Standardizing the set of states and the neighborhood of asynchronous cellular automata

Thomas Worsch
worsch@kit.edu

Karlsruhe Institute of Technology
Institute for Cryptography and Security

Classical result

Alvy Ray Smith III.
Cellular automata complexity trade-offs.
Information and Control, 18:466–482, 1971.

For synchronous CA it is true that

- each CA with $k = |Q|$ states can be simulated by a CA with $2 = |\{0, 1\}|$ states
- each CA with $N = \{-r, \ldots, -1, 0, 1, \ldots r\}$ can be simulated by a CA with $N = \{-1, 0, 1\}$.

Asynchronous CA (ACA)

- w.l.o.g. consider one-dimensional CA

- $\mathbf{Z}$ set of cells
- Q set of states
- N neighborhood
- f local transition function $Q^N \to Q$

- $A \subseteq \mathbf{Z}$ "activity set"
- F_A global transition function $Q^{\mathbf{Z}} \to Q^{\mathbf{Z}}$

$$F_A(c)(i) = \begin{cases} f(c_{i+N}) & \text{iff } i \in A \\ c(i) & \text{iff } i \notin A \end{cases}$$

- F global transition relation $F = \bigcup_{A \subseteq \mathbf{Z}} F_A$

Outline

1. An idea by Lee et al.
2. 2 states are enough
3. Neighborhood radius 1 is enough
4. Generalization of the idea by Lee et al.

Idea by Lee et al.

how to simulate a synchronous CA on an asynchronous CA

- Lee/Adachi/Peper/Morita (2004), Schumacher (2012)
- synchronous CA $C_s = (Q_s, N_s, f_s)$
- asynchronous CA $C_a = (Q_a, N_a, f_a)$

$$N_a = N_s \cup -N_s \cup \{0\}$$

$$Q_a = Q_{\mathrm{B}} \cup Q_{\mathrm{T}} \cup Q_{\mathrm{E}}$$

where

$$\begin{aligned} Q_{\mathrm{B}} &= \{(q, \mathrm{B}) \mid q \in Q_s\} && \textit{begin states} \\ Q_{\mathrm{T}} &= \{(q, q') \mid q, q' \in Q_s\} && \textit{transitional states} \\ Q_{\mathrm{E}} &= \{(\mathrm{E}, q') \mid q' \in Q_s\} && \textit{end states} \end{aligned}$$

- given $c_s \in Q_s^{\mathbf{Z}}$
- define $c_a \in Q_a^{\mathbf{Z}}$ $\qquad \forall i \in \mathbf{Z} : c_a(i) = (c_s(i), \mathrm{B})$

An idea by Lee et al. (2)

local transition function for the asynchronous CA

$\mathtt{B} \to \mathtt{T}$: **if** all neighbors are in $Q_\mathtt{B} \cup Q_\mathtt{T}$
then transition $(q, \mathtt{B}) \mapsto (q, f_s(q_1, \ldots, q_k))$
where q_i first components of the neighboring states

$\mathtt{T} \to \mathtt{E}$: **if** all neighbors are in $Q_\mathtt{T} \cup Q_\mathtt{E}$
then transition $(q, q') \mapsto (\mathtt{E}, q')$

$\mathtt{E} \to \mathtt{B}$: **if** all neighbors are in $Q_\mathtt{E} \cup Q_\mathtt{B}$
then transition $(\mathtt{E}, q') \mapsto (q', \mathtt{B})$

otherwise no change

An idea by Lee et al. (3)

example: shift left

$t = 0$:	$(\_,\mathtt{B})$	$(\_,\mathtt{B})$	$(\_,\mathtt{B})$	$(\_,\mathtt{B})$	$(\#,\mathtt{B})$	$(\_,\mathtt{B})$
			$\Downarrow$	$\Downarrow$		
$t = 1$:	$(\_,\mathtt{B})$	$(\_,\mathtt{B})$	$(\_,\_)$	$(\_,\#)$	$(\#,\mathtt{B})$	$(\_,\mathtt{B})$
	$\Downarrow$			$\downarrow$	$\Downarrow$	$\Downarrow$
$t = 2$:	$(\_,\_)$	$(\_,\mathtt{B})$	$(\_,\_)$	$(\_,\#)$	$(\#,\_)$	$(\_,\_)$
		$\Downarrow$	$\downarrow$	$\Downarrow$	$\Downarrow$	
$t = 3$:	$(\_,\_)$	$(\_,\_)$	$(\_,\_)$	$(\mathtt{E},\#)$	$(\mathtt{E},\_)$	$(\_,\_)$
		$\Downarrow$	$\Downarrow$		$\downarrow$	$\Downarrow$
$t = 4$:	$(\_,\_)$	$(\mathtt{E},\_)$	$(\mathtt{E},\_)$	$(\mathtt{E},\#)$	$(\mathtt{E},\_)$	$(\mathtt{E},\_)$
		$\downarrow$	$\Downarrow$		$\Downarrow$	
$t = 5$:	$(\_,\_)$	$(\mathtt{E},\_)$	$(\_,\mathtt{B})$	$(\mathtt{E},\#)$	$(\_,\mathtt{B})$	$(\mathtt{E},\_)$
	$\Downarrow$			$\Downarrow$		
$t = 6$:	$(\mathtt{E},\_)$	$(\mathtt{E},\_)$	$(\_,\mathtt{B})$	$(\#,\mathtt{B})$	$(\_,\mathtt{B})$	$(\mathtt{E},\_)$
	$\Downarrow$	$\Downarrow$				$\Downarrow$
$t = 7$:	$(\_,\mathtt{B})$	$(\_,\mathtt{B})$	$(\_,\mathtt{B})$	$(\#,\mathtt{B})$	$(\_,\mathtt{B})$	$(\_,\mathtt{B})$

Theorem

Each ACA with $k = |Q_a| \geq 3$ states can be simulated by an ACA with $2 = |\{0, 1\}|$ states.

From k states to 2 states (1)

Encoding of one state

- "one-hot" encoding
- represent one state as

marker						old state					new state				
0	1	1	1	1	0	o_1	o_2	$\cdots$	o_k	x_{E}	n_1	n_2	$\cdots$	n_k	x_{B}

- the marker cannot appear anywhere else
- use large neighborhood
  each cell can observe all neighboring segments

From k states to 2 states (2)

Transitions

marker	old state					new state					corresponds to
		o_i	o_j					n_j			
011110	0	1	0	0	0	0	0	0	0	B	$(q_i, \mathtt{B})$
011110	0	1	0	0	0	0	0	1	0	B	
011110	0	1	0	0	0	0	0	1	0	0	(q_i, q_j)
011110	0	1	0	0	0	0	0	1	0	0	(q_i, q_j)
011110	0	1	0	0	E	0	0	1	0	0	
011110	0	0	0	0	E	0	0	1	0	0	$(\mathtt{E}, q_j)$
011110	0	0	0	0	E	0	0	1	0	0	$(\mathtt{E}, q_j)$
011110	0	0	0	0	E	0	0	1	0	B	
011110	0	0	1	0	E	0	0	1	0	B	
011110	0	0	1	0	E	0	0	0	0	B	
011110	0	0	1	0	0	0	0	0	0	B	$(q_j, \mathtt{B})$

From k states to 2 states (3)

Transitions

- for the simulation of *synchronous* CA
  - use Lee's idea

- for the simulation of *asynchronous* CA
  - start $\mathtt{B} \to \mathtt{T}$ transitions
    only if *all* neighboring segments represent an unmodified $\mathtt{B}$ state
  - do not care about neighboring segments afterwards

Optimization

segments of length $O(\log |Q|)$

- encode each state using $\log |Q| + O(1)$ bits
- represent each “logical” bit as `01` resp. `10`
  - use `00` indicating “undefined”
  - useful for checking progress of state copying during $\mathtt{E} \to \mathtt{B}$ transitions

1. An idea by Lee et al.
2. 2 states are enough
3. Neighborhood radius 1 is enough
4. Generalization of the idea by Lee et al.

Theorem

Each ACA with $N = \{-r, \ldots, -1, 0, 1, \ldots r\}$ can be simulated by an ACA with $N = \{-1, 0, 1\}$.

Construction

given ACA C

- construct ACA C'
  each cell has additional *activity bit a*
  - if $a = 1$: active cell will use original f of C
  - if $a = 0$: active cell will not change its state
- construct ACA C''
  - apply to C' construction by Smith (1971) for the synchronous case
  - no need to copy the a bits from neighbors
- construct C'''
  - apply to C'' the construction to make the local transition function robust for asychronous updating
- construct C''''
  - "plug in" local rules for computing a from data present in $N = \{-1, 0, 1\}$
  - *such that any arbitrary subset of cells can have its activity bits set*

Generalization of the idea

- extend the set of states further

$$Q_a = Q_{\mathtt{B}} \cup Q_{\mathtt{R}} \cup Q_{\mathtt{A}} \cup Q_{\mathtt{T}} \cup Q_{\mathtt{E}}$$

- where for example

$$\begin{aligned}
Q_{\mathtt{B}} &= \{(q, \mathtt{B}) \mid q \in Q\} && \times \{\perp\} && \times \{\perp\} \\
Q_{\mathtt{R}} &= \{(q, \mathtt{R}) \mid q \in Q\} && \times \{0,1\}^m && \times \{\perp\} \\
Q_{\mathtt{A}} &= \{(q, \mathtt{A}) \mid q \in Q\} && \times \{0,1\}^m && \times \{0,1\} \\
Q_{\mathtt{T}} &= \{(q, q') \mid q, q' \in Q\} && \times \{\perp\} && \times \{\perp\} \\
Q_{\mathtt{E}} &= \{(\mathtt{E}, q') \mid q' \in Q\} && \times \{\perp\} && \times \{\perp\}
\end{aligned}$$

"auxiliary" bits "activity" bit

Generalization of the idea (2)

local transition function

for $A \subseteq \mathbf{Z}$ realize corresponding activity bits *for example* as follows

	1	2	3	4	5	6	7	8	9
	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
even	$\perp$	0	$\perp$	0	$\perp$	0	$\perp$	0	$\perp$
even, A	$\perp$	0	$\perp$	1	$\perp$	0	$\perp$	0	$\perp$
odd	0	0	0	1	0	0	0	0	0
even	0	$0'$	0	$1'$	0	$0'$	0	$0'$	0
odd, A	0	$0'$	1	$1'$	0	$0'$	1	$0'$	0
even	0	$0''$	1	$1''$	0	$0''$	1	$0''$	0
odd	$0''$	$0''$	$1''$	$1''$	$0''$	$0''$	$1''$	$0''$	$0''$

Summary and Outlook

- ACA with k states can be simulated by ACA with 2 states (using larger N)
- ACA with N_r can be simulated by ACA with N_1 (using larger Q)

Open problems:

- different plugins for computing activity bits
- "speed-up" by constant factor for ACA?

Summary and Outlook

- ACA with k states can be simulated by ACA with 2 states (using larger N)
- ACA with N_r can be simulated by ACA with N_1 (using larger Q)

Open problems:

- different plugins for computing activity bits
- “speed-up” by constant factor for ACA?

Thank you very much for your attention!