ITT8040 — Cellular Automata
Lecture 6

Silvio Capobianco

Institute of Cybernetics at TUT

April 19, 2013

Turing machines

A (deterministic) Turing machine is specified by:

- a finite set of states Q;
- three special states $q_I, q_A, q_R \in Q$ called the initial, accepting, and rejecting state;
- a finite tape alphabet Γ;
- a finite input alphabet $\Sigma \subset \Gamma$;
- a blank tape symbol $b \in \Gamma \setminus \Sigma$; and
- a transition function $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1, +1\}$.

The transition function specifies the behavior of a read-write head on a bi-infinite tape, in the following way:

if you are in current state and read current symbol,
then take next state, write next symbol, and move by one block

We require $\delta(q_A, \gamma) = (q_A, \gamma, +1)$ and $\delta(q_R, \gamma) = (q_R, \gamma, +1)$.

Turing machines (cont.)

An instantaneous description of a configuration of a Turing machine is a triple $(q, i, t) \in Q \times \mathbb{Z} \times \Gamma^{\mathbb{Z}}$ where:

- q is the current state
- i is the current position of the head
- t is the global state of the tape

The next configuration (q', i', t') is defined straightforwardly:

$$\text{if } \delta(q, t(i)) = (q', \gamma, x)$$
$$\text{then } i' = i + x, \ t'(i) = \gamma, \ t'(j) = t(j) \text{ for } i \neq j$$

If t_w is the tape with w in positions 1 to $|w|$ and blank elsewhere, then the machine accepts w if $(q_I, 1, t_w) \to^* (q_A, i, t)$.
It rejects w if either $(q_I, 1, t_w) \to^* (q_R, i, t)$ or never halts.

Turing machines as a formal model of semi-algorithms

We say that a Turing machine T and a semi-algorithm S are equivalent if they recognize the same language, that is, for every input x:

- T accepts x if and only if S returns "yes" on x; and
- T rejects x if and only if S returns "no" on x or does not halt on x.

Then there is an algorithm that, given an arbitrary Turing machine T, constructs an equivalent semi-algorithm S.

Theorem: There exists an algorithm that, given an arbitrary semi-algorithm S, constructs an equivalent Turing machine T.

Turing's halting problem

Consider the following problem:

- given an arbitrary Turing machine T,
- determine whether or not T halts on the empty tape.

Theorem:
Turing's halting problem is semi-decidable but not decidable.

- Given A and w, construct the semi-algorithm:
$$\mathrm{B(u) = A(w)}$$
- Construct a Turing machine T equivalent to B.
- Then T halts on the empty tape if and only if A halts on w.
- This is a reduction of semi-algorithm halting to Turing's halting problem.
  As the former is undecidable, so is the latter.

The seeded tiling problem

Consider the following problem:

- given a finite tile set $\mathcal{T} = (T, N, R)$ and a special seed tile $s \in T$,
- determine whether or not $\mathcal{T}$ admits a valid tiling t such that $t(0,0) = s$

The complement of this problem is semi-decidable:

- for every $n \geq 1$:
- if every tiling of $n \times n$ squares containing s is not valid then return "no"

The seeded tiling problem is undecidable

Let T be a Turing machine.
Consider the tile set from Figures 27 and 28.

- Tiles from Figure 27 represent ongoing computations.
- Tiles from Figure 28 represent initial empty conditions (including a special “start” tile) and a blank.

Then the following can be seen:

the tile set admits a valid tiling with the “start” tile as seed
if and only if
the Turing machine does not halt from the empty tape

This is a reduction of Turing’s halting problem to (the complement of) the seeded tiling problem.
As the former is undecidable, so is the latter.

The finite tiling problem

Consider the following problem:

- given a finite tile set $\mathcal{T} = (T, N, R)$ and a special blank tile $B \in T$ such that $(B, \ldots, B) \in R$,
- determine whether or not $\mathcal{T}$ admits a finite nontrivial valid tiling t

This problem is semi-decidable:

- for every $n \geq 1$:
- if there is a valid non-trivial tiling of an $n \times n$ square with blank border then return "yes"

The finite tiling problem is undecidable

Let T be a Turing machine.
Consider the tile set from Figures 27 and 29.

- Tiles from Figure 27 represent ongoing computations.
- Tiles from Figure 29 represent space-time bounds of the computation, plus a blank.

Then the following can be seen:

the tile set admits a valid finite nontrivial tiling
if and only if
the Turing machine halts on the empty tape

This is a reduction of Turing's halting problem to the finite tiling problem.
As the former is undecidable, so is the latter.

The tiling problem

Consider the following problem:

- given a finite tile set $\mathcal{T} = (T, N, R)$,
- determine whether or not $\mathcal{T}$ admits a valid tiling t

Theorem: (Berger, 1966)
The tiling problem is undecidable, even for Wang tiles.

Recall that the complement of the tiling problem is semi-decidable.

Two more undecidable problems

The NW-deterministic tiling problem:

- given a NW-deterministic tile set $\mathcal{T} = (T, N, R)$,
- determine whether or not $\mathcal{T}$ admits a valid tiling t

is undecidable: its complement is semi-decidable.

The periodic tiling problem:

- given a tile set $\mathcal{T} = (T, N, R)$,
- determine whether or not $\mathcal{T}$ admits a valid periodic tiling t

is semi-decidable: its complement is not semi-decidable.

Nilpotent cellular automata

A cellular automaton $A = (S, d, N, f)$ with quiescent state q and global function G is nilpotent if every configuration ultimately evolves into the quiescent configuration.

This is the same as satisfiying the following condition:
There exists $n \geq 1$ such that the n-th iteration G^n sends every configuration into the quiescent configuration.

- Let c be a rich configuration containing every possible pattern.
- Then G^n makes every configuration quiescent if and only if it makes q quiescent.

Nilpotency is thus semi-decidable.

Nilpotency of 2D cellular automata is undecidable

Let T be a finite set of Wang tiles.

- Set $S = T \sqcup \{q\}$.
- Let f leave the middle tile unchanged if the tiling is correct, and turn it to q otherwise.
- This CA is nilpotent if and only if T does not admit a valid tiling.

We have reduced the tiling problem to (non-)nilpotency of 2D CA. As the former is undecidable, so is the latter.

Nilpotency of 1D cellular automata is undecidable

Let T be a NW-deterministic finite set of Wang tiles.

- Set $S = T \sqcup \{q\}$ and

$$f(x,y) = \begin{cases} z & \text{if } \begin{array}{|c|c|} \hline & y \\ \hline x & z \\ \hline \end{array} \text{ is a match}\,, \\ q & \text{otherwise}\,. \end{cases}$$

- This CA is nilpotent if and only if T does not admit a valid tiling.

We have reduced the NW-deterministic tiling problem to (non-)nilpotency of 1D CA.
As the former is undecidable, so is the latter.

Reversibility of 2D cellular automata is undecidable

Let T be a finite tile set.

Let D be a finite tile set with the plane filling property.

- Let $S = T \times D \times \{0, 1\}$.
- Define the local update rule as follows:
  - if both tiling components are valid then XOR the bit component with that of the follower;
  - otherwise do nothing
- The resulting CA is reversible if and only if T does not admit a valid tiling.

This reduces the tiling problem to 2D CA (non-)reversibility.