.-complete.)

Such machines exist because r.e.-complete languages exist, and every such language is recognized by some Turing machine.

Remark. One can imagine many alternative notions of computational universality of Turing machines: One could say, for example, that M is universal if one of the problems below is r.e.-complete for M:

- Given two finite configurations α and β, does $\alpha \vdash^* \beta$?
  (A finite ID has finitely many non-blank cells.)
- Given a finite configuration α does $\alpha \vdash^* \beta$ for some accepting β ?
- Given a configuration α with an ultimately periodic tape content $\ldots uuu\, w\, vvv \ldots$, does $\alpha \vdash^* \beta$ for some accepting configuration β ?
- $\ldots$

These definitions of universality are not equivalent: A Turing machine may be universal in one sense without being universal in another sense!

Analogously to Turing machines, computational universality of a cellular automaton can be defined in **various non-equivalent ways**. The situation is even worse since there is no standard manner to encode the input in a CA, read the result, etc.

For this reason we never write a theorem simply stating that a certain CA is computationally universal, but rather it is always stated what is the r.e.-complete problem for the CA.

Using a straightforward simulation of a universal Turing machine we get the following:

Proposition. There exists a one-dimensional CA $G : S^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$ with a quiescent state $q \in S$ and a subset $F \subseteq S$ of final states such that the following decision problem is r.e. complete:

Instance: A q-finite initial configuration $c \in S^{\mathbb{Z}}$

Positive instance: $G^n(c)_k \in F$ for some $n \in \mathbb{N}, m \in \mathbb{Z}$.

(In other words, the input is encoded as a q-finite initial configuration, and the input is accepted iff the CA eventually has a final state $\in F$ in some cell.)

Proof. Let

$$M = (Q, \Gamma, \Sigma, \delta, q_0, q_a, q_r, B)$$

be a universal Turing machine (that is, $L(M)$ is r.e.-complete). We construct a 1D cellular automaton G with the state set

$$S = (Q \cup \{0\}) \times \Gamma$$

where $0 \notin Q$ is a new blank symbol (shown as blank in the drawings).

So configurations have two layers:

- Track 1 stores the current state $q \in Q$ of M in the current tape position of M. Other cells have the blank 0 on track 1.
- Track 2 stores the current tape content of M.

- - - - -

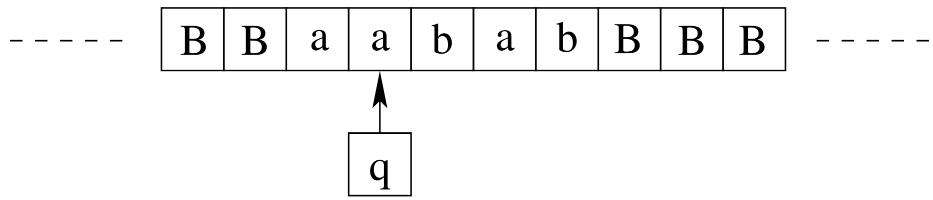
			q						
B	B	a	a	b	a	b	B	B	B

- - - - -

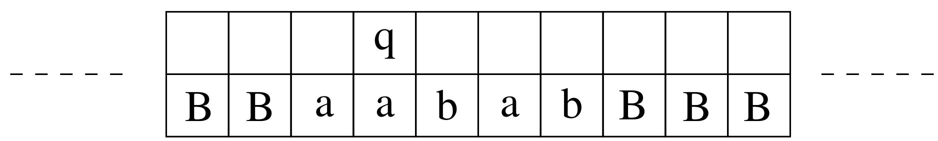
Let

$$E : Q \times \mathbb{Z} \times \Gamma^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$$

be the configuration **encoding function** that maps



to



More precisely, $E(q, i, t) = c$ where

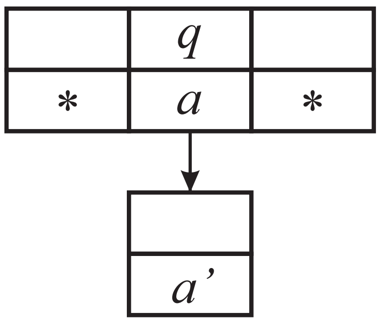
$$\forall j \in \mathbb{Z} \ : \ c(j) = \begin{cases} (q, t(j)) & \text{if } j = i, \\ (0, t(j)) & \text{if } j \neq i. \end{cases}$$

The CA local rule is designed so that the following diagram commutes:

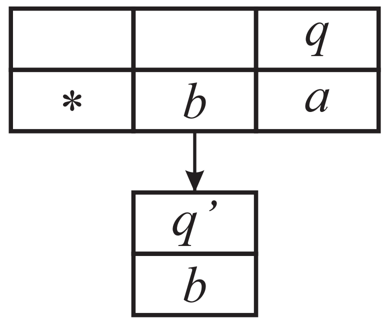
$$
\begin{array}{ccc}
Q \times \mathbb{Z} \times \Gamma^{\mathbb{Z}} & \xrightarrow{\ \vdash\ } & Q \times \mathbb{Z} \times \Gamma^{\mathbb{Z}} \\
\downarrow E & & \downarrow E \\
S^{\mathbb{Z}} & \xrightarrow{\ G\ } & S^{\mathbb{Z}}
\end{array}
$$

This means that: if the Turing machine changes (q, i, t) into (q', i', t') in k steps then the CA changes configuration $E(q, i, t)$ into $E(q', i', t')$ in k steps.

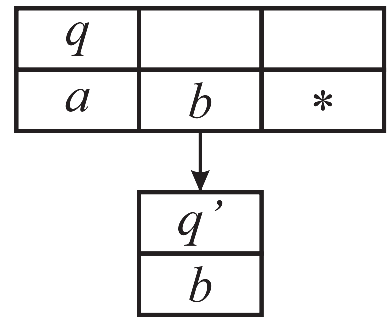
The CA uses radius-1 neighborhood and the local rule f that is defined as follows: $f(x, y, z) = y$ (i.e., no change) except in the following cases ($q \neq 0$):



$\delta(q, a) = (q', a', d)$

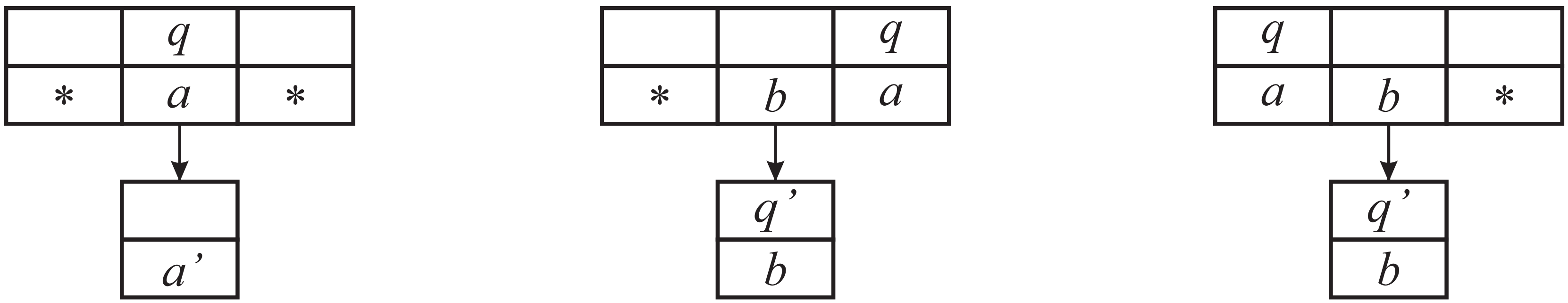


$\delta(q, a) = (q', a', -1)$



$\delta(q, a) = (q', a', +1)$

The CA uses radius-1 neighborhood and the local rule f that is defined as follows: $f(x, y, z) = y$ (i.e., no change) except in the following cases ($q \neq 0$):



$\delta(q, a) = (q', a', d)$ $\quad$ $\delta(q, a) = (q', a', -1)$ $\quad$ $\delta(q, a) = (q', a', +1)$

With this it is clear that

$$(q, i, t) \vdash (q', i', t') \quad \Longleftrightarrow \quad G(E(q, i, t)) = E(q', i', t'),$$

as required.

Let the quiescent state be

$$q = (0, B)$$

and let

$$F = \{q_a, x) \mid x \in \Gamma\}$$

be the set of states having the accepting state q_a of M on the first track.

Now it is easy to see that the decision problem

Instance: A q-finite initial configuration $c \in S^{\mathbb{Z}}$
Positive instance: $G^n(c)_k \in F$ for some $n \in \mathbb{N}, m \in \mathbb{Z}$.

is r.e. complete.

One can modify the above construction to obtain a one-dimensional CA G with radius-1 neighborhood that is computationally universal in the sense that the problem given is r.e.-complete:

- Does a given finite configuration c evolve into the quiescent configuration ?

One can modify the above construction to obtain a one-dimensional CA G with radius-1 neighborhood that is computationally universal in the sense that the problem given is r.e.-complete:

- Does a given finite configuration c evolve into the quiescent configuration ?
- For a fixed spreading state s, given a finite configuration c does there exist n such that $G^n(c)(0) = s$?

One can modify the above construction to obtain a one-dimensional CA G with radius-1 neighborhood that is computationally universal in the sense that the problem given is r.e.-complete:

- Does a given finite configuration c evolve into the quiescent configuration ?
- For a fixed spreading state s, given a finite configuration c does there exist n such that $G^n(c)(0) = s$?
- Is a given finite configuration c eventually fixed ?

One can modify the above construction to obtain a one-dimensional CA G with radius-1 neighborhood that is computationally universal in the sense that the problem given is r.e.-complete:

- Does a given finite configuration c evolve into the quiescent configuration ?
- For a fixed spreading state s, given a finite configuration c does there exist n such that $G^n(c)(0) = s$?
- Is a given finite configuration c eventually fixed ?
- Given two finite configurations c and e, does there exist $n \geq 0$ such that $G^n(c) = e$?

One can modify the above construction to obtain a one-dimensional CA G with radius-1 neighborhood that is computationally universal in the sense that the problem given is r.e.-complete:

- Does a given finite configuration c evolve into the quiescent configuration ?
- For a fixed spreading state s, given a finite configuration c does there exist n such that $G^n(c)(0) = s$?
- Is a given finite configuration c eventually fixed ?
- Given two finite configurations c and e, does there exist $n \geq 0$ such that $G^n(c) = e$?
- $\ldots$

Example. **Game-Of-Life** has been claimed to be universal in the sense that the following problem is r.e.-complete (where q is the state "dead"):

- Does a given q-finite configuration evolve into the q-uniform configuration ?

The proof was sketched in 1982 in

E.R.Berlekamp, J.H.Conway, R.K.Guy: Winning Ways for your Mathematical Plays.

(Filling the missing details is not obvious.)

Example. **Game-Of-Life** has been claimed to be universal in the sense that the following problem is r.e.-complete (where q is the state "dead"):

- Does a given q-finite configuration evolve into the q-uniform configuration ?

The proof was sketched in 1982 in

E.R.Berlekamp, J.H.Conway, R.K.Guy: Winning Ways for your Mathematical Plays.

(Filling the missing details is not obvious.)

A detailed existing construction shows that the following problem is r.e.-complete for GOL:

- Does a given q-finite configuration evolve into a configuration that contains a fixed specific finite pattern p ?

Example. Computational universality of **Rule 110** was proved in

Matthew Cook: Universality in Elementary Cellular Automata. Complex Systems. 15: 1–40 (2004).

It was shown that there are fixed finite words u, v such that it is r.e.-complete to decide for a given finite word w whether the initial configuration $\ldots uuu\, w\, vvv \ldots$ is eventually periodic.

It is also r.e.-complete to decide for given words w and p whether $\ldots uuu\, w\, vvv \ldots$ evolves into configuration that contains word p.