

2. Natural Computing - Cellular Automata

21

Slides 1-24 of CA-Kari.pdf

* Turing machine: tape, state, control unit

transition function δ :

initial state $\rightarrow (A, 0) \mapsto (B, 1, 1)$
 $(A, 1) \mapsto (B, 1, -1)$
 blank symbol $(B, 0) \mapsto (A, 1, -1)$
 $(B, 1) \mapsto (q_a, 1, 1)$
 \uparrow accepting state

blank tape:

```

      A
... 0 0 0 0 0 0 0 0 ...
      B
├   ... 0 0 0 0 1 0 0 0 ...
      A
├   ... 0 0 0 0 1 1 0 0 ...
      B
├   ... 0 0 0 0 1 1 0 0 ...
      A
├   ... 0 0 0 1 1 1 0 0 ...
      B
├   ... 0 0 1 1 1 1 0 0 ...
      q_a
├   ... 0 0 1 1 1 1 0 0
  
```

Every recursively enumerable language is recognized by a Turing machine

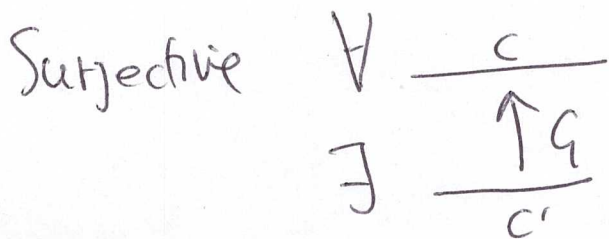
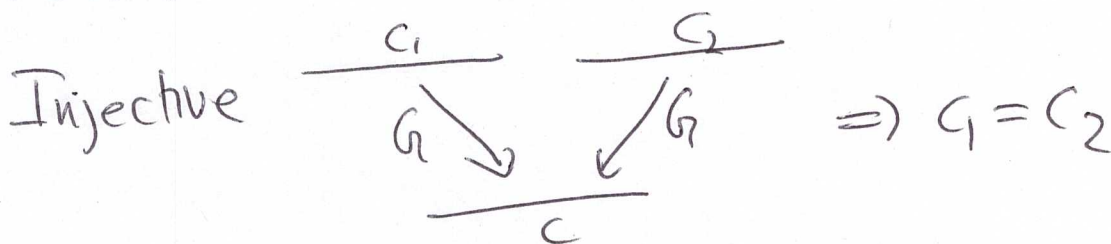
Game-of-Life can simulate any Turing machine !!

Rule 110 is also computationally universal!

Slides 25-35

* Cellular Automata - Definitions

Slides 36-51



Bijective = Injective + Surjective

Periodic: $\exists k, c \quad G^k(c) = c \rightarrow W2$

Fixed point $G(c) = c$

Orbit: $c \rightarrow G(c) \rightarrow G^2(c) \rightarrow \dots$

Nilpotent $\exists n, c \quad G^n(S^{\mathbb{Z}^d}) = \{c\} \rightarrow W1$

CA $\left\{ \begin{array}{l} \text{Shift function } \sigma_i : N = \{e_i\}, f = \text{identity} \\ \text{Translation } \tau_y : N = \{-y\}, f = \text{id} \\ \text{In } \tau_y(c) \text{ cell } \underline{x+y} \text{ has state } c(\underline{x}) \end{array} \right.$

$\tau_y \circ G = G \circ \tau_y$: All CA commute with translations.

* Topology

Introduce a metric (distance measure) between configurations

→ pattern $g: D \rightarrow S$
finite domain $\subseteq \mathbb{Z}^d$

$$\text{Cyl}(g) = \{ c \in S^{\mathbb{Z}^d} : c(x) = g(x) \text{ for } x \in D \}$$

- 1) A union of open balls
 - 2) Open balls are cylinders
- } cylinders form a basis of the topology

→ can talk about convergence, limits

→ " " " continuous functions

Th 1: (G is a CA) iff (G is continuous & G commutes with translations)

Cor 1: (G is reversible) iff (G is a bijection)

Slide 61, Slide 70, 71

* Balance

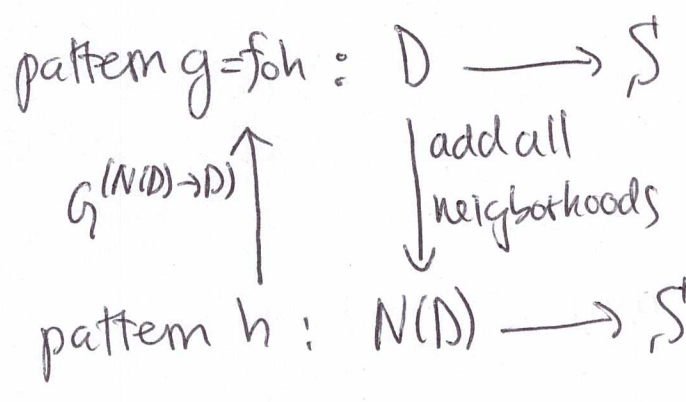
Rule 110 } 5 neighborhood patterns → 1
 } 3 " " " → 0

Ex 2

$$e: \quad \begin{matrix} 1 & 2 & 3 & & k \\ \underline{xxx} & \underline{xxx} & \underline{xxx} & \dots & \underline{xxx} \end{matrix} \rightsquigarrow 3^k \text{ choices}$$
$$c: \quad \begin{matrix} \downarrow & \downarrow & \downarrow & & \downarrow \\ 0xx & 0x & 0xx & \dots & xx0 \end{matrix} \rightsquigarrow (2^2)^{k-1} = 4^{k-1} \text{ choices}$$

Some choice of c does not have a pre-image → not surjective

$$Q = (d, S, N, f)$$



$$g(x) = f(h(x+x_1), \dots, h(x+x_m))$$

where

$$N = \{x_1, \dots, x_m\}$$

g Orphan: \exists pattern $h : Q^{(N(D) \rightarrow D)} / h = g$

A configuration c that contains an orphan pattern is
 Garden-of-Eden (\exists preimage of c)

Prop 2 CA has orphan iff CA is non-surjective

Surjective CA $\Rightarrow \forall$ patterns g of D $\# \{h : Q^{(N(D) \rightarrow D)} / h = g\} = \text{const}$
 ($D = \text{singleton}$)
 \Rightarrow Each state in the local rule table appears equal # times

Garden-of-Eden

g is stable if $f(g, \dots, g) = g$ (in figures g corresponds to the stable cell)

$c \in S^{\mathbb{Z}^d}$ is g -finite iff $\{x \in \mathbb{Z}^d : c(x) \neq g\}$ is finite

$\mathcal{Z}_g \subseteq S^{\mathbb{Z}^d}$ is the subset of g -finite configurations

If $c \in \mathcal{Z}_g$ then $Q(c) \in \mathcal{Z}_g \Rightarrow$ Define $Q_F : \mathcal{Z}_g \rightarrow \mathcal{Z}_g$

In other words: G_F is the global update rule for ~~cellular~~ configurations with a finite # non-blank cells.

We call G preinjective iff G_F is injective

(clearly, injective \Rightarrow pre-injective)

Rule 110 is not surjective: e.g., 01010 is an orphan

EX4

$c: \dots 000xxx \dots xxx00 \dots \approx 2^{5k-2} = 32^k/4$ choices

$G(c)$

00	x x x x x	x x x x x	x x x x x
	1	2	k

orphan cannot occur

\approx at most $(2^5 - 1)^k = 31^k$ choices

There must be 2 o-finite configurations with the same image

Th 3: (G is pre-injective) \Rightarrow (G is surjective)

for injective \equiv bijective \equiv reversible !

Also Th 4 (G is surjective) \Rightarrow (G is pre-injective) (Slide 93)

Rule 110

$c_1 = \dots 000011010000 \dots$

$c_2 = \dots 000010110000 \dots$

$G(c_1) = G(c_2)$
(not pre-injective)

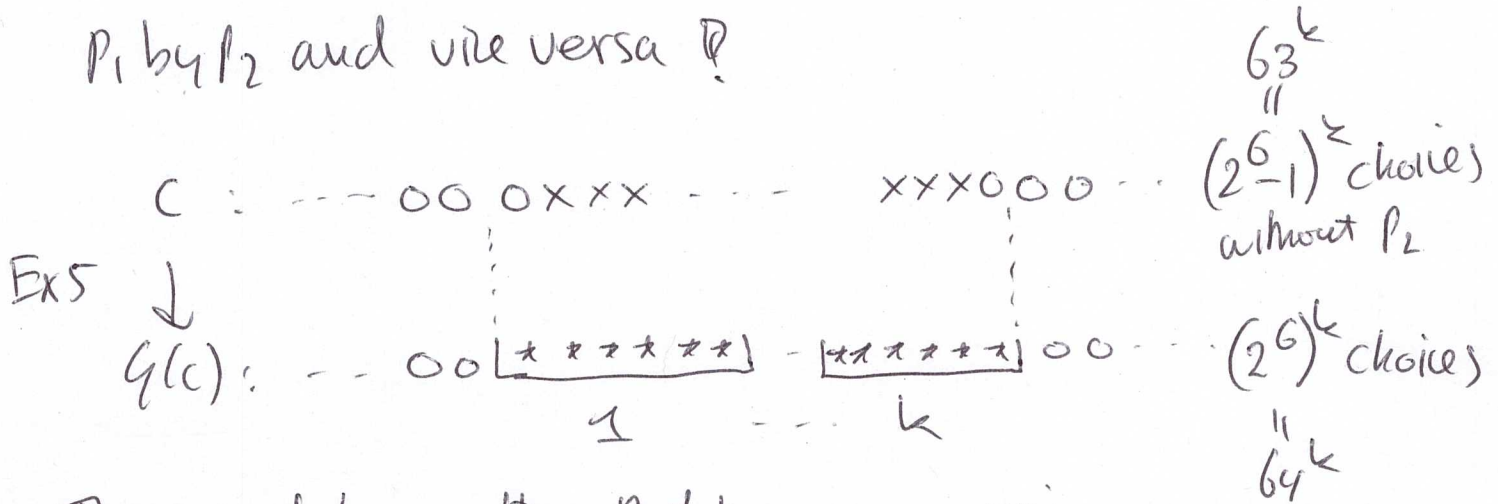
$p_1 = 011010 \rightarrow 1111$

$p_2 = 010110 \rightarrow 1111$

p_1 and p_2 only differ in the center two positions

Rule 110 has radius-1 neighborhood

So, without affecting $q(c)$ we may replace any pattern P_1 by P_2 and vice versa!



There must be a pattern that has no pre-image

q is not surjective

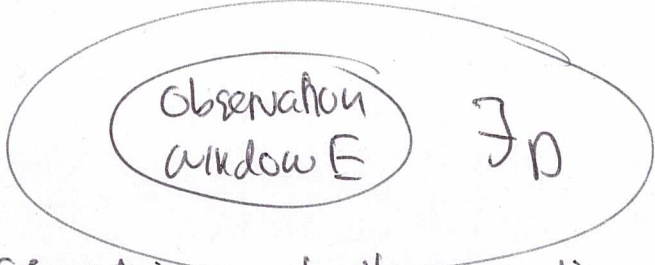
Algorithmic questions

Decidability in 1D : \exists alg. that on input always correctly determines (after finite computation) if the input has the desired property

Chaotic behavior

1) Means sensitivity to small perturbations

Equicontinuous:



States outside D do not affect states in E

Equicontinuous iff CA is eventually periodic

2) Mixing of configuration space is another property associated to chaos

(A is transitive) $\Leftrightarrow \forall$ cylinders $U, V \exists_n G^n(U) \cap V \neq \emptyset$

\Uparrow

(\Rightarrow sensitivity to initial cond.)

(A is mixing) \Leftrightarrow -----

\Uparrow

$\exists_n \forall_{k \geq n} G^k(U) \cap V \neq \emptyset$

positively expansive

Next class: (176)

- Read Sections 1 & 2 in detail
- Glance over Sect. 3 (definitions are repeated)
- Read Section 4
- Read 5: glance over 5.2
- Read 6.1, 6.2 in detail
- Glance over 6.3 - d
- Glance over 9 (introduces (17))

What does universal computation / machine mean to you?
 Have we come to a final answer with respect to defining
 universality?

Is the concept of universality important, worth studying?
 What do you need to understand about universality?