ITT8040 Cellular Automata
Solutions to Assignment 6

Exercise 1

Let $\mathcal{T} = (T, N, R)$ be a tile set. Consider the CA $A = (T \times \{0, 1\}, 2, N, f)$ where f flips the bit component of the cell at $\vec{n}$ if and only if the tiling component is *not* valid at $\vec{n}$. Then the CA has a fixed point if and only if the tile set admits a valid tiling. If the 2D CA fixed point problem was decidable, then so would be the tiling problem: which is not the case.

Exercise 2

We can construct the Turing machine T by considering the following recursive definition for the sum of the elements of a list of natural numbers:

```
sum [] = 0
sum (x:xs) = x + sum xs
```

Construct the sequence $(x_1, \ldots, x_n)$ on the tape of the Turing machine as follows: $x_1 + 1$ dashes, one blank, $x_2 + 1$ dashes, one blank, $\ldots$, $x_n + 1$ dashes, with the head of the machine at the beginning of the first element, if the list is nonempty, and in an arbitrary position otherwise. Then two consecutive blanks mean the end of the sequence of numbers, while a single blank followed by a dash means moving from an element of the sequence to the next one. On the other hand, as the number x is represented by $x+1$ consecutive dashes, the sum of x and y will be represented by $x + y + 1 = (x + 1) + (y + 1) - 2$ dashes: this can be performed by filling the blank between them with a dash, then remove two dashes from the right of the newly obtained sequence.

A Turing machine that computes the sum of a list of integers can thus be constructed by specifying the following rules:

$(q_I, 0)$	$\mapsto$	$(q_1, 0, +)$	check if end of sequence
$(q_I, 1)$	$\mapsto$	$(q_I, 1, +)$	go to end of element
$(q_1, 0)$	$\mapsto$	$(q_2, 1, -)$	end of list
$(q_1, 1)$	$\mapsto$	$(q_I, 1, +)$	new element in list
$(q_2, 0)$	$\mapsto$	$(q_3, 0, -)$	check if anything to merge
$(q_3, 0)$	$\mapsto$	$(q_A, 0, +)$	nothing to merge
$(q_3, 1)$	$\mapsto$	$(q_4, 1, +)$	something to merge
$(q_4, 0)$	$\mapsto$	$(q_5, 1, +)$	merge the sequences
$(q_5, 0)$	$\mapsto$	$(q_6, 0, -)$	end of sequence reached
$(q_5, 1)$	$\mapsto$	$(q_5, 1, +)$	go to end of sequence
$(q_6, 1)$	$\mapsto$	$(q_7, 0, -)$	clear last dash
$(q_7, 1)$	$\mapsto$	$(q_8, 0, -)$	clear second-last dash
$(q_8, 1)$	$\mapsto$	$(q_8, 1, -)$	go to beginning of merged sequence
$(q_8, 0)$	$\mapsto$	$(q_3, 0, -)$	check if anything else to merge

The Turing machine employs initial state q_I and auxiliary state q_1 to reach the end of the sequence: when it does, it writes a dash. After this, it uses states q_2 and q_3 to check whether there are any sequences of dashes before the one it has just written: if there is none, then the machine halts. If there is another sequence before, the machine first merges them by turning the blank between them into a dash, then removes two dashes from the end of the sequence. The merging is repeated until the beginning of the original sequence is reached again, which marks the end of the computation.