

EXERCISE SHEET 11: LOCAL SIMULATION OF CELLULAR AUTOMATA

Exercise 1 (Surjective Simulation). *Attempt to prove that a surjective CA can only globally simulate surjective cellular automata. At which point does the argument fail? What if we required the decoder to have a full domain?*

Exercise 2 (Comparing CAs in different dimensions). *Can you think of a way to “lift” a d -dimensional CA \mathcal{A} into a $d + 1$ -dimensional CA \mathcal{A}' such that \mathcal{A} is conjugate to a subsystem of \mathcal{A}' ? Use this to suggest how global simulation could be extended between cellular automata in different dimensions.*

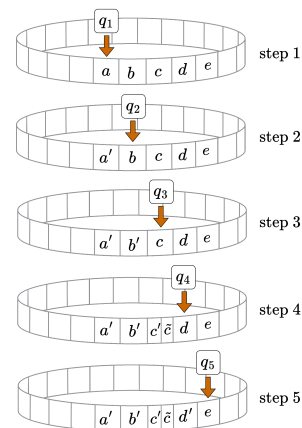
Exercise 3 (Admissible cyclic tag systems). *Let \mathcal{T}_1 be a cyclic tag system defined by an appendant list $(\alpha_0, \alpha_1, \dots, \alpha_{z-1})$ with each $\alpha_i \in \{Y, N\}^*$. Show that there exists a cyclic tag system \mathcal{T}_2 which can simulate it and whose appendant list only consists of words which are either empty or whose length is a multiple of six.*

Exercise 4 (Clockwise Turing machine). *We will consider one more alternative example. A clockwise Turing machine differs from a standard Turing machine in the following aspects:*

- the tape is finite and circular
- head moves only clockwise on the tape
- before moving to the next cell, the head can either write a single symbol into the current cell, or split the current cell in two and write two new symbols; afterwards, the head moves clockwise, skipping all symbols it wrote.

The computational process defined by such a machine is illustrated below.

- the tape is finite and circular
- head moves only clockwise on the tape
- the head can write symbols in two ways
- either it writes a single symbol in a standard way (such as $a \rightarrow a'$ in step 1)
- or it writes two symbols, splitting the current cell, then moving clockwise and skipping both symbols it wrote (such as $c \rightarrow c'\tilde{c}$ in step 3)



Show that for any classical Turing machine, which cannot write the blank symbol, there exists a clockwise Turing machine that can locally simulate it.

Exercise 5 (Universal Turing machine must have periodic points). *Let \mathcal{T}_1 and \mathcal{T}_2 be Turing machines such that \mathcal{T}_2 locally simulates \mathcal{T}_1 via an encoder \mathcal{E} , decoder \mathcal{D} and delay function τ whose time-complexities are bound by some computable function $\beta(\cdot)$. Prove the following.*

- (i) For every $c \in \text{Conf}(\mathcal{T}_1)$ the set $\mathcal{D}^{-1}(c)$ is finite.

- (ii) Show that if $c \in \text{Conf}(\mathcal{T}_1)$ is a fixed point of \mathcal{T}_1 ; i.e., $F_{\mathcal{T}_1}(c) = c$ then the set $\mathcal{D}^{-1}(c)$ contains a periodic point x of \mathcal{T}_2 ; i.e., $F_{\mathcal{T}_2}(x)^L = x$ for some $L \in \mathbb{N}^+$.
- (iii) Show that if \mathcal{U} is a locally universal Turing machine then for every $N \in \mathbb{N}$ there exists an L such that \mathcal{U} contains at least N disjoint periodic orbits of period L .

Exercise 6. Let $\mathcal{A} = (S^{\mathbb{Z}^d}, F)$ be a CA with a quiescent state $q \in S$ and denote $\mathbf{q} = q^{\mathbb{Z}^d}$.

\mathcal{A} is **nilpotent**, if there exists $N \in \mathbb{N}$ such that $F^N(c) = \mathbf{q}$ for all $c \in S^{\mathbb{Z}^d}$.
 \mathcal{A} is **asymptotically nilpotent** if $\lim_{n \rightarrow \infty} F^n(c) = \mathbf{q}$ for all $c \in S^{\mathbb{Z}^d}$.

- (i) Can a nilpotent CA be locally universal?
- (ii) Prove: \mathcal{A} is nilpotent $\iff \Lambda(F) := \bigcap_{n \in \mathbb{N}} F^n(S^{\mathbb{Z}^d}) = \{\mathbf{q}\}$
- (iii) Can an asymptotically nilpotent CA be locally universal?