Extended Two-Way Ordered Restarting Automata for Picture Languages

Friedrich Otto[1] František Mráz[2]

[1]Universität Kassel, Kassel, Germany

[2]Charles University, Prague, Czech Republic

LATA 2014

Madrid

March 10–14, 2014

1. Introduction

The restarting automaton models the linguistic technique of analysis by reduction (Jančar et. al., 1995).

Various classes of formal languages have been characterized by certain types of restarting automata:

REG	:	R(1)-automata (Mráz, 2001)
REG	:	det-RR(1) (Reimann, 2007)
DCFL	:	det-mon-R(R)(W)(W) (Jančar et. al., 1999)
CFL	:	mon-R(R)WW-automata (Jančar et. al., 1999)
CRL	:	det-R(R)WW-automata (Niemann et. al., 1998)
GCSL	:	wmon-R(R)WW-automata (Jurdziński et. al., 2004)

Many studies have extended grammars and automata from word languages to picture languages, e.g.:

- 4-way finite automata (Blum, Hewitt 1967),
- isometric array grammars (Rosenfeld 1971),
- matrix grammars (Siromoney et. al. 1972),
- tiling systems and automata (Giammarresi, Restivo 1992),
- Sudoku-deterministically recognizable languages (Borchert, Reinhardt 2007).

The class REC of recognizable picture languages has been identified as a central class:

- various nice characterizations and
- good closure properties,
- but it contains NP-complete languages, that is, NP-complete membership problems (Lindgren et. al. 1998).

Many studies have extended grammars and automata from word languages to picture languages, e.g.:

- 4-way finite automata (Blum, Hewitt 1967),
- isometric array grammars (Rosenfeld 1971),
- matrix grammars (Siromoney et. al. 1972),
- tiling systems and automata (Giammarresi, Restivo 1992),
- Sudoku-deterministically recognizable languages (Borchert, Reinhardt 2007).

The class REC of recognizable picture languages has been identified as a central class:

- various nice characterizations and
- good closure properties,
- but it contains NP-complete languages, that is, NP-complete membership problems (Lindgren et. al. 1998).

Many studies have extended grammars and automata from word languages to picture languages, e.g.:

- 4-way finite automata (Blum, Hewitt 1967),
- isometric array grammars (Rosenfeld 1971),
- matrix grammars (Siromoney et. al. 1972),
- tiling systems and automata (Giammarresi, Restivo 1992),
- Sudoku-deterministically recognizable languages (Borchert, Reinhardt 2007).

The class REC of recognizable picture languages has been identified as a central class:

- various nice characterizations and
- good closure properties,
- but it contains NP-complete languages, that is, NP-complete membership problems (Lindgren et. al. 1998).

Quest for two-dimensional automata with the following properties:

- intuitive model, that is, easy way of designing algorithms (automata) for interesting languages,
- membership problems decidable in polynomial time,
- restricted to word languages, only the regular languages should be accepted,
- nice closure properties.

Several models have been proposed recently:

- the restarting tiling automaton (Průša, Mráz, CIAA 2012)
- the Sgraffito automaton (Průša, Mráz, DLT 2012)
- the deterministic 2-dimensional 3-way ordered restarting automaton (Mráz, Otto, SOFSEM 2014)

Here: the determ. 2-dim. extended 2-way ordered restarting automaton (det-2D-x2W-ORWW-automaton).

Quest for two-dimensional automata with the following properties:

- intuitive model, that is, easy way of designing algorithms (automata) for interesting languages,
- membership problems decidable in polynomial time,
- restricted to word languages, only the regular languages should be accepted,
- nice closure properties.

Several models have been proposed recently:

- the restarting tiling automaton (Průša, Mráz, CIAA 2012)
- the Sgraffito automaton (Průša, Mráz, DLT 2012)
- the deterministic 2-dimensional 3-way ordered restarting automaton (Mráz, Otto, SOFSEM 2014)

Here: the determ. 2-dim. extended 2-way ordered restarting automaton (det-2D-x2W-ORWW-automaton).

Quest for two-dimensional automata with the following properties:

- intuitive model, that is, easy way of designing algorithms (automata) for interesting languages,
- membership problems decidable in polynomial time,
- restricted to word languages, only the regular languages should be accepted,
- nice closure properties.

Several models have been proposed recently:

- the restarting tiling automaton (Průša, Mráz, CIAA 2012)
- the Sgraffito automaton (Průša, Mráz, DLT 2012)
- the deterministic 2-dimensional 3-way ordered restarting automaton (Mráz, Otto, SOFSEM 2014)

Here: the determ. 2-dim. extended 2-way ordered restarting automaton (det-2D-x2W-ORWW-automaton).

2. Deterministic 3-Way ORWW-Automata

A picture P over Σ is a finite two-dimensional array of symbols from Σ.

$\mathrm{row}(P)$ ($\mathrm{col}(P)$) denotes the number of rows (columns) of P,

$P(i,j)$ is the symbol at position (i,j), $1 \leq i \leq \mathrm{row}(P)$, $1 \leq j \leq \mathrm{col}(P)$.

By $\Sigma^{m,n}$ we denote the set of all pictures of size $m \times n$ over Σ,

and $\Sigma^{*,*}$ is the set of all pictures over Σ.

Let $\mathcal{S} = \{\vdash, \dashv, \top, \bot, \#\}$ be a set of five special markers (sentinels).

In order to enable an automaton to detect the border of P easily, we define the boundary picture $\widehat{P}$ over $\Sigma \cup \mathcal{S}$ of size $(m+2) \times (n+2)$:

2. Deterministic 3-Way ORWW-Automata

A picture P over Σ is a finite two-dimensional array of symbols from Σ.

$\mathrm{row}(P)$ ($\mathrm{col}(P)$) denotes the number of rows (columns) of P,

$P(i,j)$ is the symbol at position (i,j), $1 \leq i \leq \mathrm{row}(P)$, $1 \leq j \leq \mathrm{col}(P)$.

By $\Sigma^{m,n}$ we denote the set of all pictures of size $m \times n$ over Σ,

and $\Sigma^{*,*}$ is the set of all pictures over Σ.

Let $\mathcal{S} = \{\vdash, \dashv, \top, \bot, \#\}$ be a set of five special markers (sentinels).

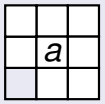
In order to enable an automaton to detect the border of P easily, we define the boundary picture $\widehat{P}$ over $\Sigma \cup \mathcal{S}$ of size $(m+2) \times (n+2)$:

$\#$	$\top$	$\top$	$\cdots$	$\top$	$\top$	$\#$
$\vdash$						$\dashv$
$\vdots$			P			$\vdots$
$\vdash$						$\dashv$
$\#$	$\bot$	$\bot$	$\cdots$	$\bot$	$\bot$	$\#$

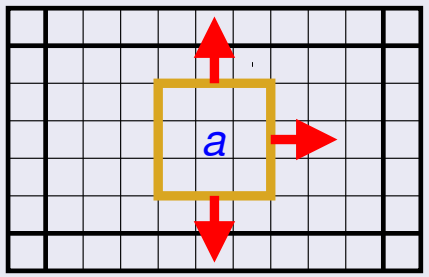
Definition 1

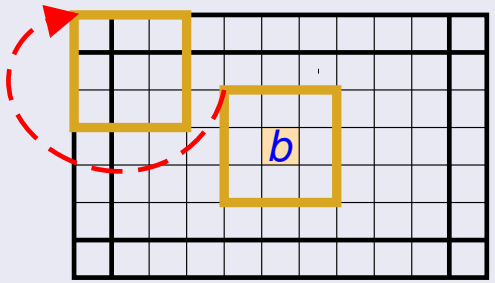
Deterministic two-dimensional three-way ordered RWW-automaton $M = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$:

- Q *is a finite set of states containing the initial state* q_0,
- Σ *is a finite input alphabet,* Γ *is a finite tape alphabet containing* Σ *such that* $\Gamma \cap \mathcal{S} = \emptyset$
- $>$ *is a partial ordering on* Γ
- δ *is the transition function,*

  for each $C =$ 

  it holds at most one of the following:
  - $\delta(q, C) = (q', \mathrm{R})$ – *move right step*
  - $\delta(q, C) = (q', \mathrm{D})$ – *move down step*
  - $\delta(q, C) = (q', \mathrm{U})$ – *move up step*
  - $\delta(q, C) = b$ *and* $a > b$ – *rewrite and restart*





Theorem 2

$\mathcal{L}(\text{det-2D-3W-ORWW}) \subseteq \text{DTIME}((\text{size}(P))^2)$.

Theorem 3 (M., O., SOFSEM 2014)

$\mathcal{L}(\text{det-2D-3W-ORWW}) \cap \Sigma^{1,*} = \text{REG}(\Sigma)$, *that is, the* det-2D-3W-ORWW*-automaton only accepts regular word languages.*

Theorem 4 (M., O., SOFSEM 2014)

The det-2D-3W-ORWW*-automaton can simulate the deterministic Sgraffito automaton, which in turn is known to be able to simulate*

- *(alternating) four-way finite automata,*
- *deterministic four-way one-marker automata,*
- *and to accept all sudoku-deterministically recognizable languages.*

Theorem 2

$\mathcal{L}$(det-2D-3W-ORWW) $\subseteq$ DTIME$((\text{size}(P))^2)$.

Theorem 3 (M., O., SOFSEM 2014)

$\mathcal{L}$(det-2D-3W-ORWW) $\cap\, \Sigma^{1,*} = \text{REG}(\Sigma)$, *that is, the* det-2D-3W-ORWW*-automaton only accepts regular word languages.*

Theorem 4 (M., O., SOFSEM 2014)

The det-2D-3W-ORWW*-automaton can simulate the deterministic Sgraffito automaton, which in turn is known to be able to simulate*

- *(alternating) four-way finite automata,*
- *deterministic four-way one-marker automata,*
- *and to accept all sudoku-deterministically recognizable languages.*

Theorem 2

$\mathcal{L}(\text{det-2D-3W-ORWW}) \subseteq \text{DTIME}((\text{size}(P))^2)$.

Theorem 3 (M., O., SOFSEM 2014)

$\mathcal{L}(\text{det-2D-3W-ORWW}) \cap \Sigma^{1,*} = \text{REG}(\Sigma)$, *that is, the* det-2D-3W-ORWW*-automaton only accepts regular word languages.*

Theorem 4 (M., O., SOFSEM 2014)

The det-2D-3W-ORWW*-automaton can simulate the deterministic Sgraffito automaton, which in turn is known to be able to simulate*

- *(alternating) four-way finite automata,*
- *deterministic four-way one-marker automata,*
- *and to accept all sudoku-deterministically recognizable languages.*

Remarks

- The det-2D-3W-ORWW-automata clearly favour vertical over horizontal movements. Accordingly, $\mathcal{L}$(det-2D-3W-ORWW) is not closed under transposition.

- It could happen that a det-2D-3W-ORWW-automaton M does not terminate on some input picture, as it may get stuck on a column, just moving up and down.

To avoid this, it is required explicitly that M halts on all input pictures!

This can be realized by providing a simple pattern, e.g.,
$\text{up}^* - \text{down}^* - \text{up}^* - \text{down}^*$, such that on each column,
the sequence of up and down movements must fit this pattern,
or one could use an external counter that counts the number of uninterrupted up and down movements on a column.

Remarks

- The det-2D-3W-ORWW-automata clearly favour vertical over horizontal movements. Accordingly, $\mathcal{L}$(det-2D-3W-ORWW) is not closed under transposition.

- It could happen that a det-2D-3W-ORWW-automaton M does not terminate on some input picture, as it may get stuck on a column, just moving up and down.

To avoid this, it is required explicitly that M halts on all input pictures!

This can be realized by providing a simple pattern, e.g., $\text{up}^* - \text{down}^* - \text{up}^* - \text{down}^*$, such that on each column, the sequence of up and down movements must fit this pattern, or one could use an external counter that counts the number of uninterrupted up and down movements on a column.

Remarks

- The det-2D-3W-ORWW-automata clearly favour vertical over horizontal movements. Accordingly, $\mathcal{L}$(det-2D-3W-ORWW) is not closed under transposition.

- It could happen that a det-2D-3W-ORWW-automaton *M* does not terminate on some input picture, as it may get stuck on a column, just moving up and down.

To avoid this, it is required explicitly that *M* halts on all input pictures!

This can be realized by providing a simple pattern, e.g., $\text{up}^* - \text{down}^* - \text{up}^* - \text{down}^*$, such that on each column, the sequence of up and down movements must fit this pattern, or one could use an external counter that counts the number of uninterrupted up and down movements on a column.

3. Deterministic Extended 2-Way ORWW-Automata

Definition 5

A deterministic two-dimensional extended two-way ordered RWW-automaton (det-2D-x2W-ORWW*-automaton*) *is given through a 7-tuple*

$$M = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >),$$

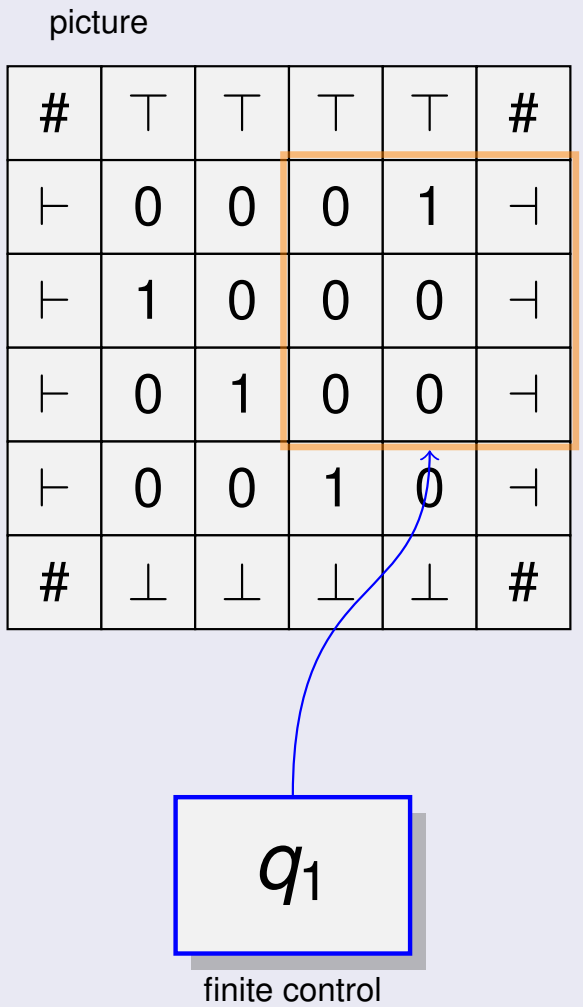
where all components are defined as for det-2D-3W-ORWW*-automata. However, the set of possible head movements is restricted to*

$$\mathcal{H} = \{\mathrm{R}, \mathrm{D}\},$$

where the move-right and move-down steps are extended as follows:

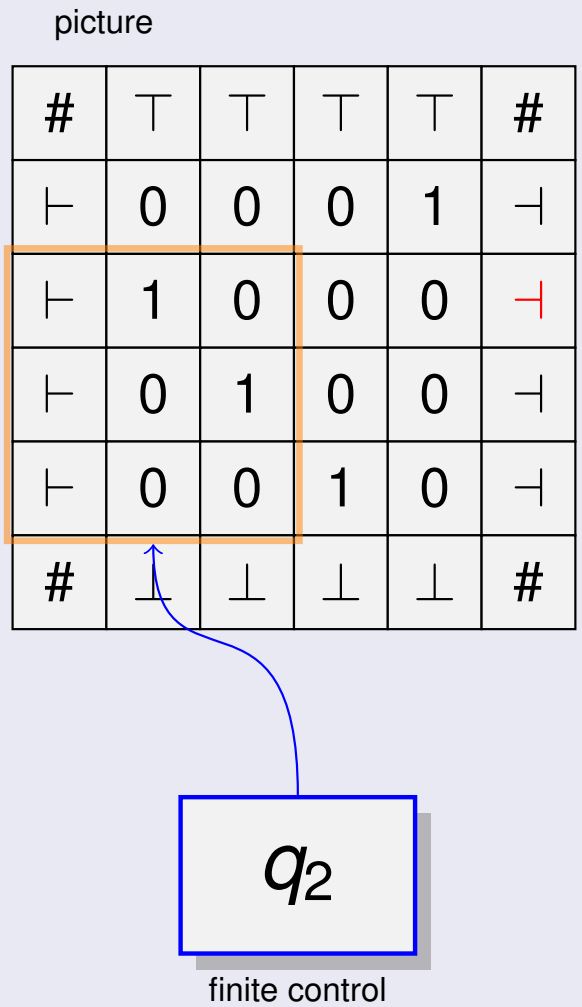
Definition 5 (cont.)

Extended Move-Right Step:



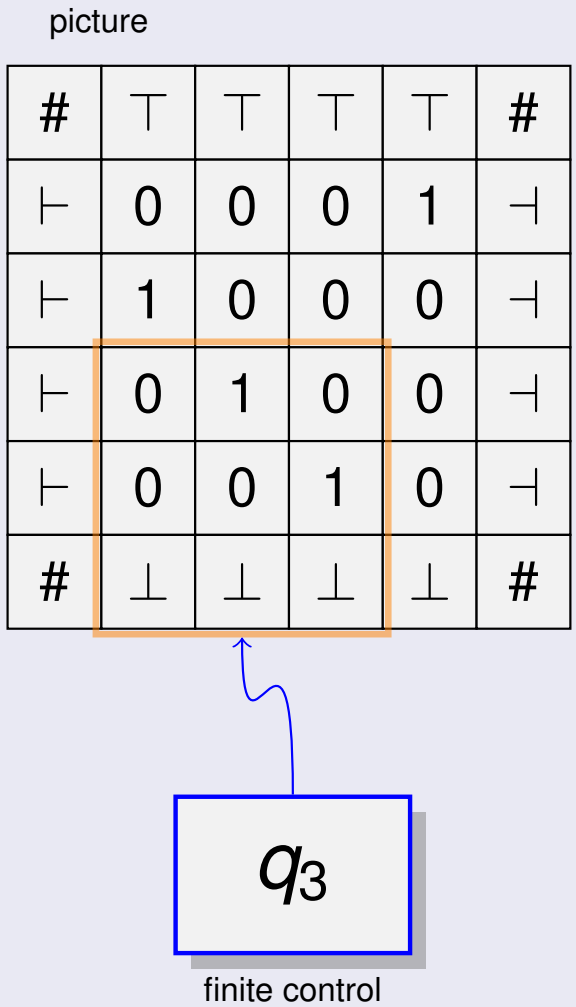
Definition 5 (cont.)

Extended Move-Right Step:



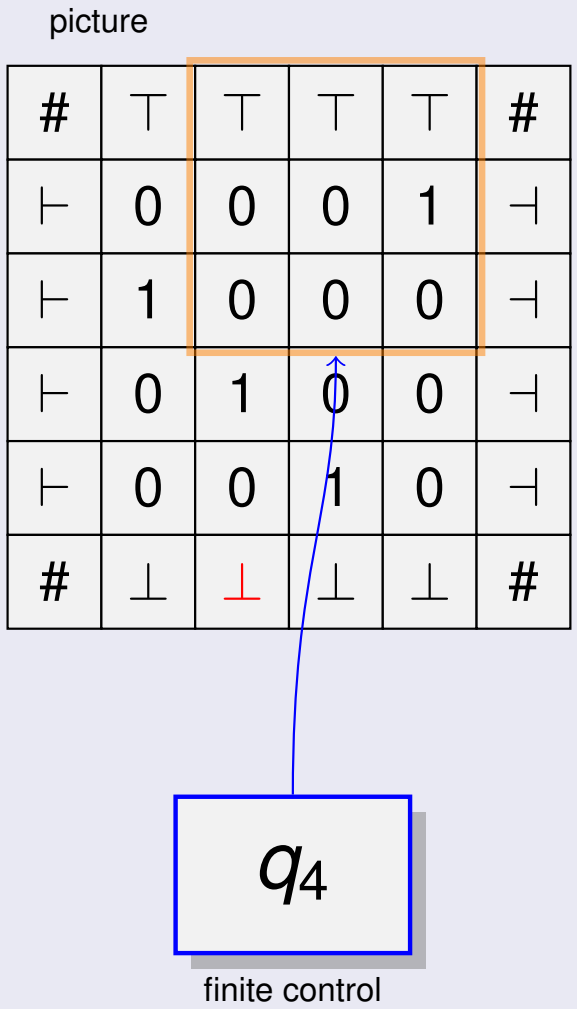
Definition 5 (cont.)

Extended Move-Down Step:



Definition 5 (cont.)

Extended Move-Down Step:



Definition 5 (cont.)

In any cycle, M can only use extended move-right or extended move-down steps, but not both!

M is a stateless det-2D-x2W-ORWW-automaton (stl-det-2D-x2W-ORWW-automaton) if it has just a single state. For such an automaton, the components Q and q_0 are suppressed.

Corollary 6

When restricted to one-dimensional input, then the det-2D-x2W-ORWW*-automaton just accepts the regular word languages. This also holds for the stateless variant.*

Theorem 7

$\mathcal{L}(\text{det-2D-x2W-ORWW}) \subseteq \text{DTIME}((\text{size}(P))^3)$.

Definition 5 (cont.)

In any cycle, M can only use extended move-right or extended move-down steps, but not both!

M is a stateless det-2D-x2W-ORWW-automaton (stl-det-2D-x2W-ORWW-automaton) if it has just a single state. For such an automaton, the components Q and q_0 are suppressed.

Corollary 6

When restricted to one-dimensional input, then the det-2D-x2W-ORWW*-automaton just accepts the regular word languages. This also holds for the stateless variant.*

Theorem 7

$\mathcal{L}(\text{det-2D-x2W-ORWW}) \subseteq \text{DTIME}((\text{size}(P))^3)$.

Definition 5 (cont.)

In any cycle, M can only use extended move-right or extended move-down steps, but not both!

M is a stateless det-2D-x2W-ORWW-automaton (stl-det-2D-x2W-ORWW-automaton) if it has just a single state. For such an automaton, the components Q and q_0 are suppressed.

Corollary 6

When restricted to one-dimensional input, then the det-2D-x2W-ORWW*-automaton just accepts the regular word languages. This also holds for the stateless variant.*

Theorem 7

$\mathcal{L}(\text{det-2D-x2W-ORWW}) \subseteq \text{DTIME}((\text{size}(\text{P}))^3)$.

Example 1

Let $\Sigma = \{0, 1\}$, and let

$$L_{\text{perm}} = \{\, P \in \Sigma^{*,*} \mid \text{row}(P) = \text{col}(P) \geq 1, \text{ each row and each column contains exactly one symbol 1} \,\}.$$

We describe a stl-det-2D-x2W-ORWW-automaton M_{perm} for L_{perm}.

Let

$$\Gamma = \Sigma \cup \{0', 1', 0'_1, 0'', 1'', 0''_1\},$$

and let

$$1 > 0 > 1' > 0' > 0'_1 > 1'' > 0'' > 0''_1.$$

The automaton M_{perm} proceeds as follows:

Example 1

Let $\Sigma = \{0, 1\}$, and let

$$L_{\text{perm}} = \{\, P \in \Sigma^{*,*} \mid \text{row}(P) = \text{col}(P) \geq 1, \text{ each row and each column contains exactly one symbol 1} \,\}.$$

We describe a stl-det-2D-x2W-ORWW-automaton M_{perm} for L_{perm}.

Let

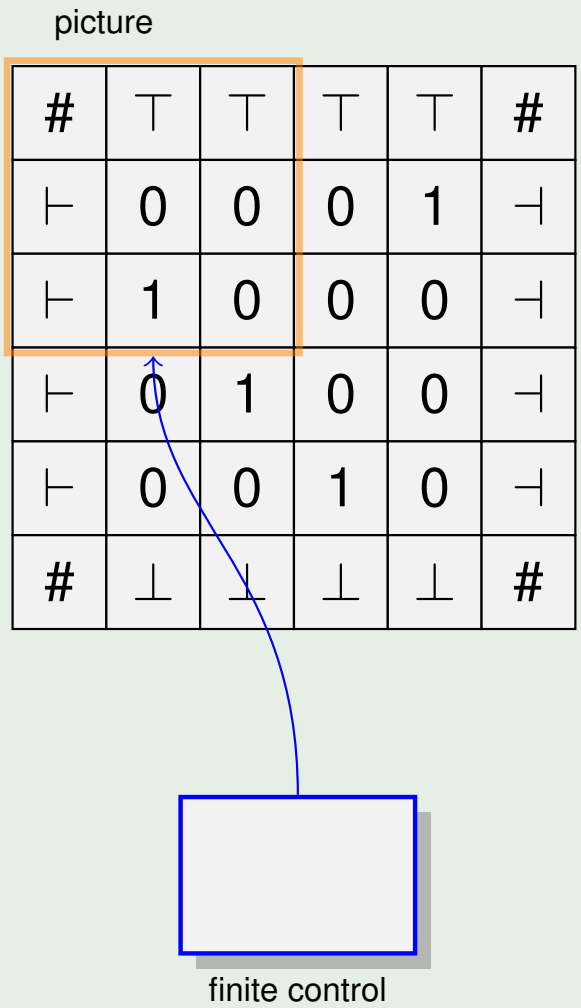
$$\Gamma = \Sigma \cup \{0', 1', 0'_1, 0'', 1'', 0''_1\},$$

and let

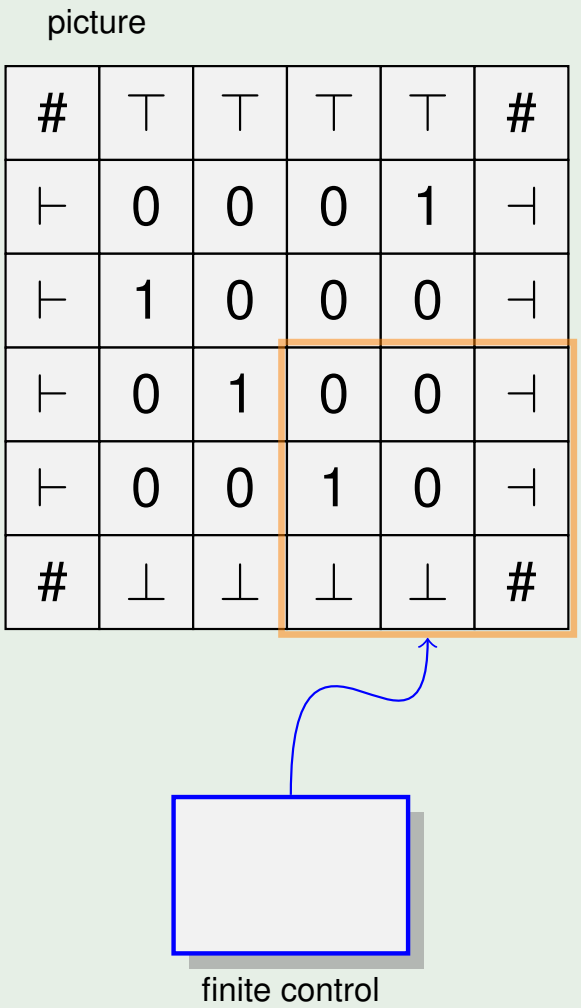
$$1 > 0 > 1' > 0' > 0'_1 > 1'' > 0'' > 0''_1.$$

The automaton M_{perm} proceeds as follows:

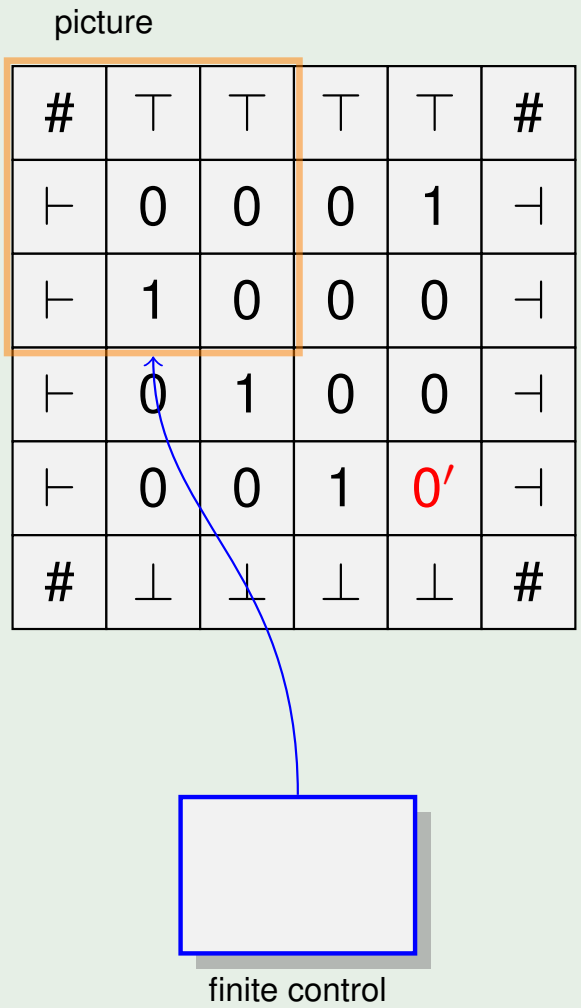
Example 1 (cont.)



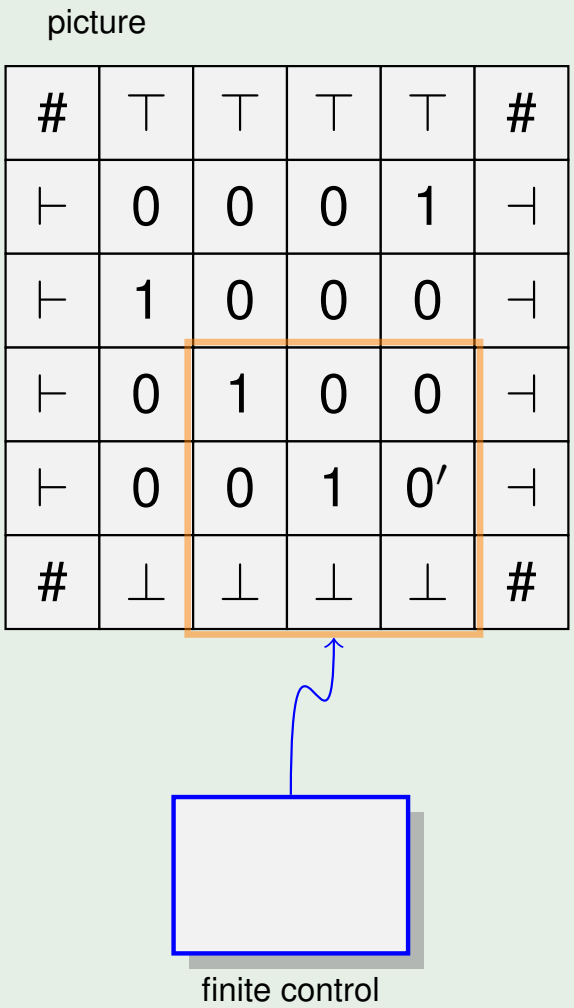
Example 1 (cont.)



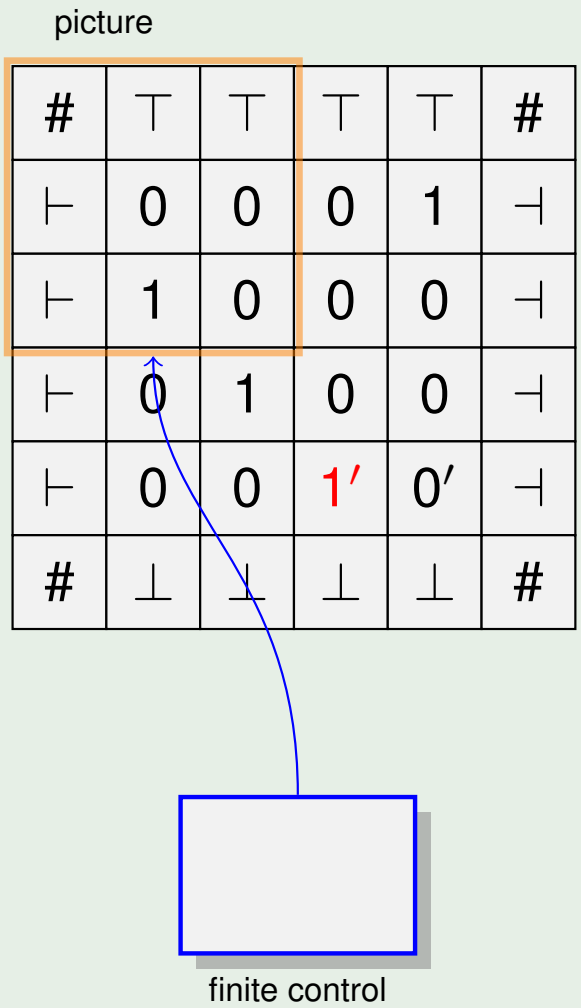
Example 1 (cont.)



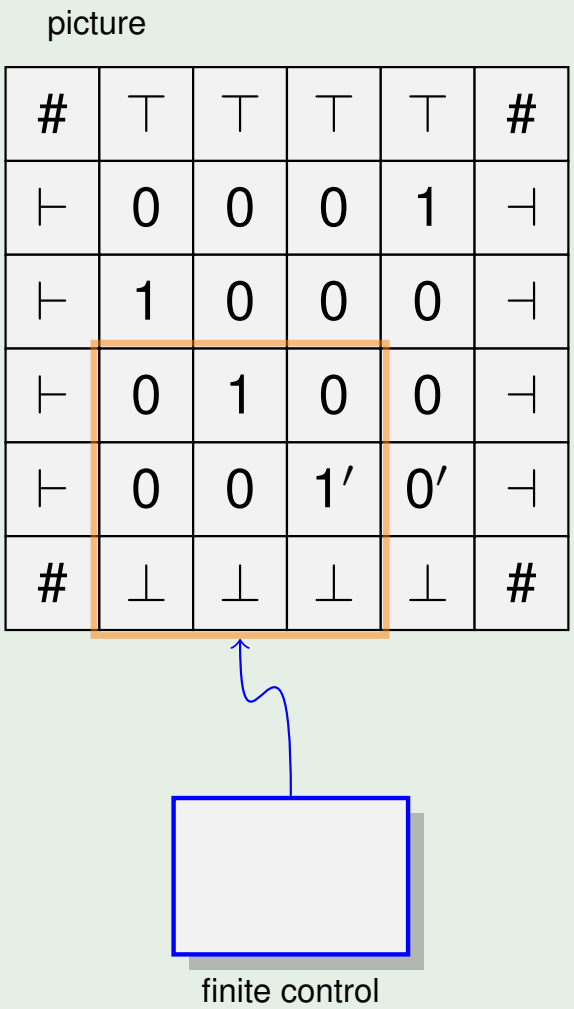
Example 1 (cont.)



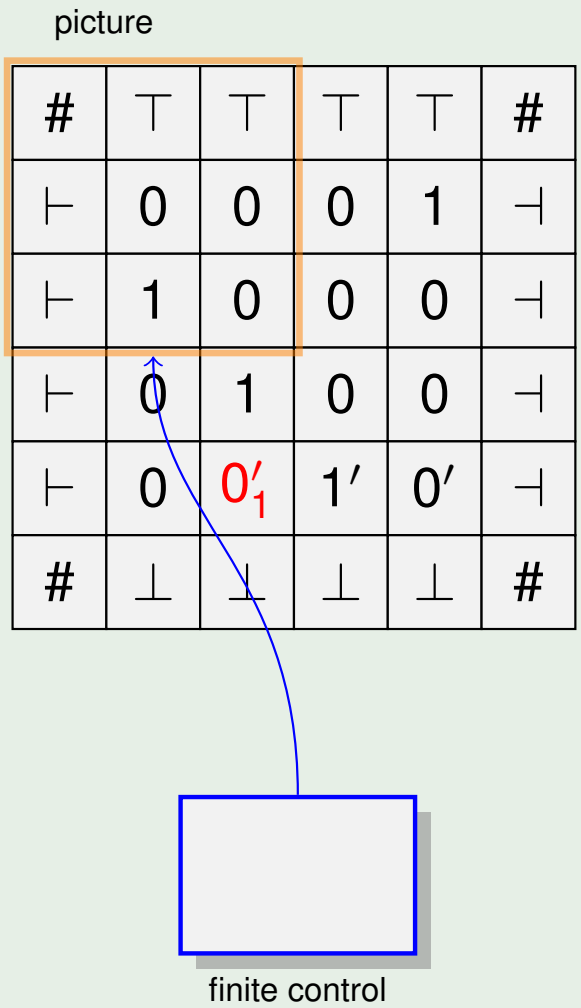
Example 1 (cont.)



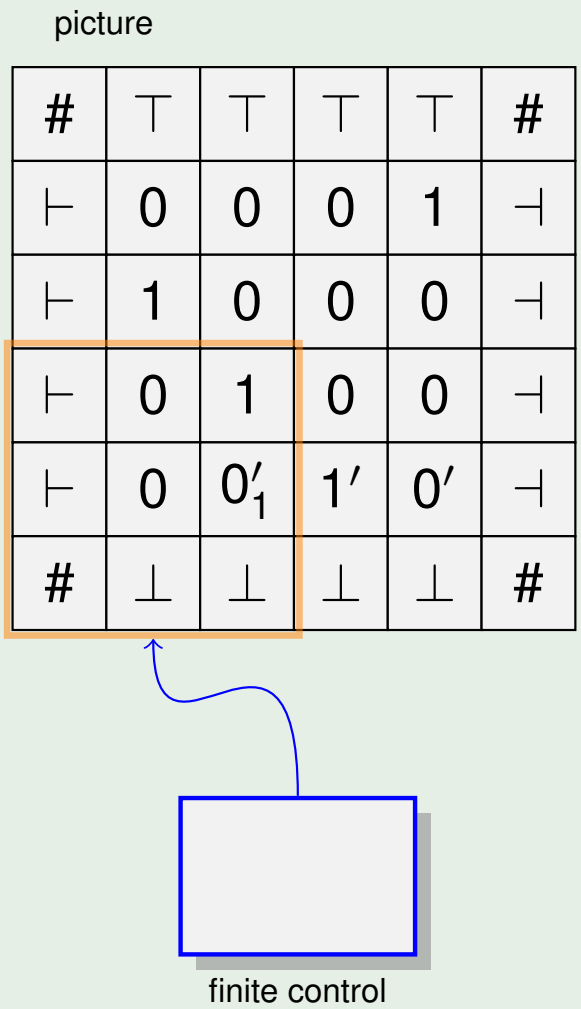
Example 1 (cont.)



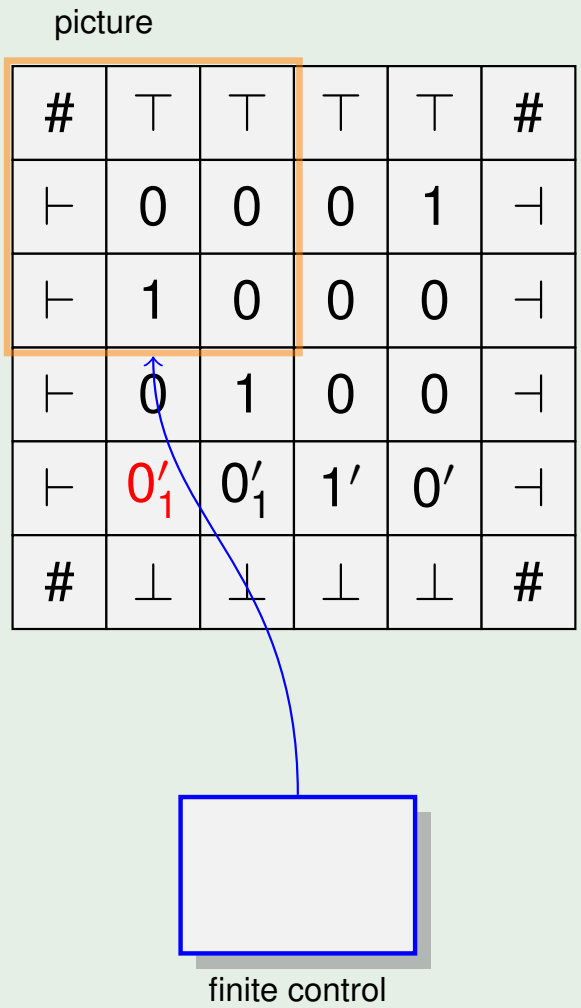
Example 1 (cont.)



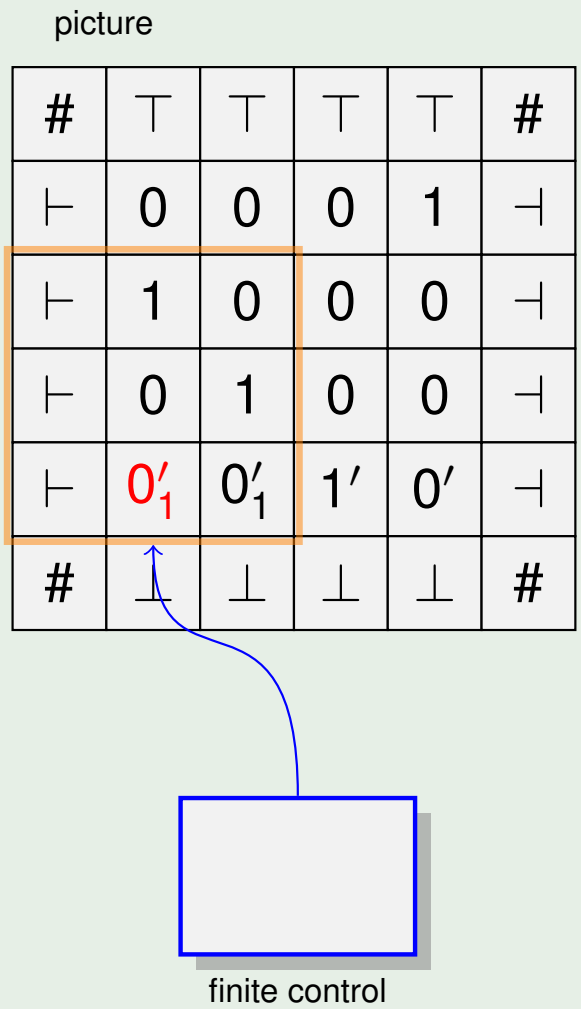
Example 1 (cont.)



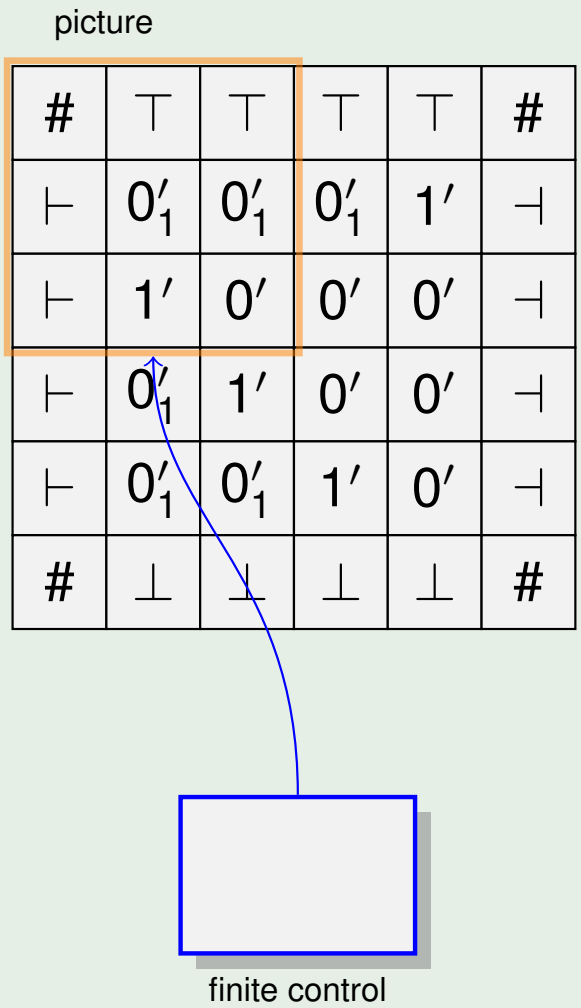
Example 1 (cont.)



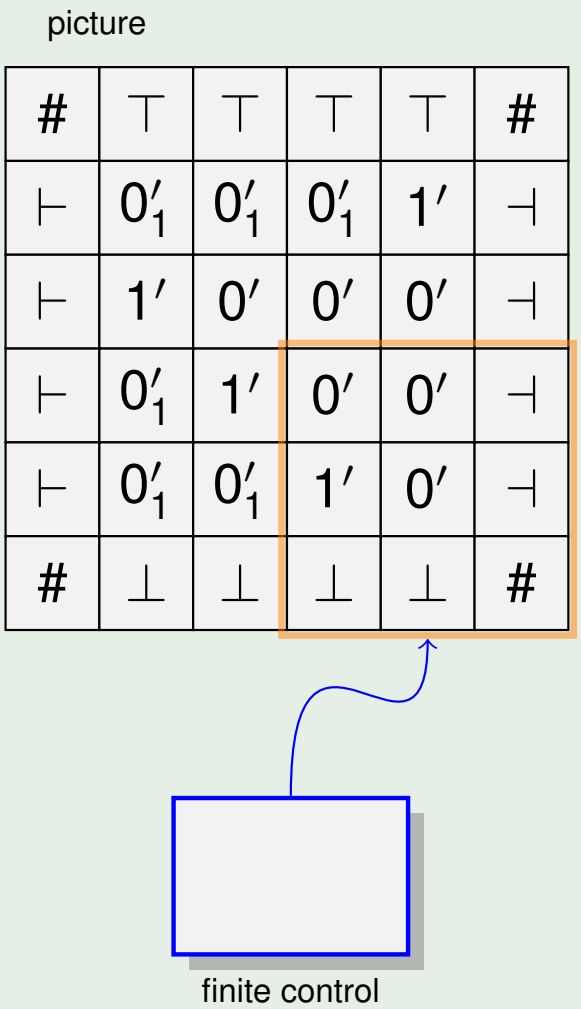
Example 1 (cont.)



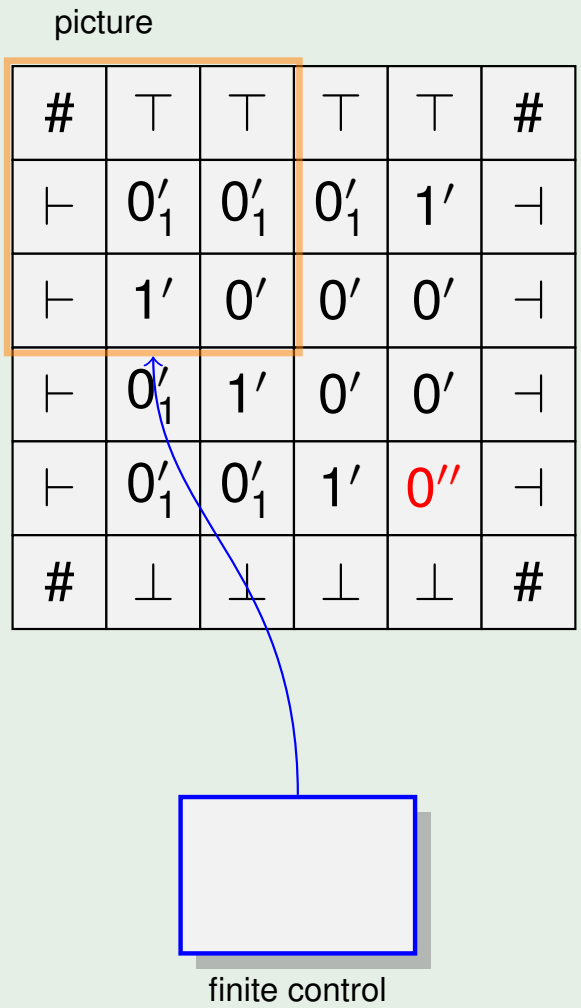
Example 1 (cont.)



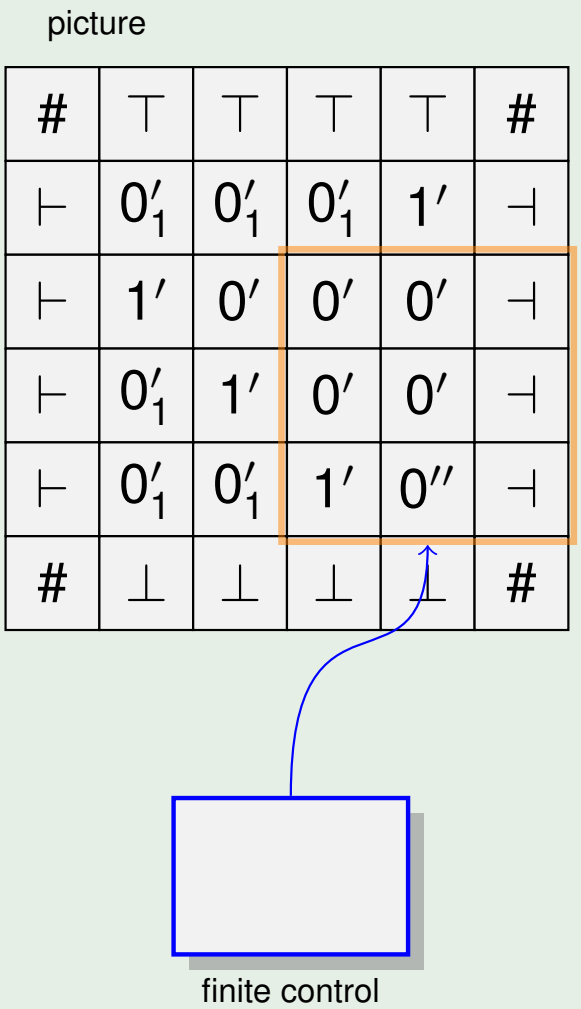
Example 1 (cont.)



Example 1 (cont.)

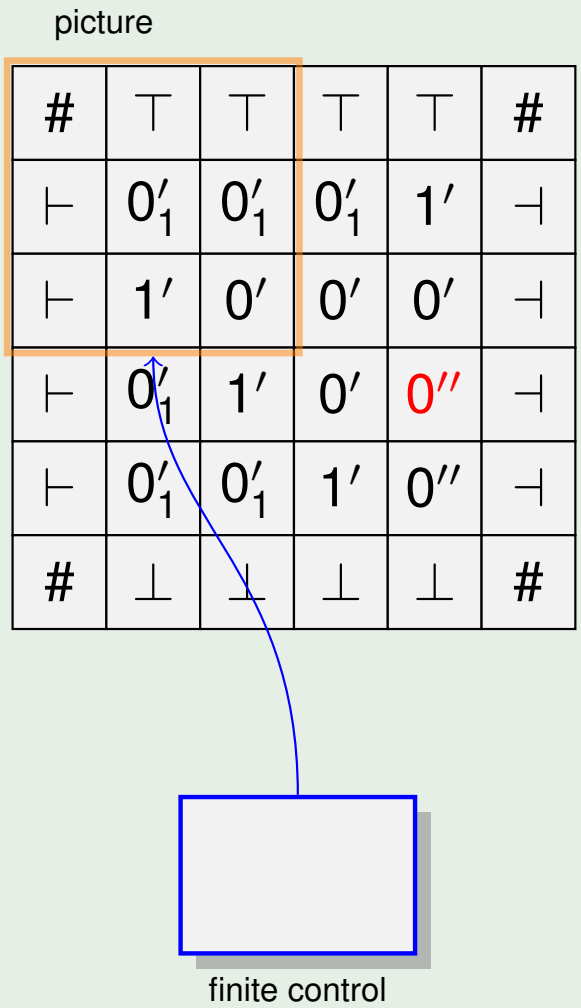


Example 1 (cont.)

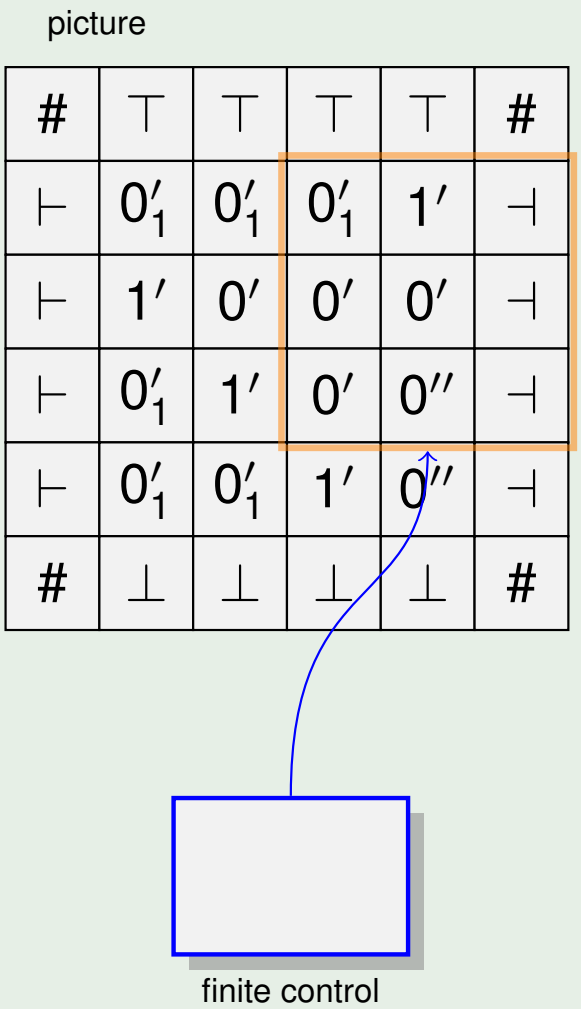


#	$\top$	$\top$	$\top$	$\top$	#
$\vdash$	$0'_1$	$0'_1$	$0'_1$	$1'$	$\dashv$
$\vdash$	$1'$	$0'$	$0'$	$0'$	$\dashv$
$\vdash$	$0'_1$	$1'$	$0'$	$0'$	$\dashv$
$\vdash$	$0'_1$	$0'_1$	$1'$	$0''$	$\dashv$
#	$\bot$	$\bot$	$\bot$	$\bot$	#

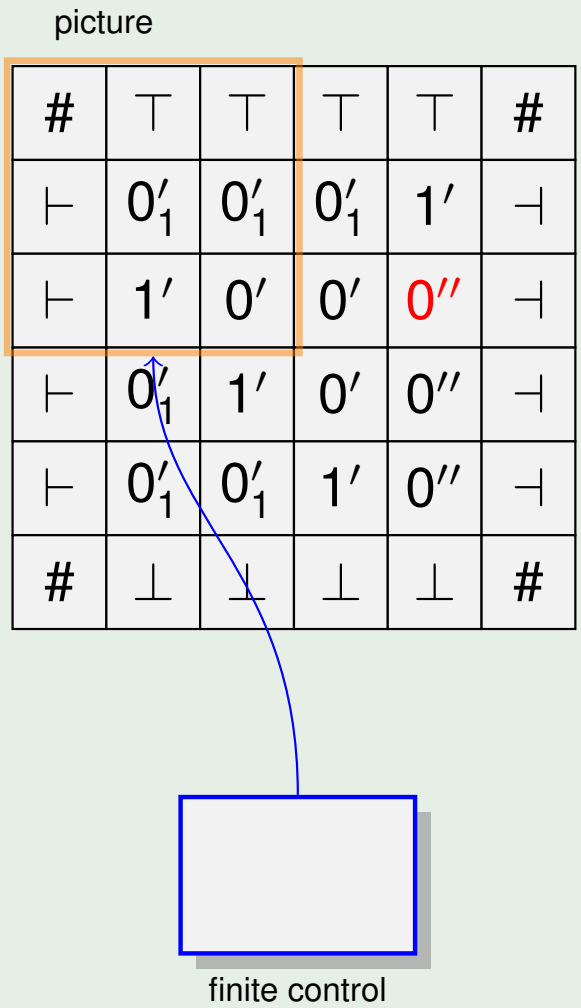
Example 1 (cont.)



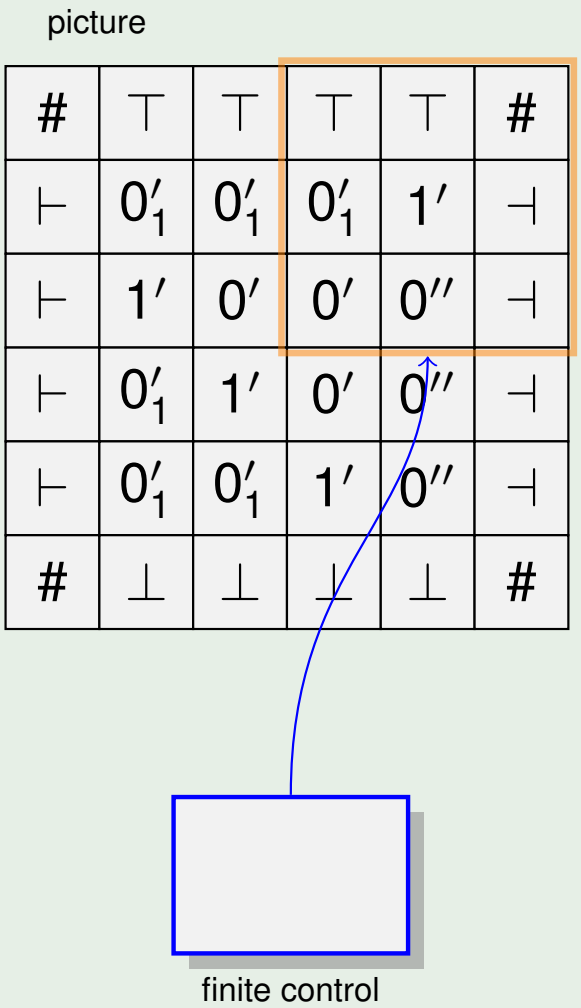
Example 1 (cont.)



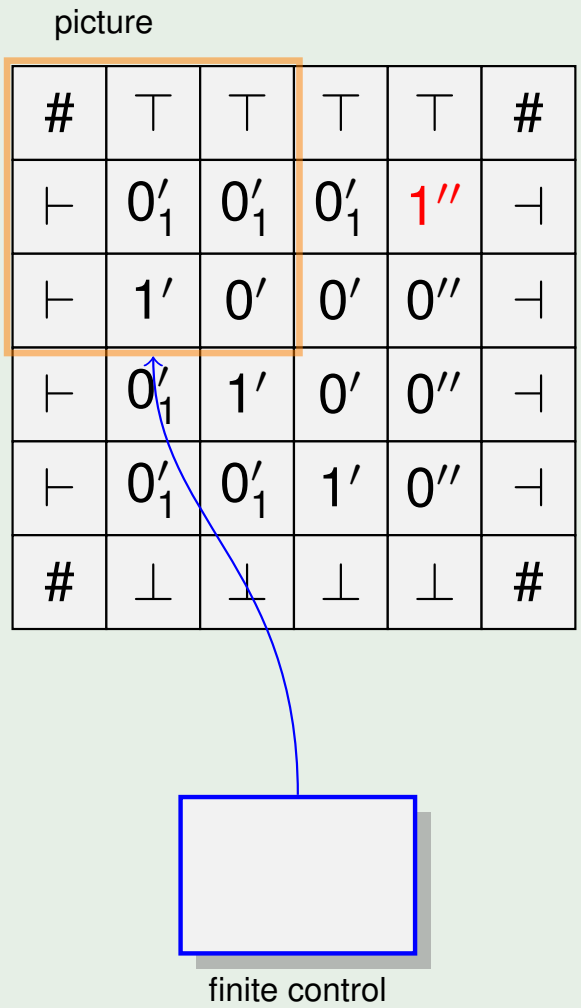
Example 1 (cont.)



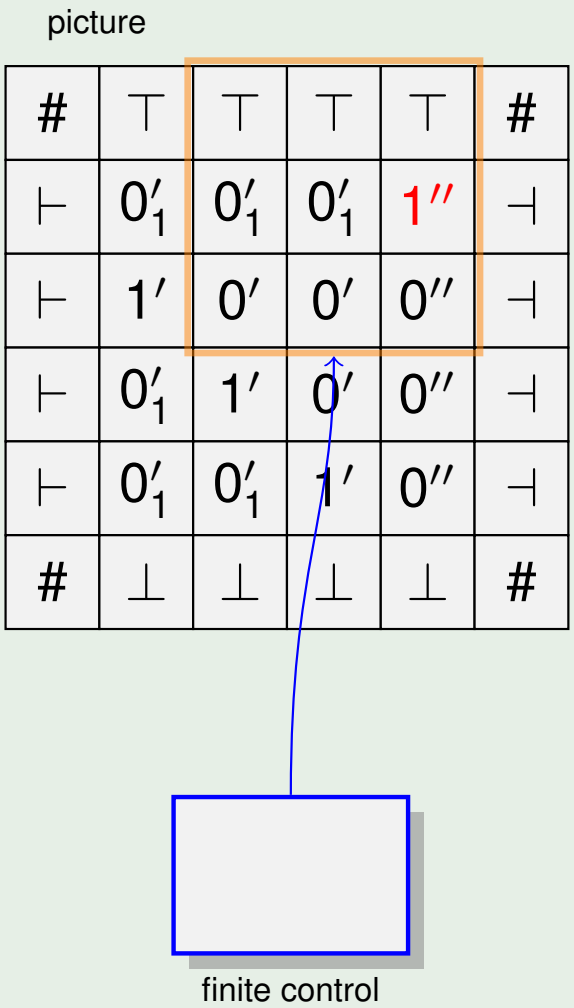
Example 1 (cont.)



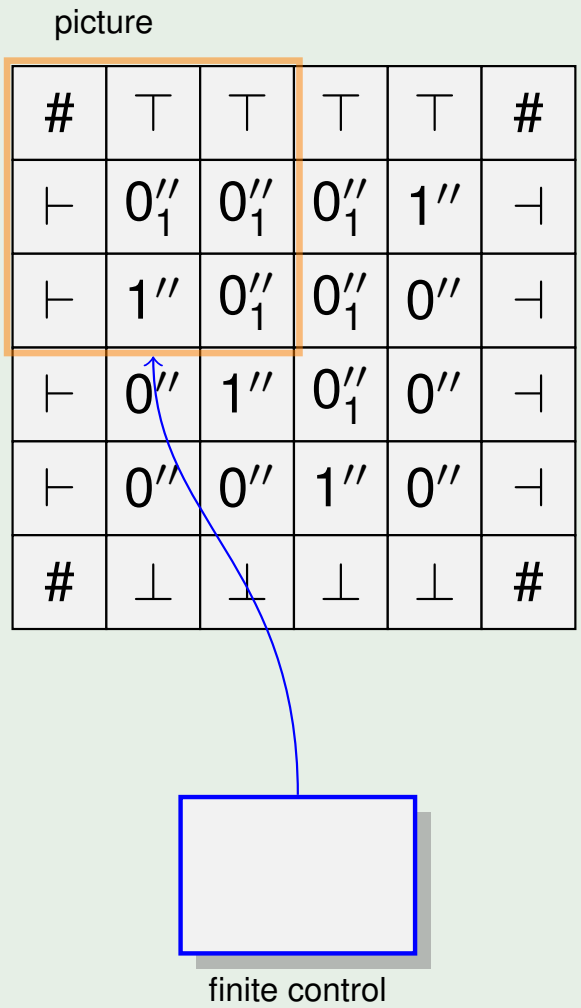
Example 1 (cont.)



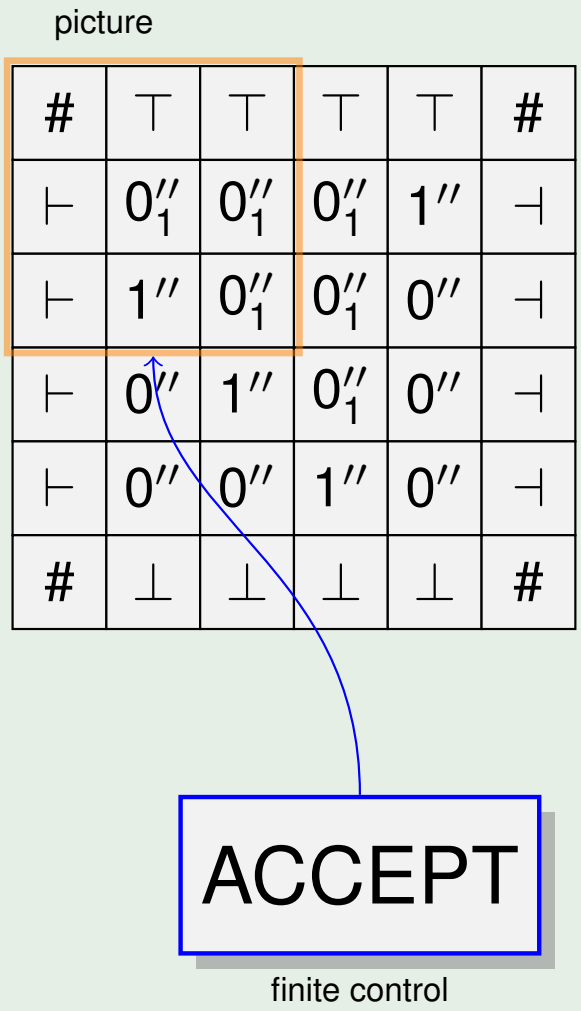
Example 1 (cont.)



Example 1 (cont.)



Example 1 (cont.)



4. On the Language Class $\mathcal{L}$(det-2D-x2W-ORWW)

Theorem 8

The stateless det-2D-x2W-ORWW-automaton can simulate the deterministic Sgraffito automaton.

Theorem 9

The classes of picture languages $\mathcal{L}$(det-2D-x2W-ORWW) *and* $\mathcal{L}$(stl-det-2D-x2W-ORWW) *are closed under transposition, union, intersection, and complementation.*

4. On the Language Class $\mathcal{L}$(det-2D-x2W-ORWW)

Theorem 8

The stateless det-2D-x2W-ORWW-automaton can simulate the deterministic Sgraffito automaton.

Theorem 9

The classes of picture languages $\mathcal{L}$(det-2D-x2W-ORWW) *and* $\mathcal{L}$(stl-det-2D-x2W-ORWW) *are closed under transposition, union, intersection, and complementation.*

In (M., O., SOFSEM 2014) it is shown that the language

$$L_{1\text{col}} = \{ P \in \Sigma^{2n,1} \mid P(1,1) \dots P(n,1) = (P(n+1,1) \dots P(2n,1))^R \},$$

is accepted by a det-2D-3W-ORWW-automaton.
The transpose $L_{1\text{col}}^t$ of this language is essentially the word language

$$L_{\text{pal}} = \{ w \in \{a,b\}^* \mid |w| \equiv 0 \bmod 2 \text{ and } w = w^R \} \notin \text{REG}.$$

Hence, $L_{1\text{col}}$ is not accepted by any det-2D-x2W-ORWW-automaton, i.e., there are det-2D-3W-ORWW-automata which cannot be simulated by det-2D-x2W-ORWW-automata.

However, also the converse holds.

Example 2

Let $L_{\mathrm{pal},2}$ be the following picture language over $\Sigma = \{a, b, \square\}$:

$$L_{\mathrm{pal},2} = \{\, P \in \Sigma^{2,2n} \mid \begin{array}{l} P(1,1)\dots P(1,n) = (P(1,n+1)\dots P(1,2n))^R, \\ P(1,i) \in \{a,b\} \text{ and } P(2,i) = \square,\ 1 \leq i \leq 2n \,\}, \end{array}$$

that is, $L_{\mathrm{pal},2}$ consists of all two-row pictures such that the first row contains a palindrome of even length over $\{a, b\}$, and the second row just contains $\square$-symbols.

We describe a det-2D-x2W-ORWW-automaton $M_{\mathrm{pal},2}$ for $L_{\mathrm{pal},2}$.

Let

$$\Gamma = \Sigma \cup \{a_1, a_2, b_1, b_2, \uparrow_1, \uparrow_2\},$$

and let

$$a > b > a_1 > b_1 > a_2 > b_2 > \square > \uparrow_1 > \uparrow_2 .$$

The automaton $M_{\mathrm{pal},2}$ proceeds as follows:

Example 2

Let $L_{\mathrm{pal},2}$ be the following picture language over $\Sigma = \{a, b, \square\}$:

$$L_{\mathrm{pal},2} = \{\, P \in \Sigma^{2,2n} \mid \begin{array}{l} P(1,1)\dots P(1,n) = (P(1,n+1)\dots P(1,2n))^R, \\ P(1,i) \in \{a,b\} \text{ and } P(2,i) = \square,\ 1 \leq i \leq 2n \,\}, \end{array}$$

that is, $L_{\mathrm{pal},2}$ consists of all two-row pictures such that the first row contains a palindrome of even length over $\{a, b\}$, and the second row just contains $\square$-symbols.

We describe a det-2D-x2W-ORWW-automaton $M_{\mathrm{pal},2}$ for $L_{\mathrm{pal},2}$.

Let

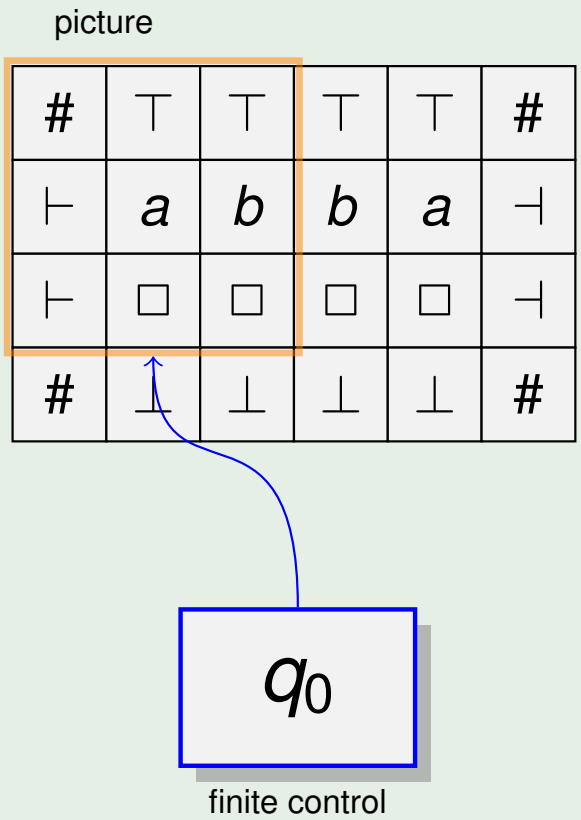
$$\Gamma = \Sigma \cup \{a_1, a_2, b_1, b_2, \uparrow_1, \uparrow_2\},$$

and let

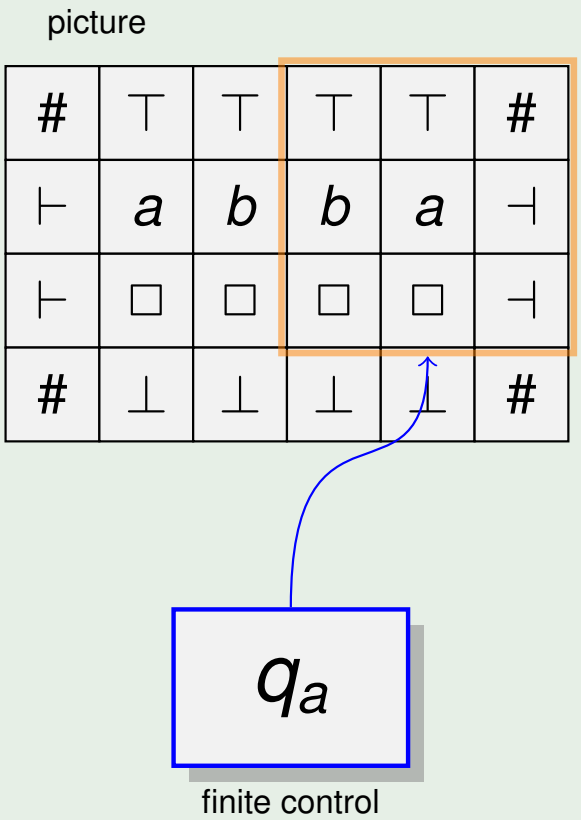
$$a > b > a_1 > b_1 > a_2 > b_2 > \square > \uparrow_1 > \uparrow_2 .$$

The automaton $M_{\mathrm{pal},2}$ proceeds as follows:

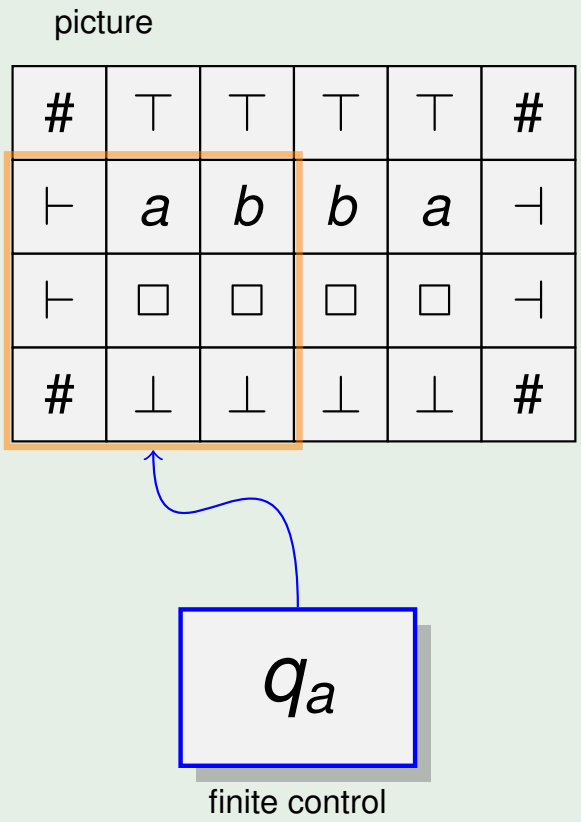
Example 2 (cont.)



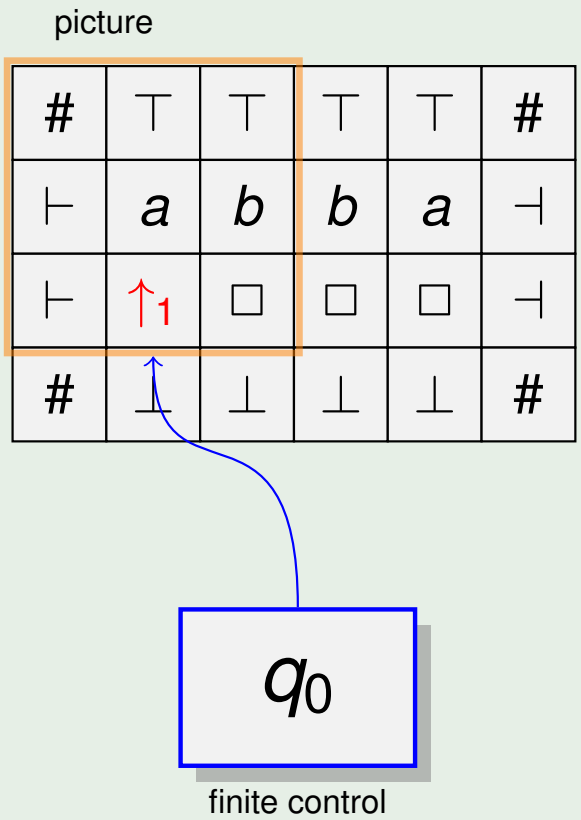
Example 2 (cont.)



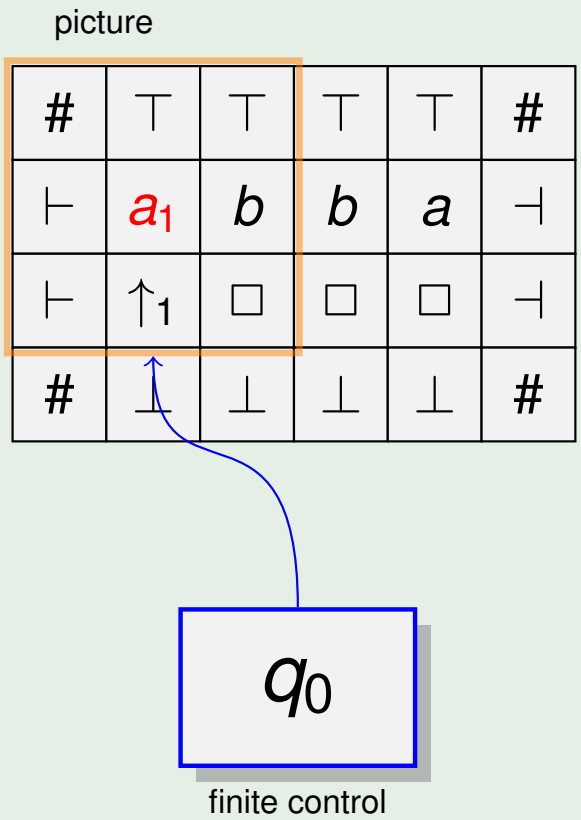
Example 2 (cont.)



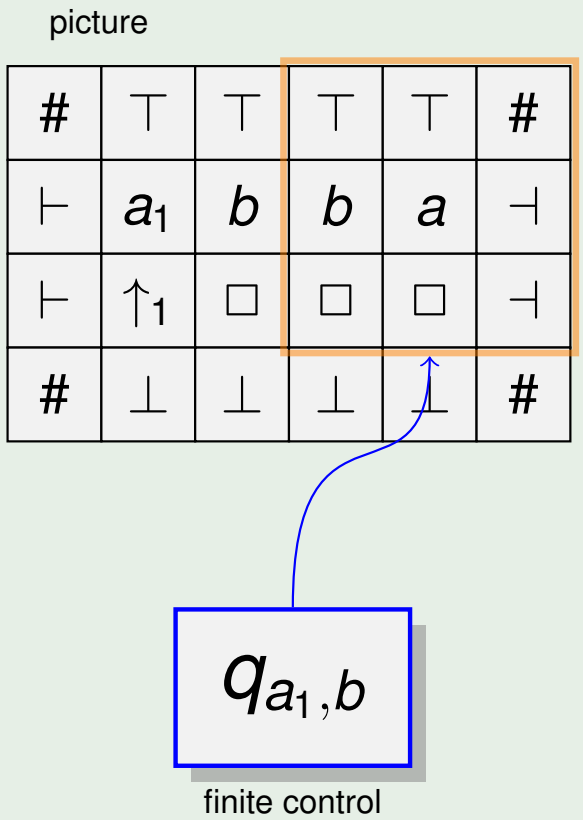
Example 2 (cont.)



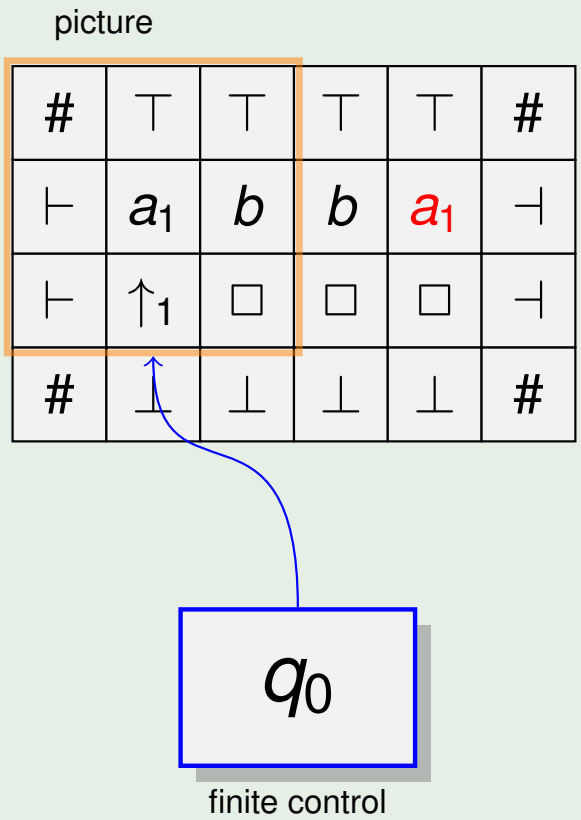
Example 2 (cont.)



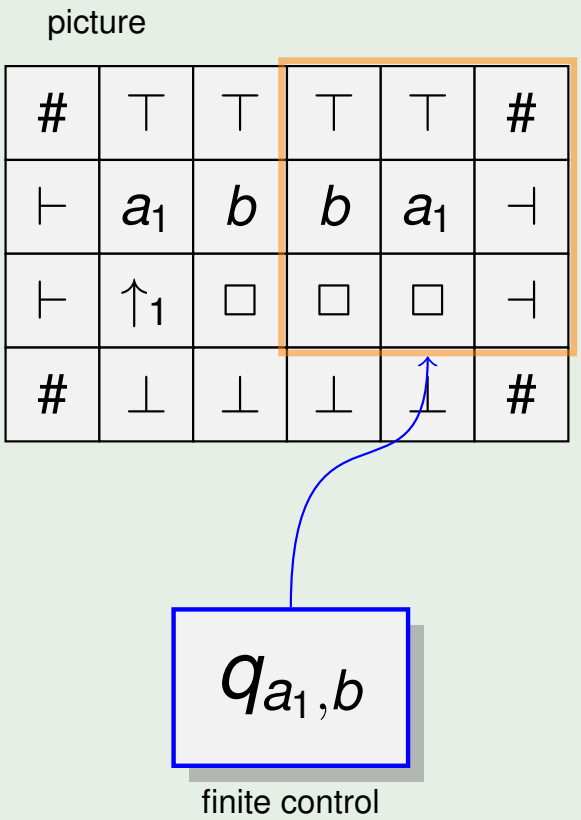
Example 2 (cont.)



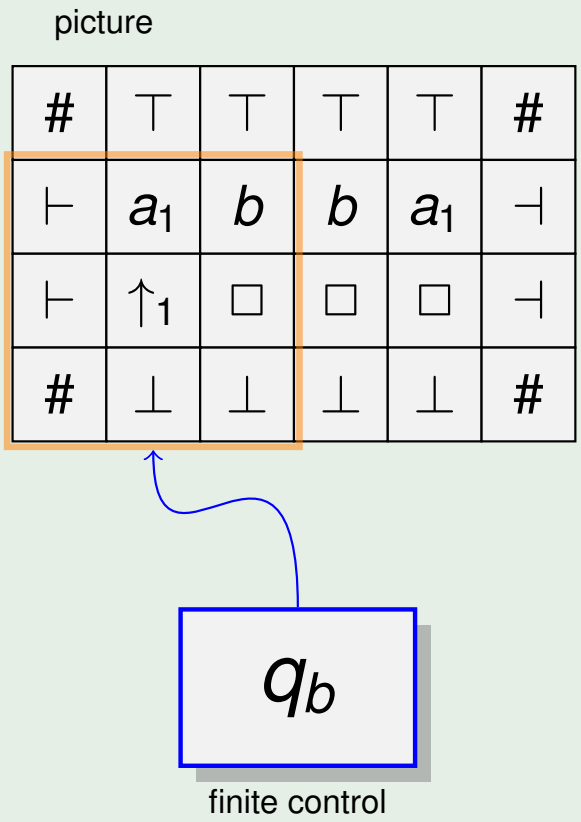
Example 2 (cont.)



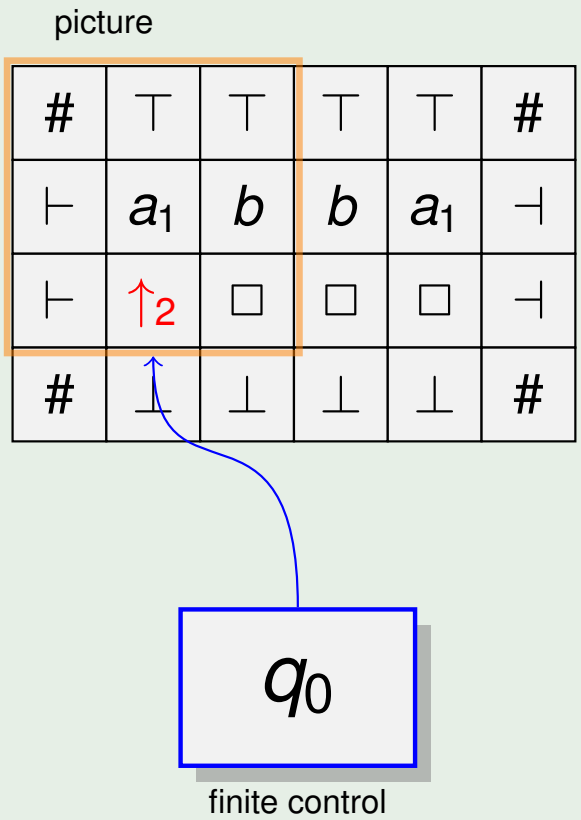
Example 2 (cont.)



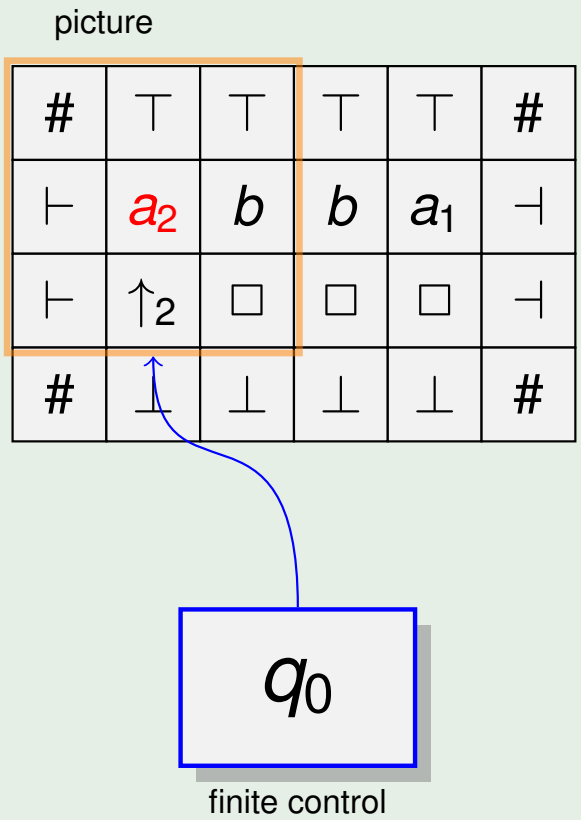
Example 2 (cont.)



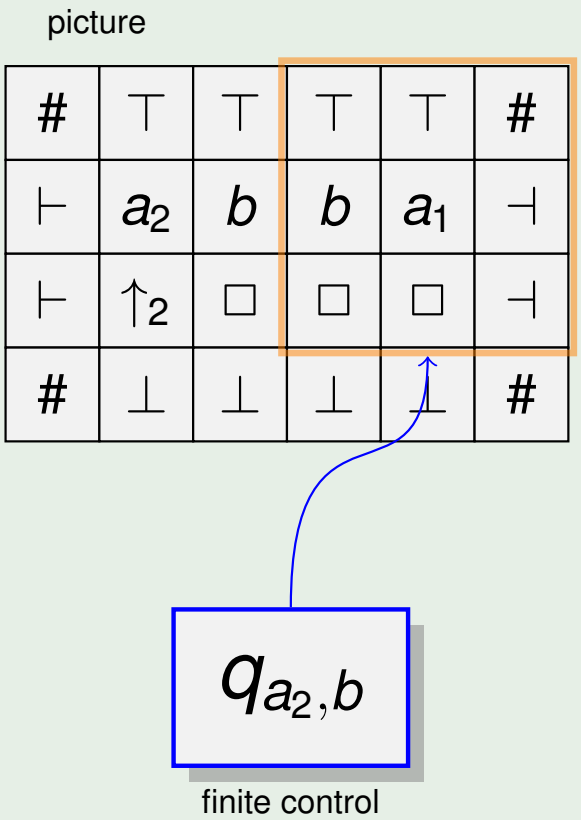
Example 2 (cont.)



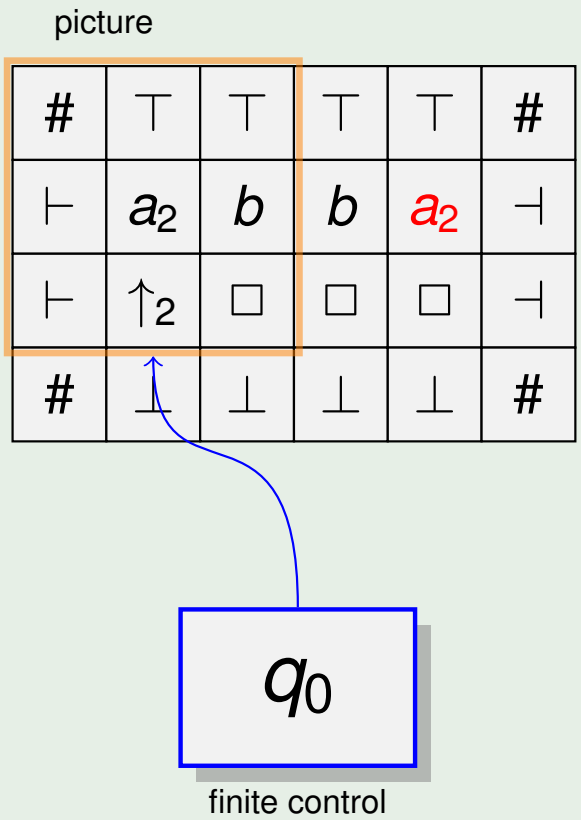
Example 2 (cont.)



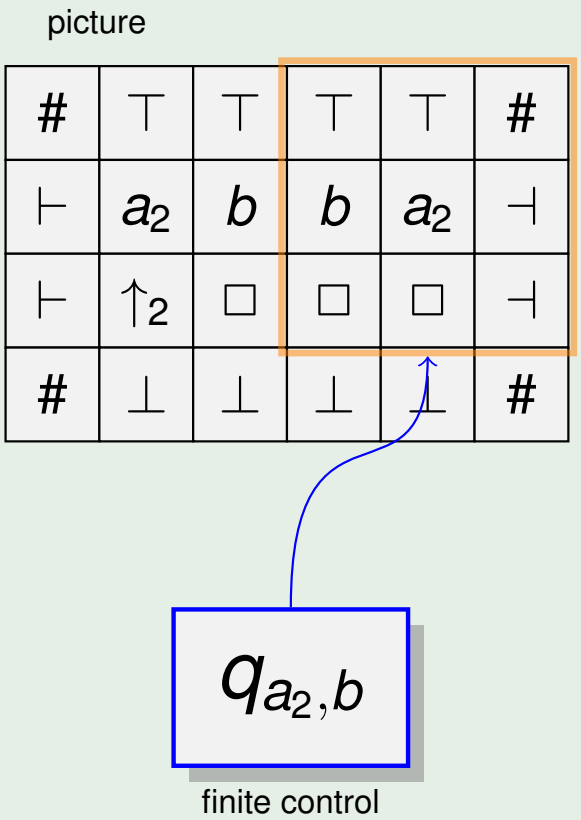
Example 2 (cont.)



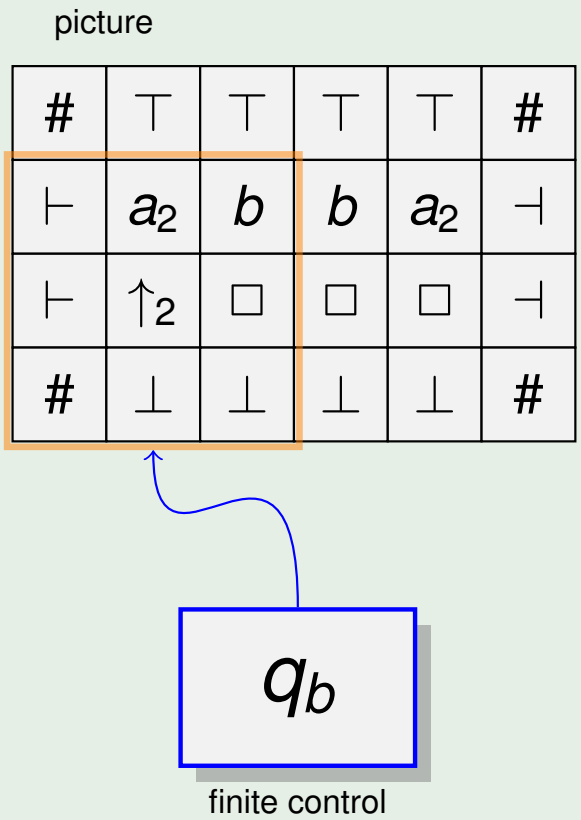
Example 2 (cont.)



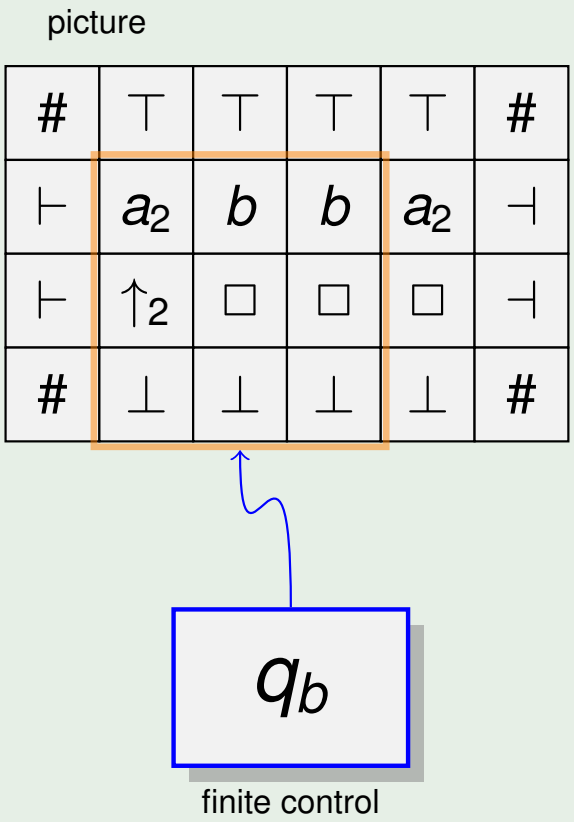
Example 2 (cont.)



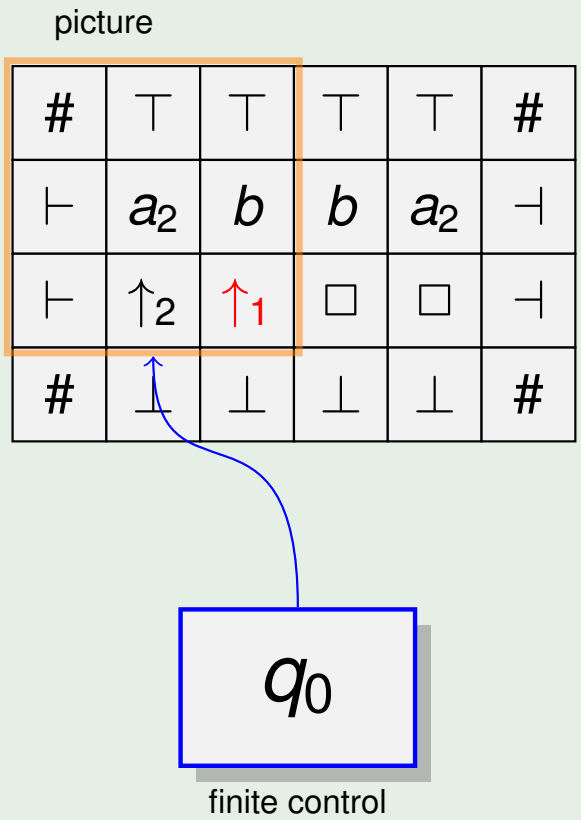
Example 2 (cont.)



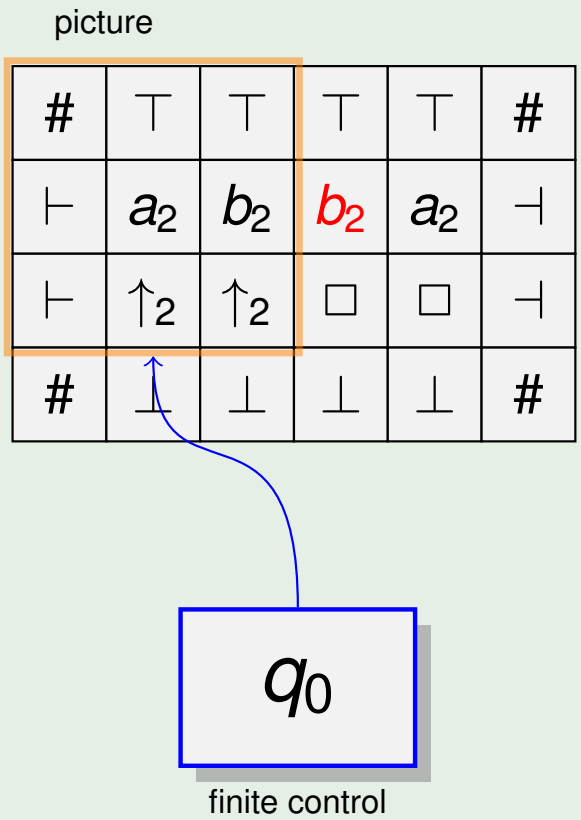
Example 2 (cont.)



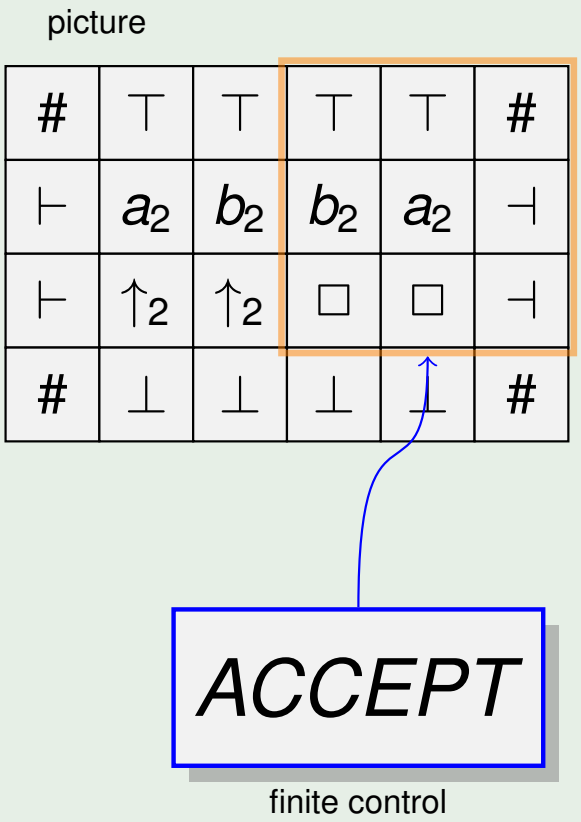
Example 2 (cont.)



Example 2 (cont.)



Example 2 (cont.)



Proposition 10

$L_{\text{pal},2} \notin \mathcal{L}$(det-2D-3W-ORWW).

Corollary 11

The class of picture languages $\mathcal{L}$(det-2D-x2W-ORWW) *is incomparable under inclusion to the class of picture languages* $\mathcal{L}$(det-2D-3W-ORWW).

Proposition 10

$L_{\text{pal},2} \notin \mathcal{L}$(det-2D-3W-ORWW).

Corollary 11

The class of picture languages $\mathcal{L}$(det-2D-x2W-ORWW) *is incomparable under inclusion to the class of picture languages* $\mathcal{L}$(det-2D-3W-ORWW).

Proposition 12

$L_{\text{pal},2} \notin \mathcal{L}$(stl-det-2D-x2W-ORWW).

Theorem 13

$\mathcal{L}$(stl-det-2D-x2W-ORWW) $\subsetneq \mathcal{L}$(det-2D-x2W-ORWW).

Proposition 12

$L_{\text{pal},2} \notin \mathcal{L}$(stl-det-2D-x2W-ORWW).

Theorem 13

$\mathcal{L}$(stl-det-2D-x2W-ORWW) $\subsetneq \mathcal{L}$(det-2D-x2W-ORWW).

Proof outline of Prop. 12.

Assume that $M = (\Sigma, \Gamma, \mathcal{S}, \delta, >)$ is a stl-det-2D-x2W-ORWW-automaton over $\Sigma = \{a, b, \square\}$ such that $L(M) = L_{\text{pal},2}$.

For $w = a_1 \dots a_n$, where $n \geq 1$ and $a_1, \dots, a_n \in \{a, b\}$, let

$$P_w = \begin{bmatrix} a_1 & \dots & a_n & a & a & a_n & \dots & a_1 \\ \square & \dots & \square & \square & \square & \square & \dots & \square \end{bmatrix} \in L_{\text{pal},2}.$$

Given P_w as input, M will perform an accepting computation, which can be split into a finite number of phases, where we distinguish between four types of phases:

- A left-only phase consists of a sequence of cycles in which the window of M stays on the left half of the picture.
- An upper-right phase consists of a sequence of cycles in which all rewrite steps are performed on the right half of the picture, and in addition, in the first of these cycles, M enters the right half of the picture through a move-right step in row 1.

Proof outline of Prop. 12.

Assume that $M = (\Sigma, \Gamma, \mathcal{S}, \delta, >)$ is a stl-det-2D-x2W-ORWW-automaton over $\Sigma = \{a, b, \square\}$ such that $L(M) = L_{\text{pal},2}$.

For $w = a_1 \dots a_n$, where $n \geq 1$ and $a_1, \dots, a_n \in \{a, b\}$, let

$$P_w = \begin{bmatrix} a_1 & \dots & a_n & a & a & a_n & \dots & a_1 \\ \square & \dots & \square & \square & \square & \square & \dots & \square \end{bmatrix} \in L_{\text{pal},2}.$$

Given P_w as input, M will perform an accepting computation, which can be split into a finite number of phases, where we distinguish between four types of phases:

- A left-only phase consists of a sequence of cycles in which the window of M stays on the left half of the picture.
- An upper-right phase consists of a sequence of cycles in which all rewrite steps are performed on the right half of the picture, and in addition, in the first of these cycles, M enters the right half of the picture through a move-right step in row 1.

Proof outline of Prop. 12.

Assume that $M = (\Sigma, \Gamma, \mathcal{S}, \delta, >)$ is a stl-det-2D-x2W-ORWW-automaton over $\Sigma = \{a, b, \square\}$ such that $L(M) = L_{\text{pal},2}$.

For $w = a_1 \dots a_n$, where $n \geq 1$ and $a_1, \dots, a_n \in \{a, b\}$, let

$$P_w = \begin{bmatrix} a_1 & \dots & a_n & a & a & a_n & \dots & a_1 \\ \square & \dots & \square & \square & \square & \square & \dots & \square \end{bmatrix} \in L_{\text{pal},2}.$$

Given P_w as input, M will perform an accepting computation, which can be split into a finite number of phases, where we distinguish between four types of phases:

- A left-only phase consists of a sequence of cycles in which the window of M stays on the left half of the picture.
- An upper-right phase consists of a sequence of cycles in which all rewrite steps are performed on the right half of the picture, and in addition, in the first of these cycles, M enters the right half of the picture through a move-right step in row 1.

Proof outline of Prop. 12.

Assume that $M = (\Sigma, \Gamma, \mathcal{S}, \delta, >)$ is a stl-det-2D-x2W-ORWW-automaton over $\Sigma = \{a, b, \square\}$ such that $L(M) = L_{\text{pal},2}$.

For $w = a_1 \dots a_n$, where $n \geq 1$ and $a_1, \dots, a_n \in \{a, b\}$, let

$$P_w = \begin{bmatrix} a_1 & \dots & a_n & a & a & a_n & \dots & a_1 \\ \square & \dots & \square & \square & \square & \square & \dots & \square \end{bmatrix} \in L_{\text{pal},2}.$$

Given P_w as input, M will perform an accepting computation, which can be split into a finite number of phases, where we distinguish between four types of phases:

- A left-only phase consists of a sequence of cycles in which the window of M stays on the left half of the picture.
- An upper-right phase consists of a sequence of cycles in which all rewrite steps are performed on the right half of the picture, and in addition, in the first of these cycles, M enters the right half of the picture through a move-right step in row 1.

Proof outline (cont.)

- A lower-left phase is a sequence of cycles in which all rewrite steps are performed in the left half of the picture, and in addition, the first of these cycles contains an extended move-right step.
- A lower-right phase is a sequence of cycles in which all rewrite steps are performed in the right half of the picture, and in addition, in the first of these cycles, M enters the right half of the picture through a move-right step in row 2 or through an extended move-down step.

The sequence of cycles of the computation of M on input P_w can uniquely be split into a sequence of phases of maximum length.

Thus, this computation can be described in a unique way by a string α over the alphabet $\Omega = \{O, U, L, R\}$, where O denotes a left-Only phase, U stands for an Upper-right phase, L denotes a lower-Left phase, and R stands for a lower-Right phase.

Proof outline (cont.)

- A lower-left phase is a sequence of cycles in which all rewrite steps are performed in the left half of the picture, and in addition, the first of these cycles contains an extended move-right step.
- A lower-right phase is a sequence of cycles in which all rewrite steps are performed in the right half of the picture, and in addition, in the first of these cycles, M enters the right half of the picture through a move-right step in row 2 or through an extended move-down step.

The sequence of cycles of the computation of M on input P_w can uniquely be split into a sequence of phases of maximum length.

Thus, this computation can be described in a unique way by a string α over the alphabet $\Omega = \{O, U, L, R\}$, where O denotes a left-Only phase, U stands for an Upper-right phase, L denotes a lower-Left phase, and R stands for a lower-Right phase.

Proof outline (cont.)

- A lower-left phase is a sequence of cycles in which all rewrite steps are performed in the left half of the picture, and in addition, the first of these cycles contains an extended move-right step.
- A lower-right phase is a sequence of cycles in which all rewrite steps are performed in the right half of the picture, and in addition, in the first of these cycles, M enters the right half of the picture through a move-right step in row 2 or through an extended move-down step.

The sequence of cycles of the computation of M on input P_w can uniquely be split into a sequence of phases of maximum length.

Thus, this computation can be described in a unique way by a string α over the alphabet $\Omega = \{O, U, L, R\}$, where O denotes a left-Only phase, U stands for an Upper-right phase, L denotes a lower-Left phase, and R stands for a lower-Right phase.

Proof outline (cont.)

Concerning the possible changes from one phase to the next there are some restrictions based on the fact that M is stateless, e.g.,

- while M is in a lower-right phase (R), it just moves through the left half of the current picture after each rewrite/restart step. Thus, M cannot get into another phase until it performs a rewrite step that replaces a symbol in the first column of the right half of the picture. However, in a fixed column, less than $2 \cdot |\Gamma|$ many rewrite steps can be performed, and so $|\alpha|_R \leq 1 + 2 \cdot |\Gamma|$.

It can be shown that

$$|\alpha| \leq |\alpha|_O + |\alpha|_R + |\alpha|_L + |\alpha|_U \leq 15 + 28 \cdot |\Gamma|,$$

i.e., each computation of M consists of $\leq 15 + 28 \cdot |\Gamma|$ many phases.

Using a notion of generalized crossing sequence and counting arguments it can now be shown that M will also accept some pictures that do not belong to the language $L_{\text{pal},2}$, a contradiction! □

Proof outline (cont.)

Concerning the possible changes from one phase to the next there are some restrictions based on the fact that M is stateless, e.g.,

- while M is in a lower-right phase (R), it just moves through the left half of the current picture after each rewrite/restart step. Thus, M cannot get into another phase until it performs a rewrite step that replaces a symbol in the first column of the right half of the picture. However, in a fixed column, less than $2 \cdot |\Gamma|$ many rewrite steps can be performed, and so $|\alpha|_R \leq 1 + 2 \cdot |\Gamma|$.

It can be shown that

$$|\alpha| \leq |\alpha|_O + |\alpha|_R + |\alpha|_L + |\alpha|_U \leq 15 + 28 \cdot |\Gamma|,$$

i.e., each computation of M consists of $\leq 15 + 28 \cdot |\Gamma|$ many phases.

Using a notion of generalized crossing sequence and counting arguments it can now be shown that M will also accept some pictures that do not belong to the language $L_{\text{pal},2}$, a contradiction! □

Proof outline (cont.)

Concerning the possible changes from one phase to the next there are some restrictions based on the fact that M is stateless, e.g.,

- while M is in a lower-right phase (R), it just moves through the left half of the current picture after each rewrite/restart step. Thus, M cannot get into another phase until it performs a rewrite step that replaces a symbol in the first column of the right half of the picture. However, in a fixed column, less than $2 \cdot |\Gamma|$ many rewrite steps can be performed, and so $|\alpha|_R \leq 1 + 2 \cdot |\Gamma|$.

It can be shown that

$$|\alpha| \leq |\alpha|_O + |\alpha|_R + |\alpha|_L + |\alpha|_U \leq 15 + 28 \cdot |\Gamma|,$$

i.e., each computation of M consists of $\leq 15 + 28 \cdot |\Gamma|$ many phases.

Using a notion of generalized crossing sequence and counting arguments it can now be shown that M will also accept some pictures that do not belong to the language $L_{\text{pal},2}$, a contradiction! □

Let $\Sigma = \{0, 1\}$, and let

$$L_{\text{dup}} = \{\, P \oplus P \mid P \text{ is a quadratic picture over } \Sigma \,\},$$

where $P \oplus P$ denotes the column concatenation of two copies of P.

It is known that $L_{\text{dup}} \notin \mathcal{L}$(2SA) (Pruša, Mráz, DLT 2012) .

However, by using the technique from Example 2, the following can be shown.

Proposition 14

$L_{\text{dup}} \in \mathcal{L}$(det-2D-x2W-ORWW).

Corollary 15

$\mathcal{L}$(det-2D-x2W-ORWW) $\not\subset \mathcal{L}$(2SA).

Let $\Sigma = \{0, 1\}$, and let

$$L_{\text{dup}} = \{\, P \oplus P \mid P \text{ is a quadratic picture over } \Sigma \,\},$$

where $P \oplus P$ denotes the column concatenation of two copies of P.

It is known that $L_{\text{dup}} \notin \mathcal{L}$(2SA) (Pruša, Mráz, DLT 2012) .

However, by using the technique from Example 2, the following can be shown.

Proposition 14

$L_{\text{dup}} \in \mathcal{L}$(det-2D-x2W-ORWW).

Corollary 15

$\mathcal{L}$(det-2D-x2W-ORWW) $\not\subset \mathcal{L}$(2SA).

Let $\Sigma = \{0, 1\}$, and let

$$L_{\text{dup}} = \{\, P \oplus P \mid P \text{ is a quadratic picture over } \Sigma \,\},$$

where $P \oplus P$ denotes the column concatenation of two copies of P.

It is known that $L_{\text{dup}} \notin \mathcal{L}$(2SA) (Pruša, Mráz, DLT 2012) .

However, by using the technique from Example 2, the following can be shown.

Proposition 14

$L_{\text{dup}} \in \mathcal{L}$(det-2D-x2W-ORWW).

Corollary 15

$\mathcal{L}$(det-2D-x2W-ORWW) $\not\subseteq \mathcal{L}$(2SA).

5. Concluding Remarks

The det-2D-x2W-ORWW-automaton has the following nice properties:

- The membership problem for the picture languages accepted is decidable in polynomial time.
- It is even more expressive than the deterministic Sgraffito automaton.
- It only accepts regular word languages.
- It avoids the artificial termination condition of the det-2D-3W-ORWW-automaton.

5. Concluding Remarks

The det-2D-x2W-ORWW-automaton has the following nice properties:

- The membership problem for the picture languages accepted is decidable in polynomial time.
- It is even more expressive than the deterministic Sgraffito automaton.
- It only accepts regular word languages.
- It avoids the artificial termination condition of the det-2D-3W-ORWW-automaton.

5. Concluding Remarks

The det-2D-x2W-ORWW-automaton has the following nice properties:

- The membership problem for the picture languages accepted is decidable in polynomial time.
- It is even more expressive than the deterministic Sgraffito automaton.
- It only accepts regular word languages.
- It avoids the artificial termination condition of the det-2D-3W-ORWW-automaton.

5. Concluding Remarks

The det-2D-x2W-ORWW-automaton has the following nice properties:

- The membership problem for the picture languages accepted is decidable in polynomial time.
- It is even more expressive than the deterministic Sgraffito automaton.
- It only accepts regular word languages.
- It avoids the artificial termination condition of the det-2D-3W-ORWW-automaton.

Open Problems:

- Is the language class $\mathcal{L}$(det-2D-x2W-ORWW) closed under projection, under horizontal product, or under vertical product?
- Can stateless det-2D-x2W-ORWW-automata accept any languages that are not accepted by deterministic Sgraffito automata, or do these two types of automata have exactly the same expressive power?
- Can stateless det-2D-x2W-ORWW-automata be simulated by det-2D-3W-ORWW-automata with states?

Open Problems:

- Is the language class $\mathcal{L}$(det-2D-x2W-ORWW) closed under projection, under horizontal product, or under vertical product?
- Can stateless det-2D-x2W-ORWW-automata accept any languages that are not accepted by deterministic Sgraffito automata, or do these two types of automata have exactly the same expressive power?
- Can stateless det-2D-x2W-ORWW-automata be simulated by det-2D-3W-ORWW-automata with states?

Open Problems:

- Is the language class $\mathcal{L}$(det-2D-x2W-ORWW) closed under projection, under horizontal product, or under vertical product?
- Can stateless det-2D-x2W-ORWW-automata accept any languages that are not accepted by deterministic Sgraffito automata, or do these two types of automata have exactly the same expressive power?
- Can stateless det-2D-x2W-ORWW-automata be simulated by det-2D-3W-ORWW-automata with states?

Thank you for your attention!