Networks of Evolutionary Processors: Words and Pictures

Group of Mathematical Modelling and Bioinspired Algorithms

Universidad Politécnica de Madrid

Summary

1. Preliminaries
2. Evolutionary Pictures Operations
3. Circular Permutation Rules
4. Filters Predicates
5. Picture Processors
6. Accepting Networks of Picture Processors
7. Problems

1. Preliminaries

POLITÉCNICA

- **Goal**: To apply distributed computing methods to networks of sub-microprocessor devices, e.g., biological cellular networks or networks of nano-devices.
- **Question**: do tiny bio/nano nodes "compute" and/or "communicate" essentially the same as a computer?
- **Our attempt**: Although the computation and communication capabilities of each individual device in the new model are, by design, much weaker than those of a computer, we show that some of the most important and extensively studied distributed computing problems can still be solved efficiently.
- Theorists: YES, WE CAN!
- Engineers: ?
- Biologists: ?

1. Preliminaries

- V is an alphabet

$$V^* = \{\omega \mid \omega = \omega_1\omega_2 \ldots \omega_p \,,\ \omega_i \in V \ \wedge p \ \in \mathbb{N}\}$$

- ω is a word over V and V* is the set of words over V
- The alphabet of ω is defined by $alph(\omega) = \{\, a \in V \mid a \ \in \omega\}$
- $L \ \subseteq V^*$ is a language over V

1. Preliminaries

- A picture π over an alphabet *V* is a two-dimensional array of elements from *V*.

$$\pi = (p_i^j)_{i\in\mathbb{N}}^{j\in\mathbb{N}} \wedge p_i^j \in V$$

$$V_*^* = \{\pi |\ \pi\ is\ a\ picture\ over\ V\}$$

$\overline{\underline{\pi}}$ is the number of rows of π

$|\pi|$ is the number of columns of π

size (π) = ($\overline{\underline{\pi}}$, $|\pi|$)

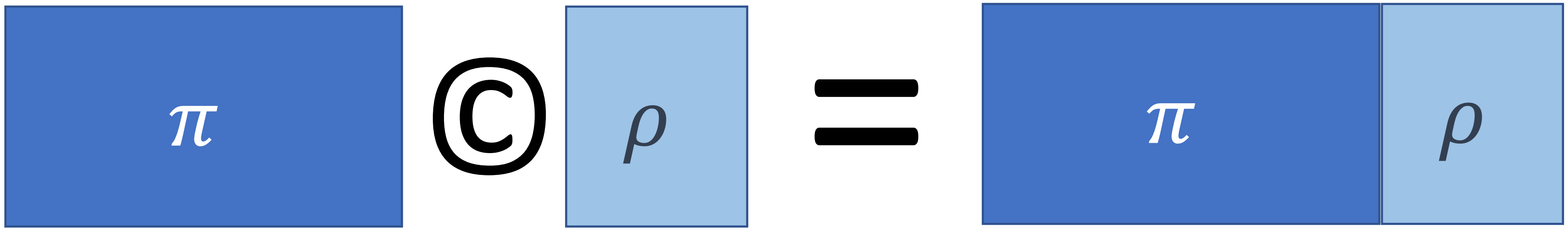
2. Evolutionary Pictures Operations

- Row and column concatenation operations

$$Let\ \pi\ and\ \rho \in V_*^*\ two\ pictures\ over\ V$$
$$size(\pi) = (m, n), size\ (\rho) = (m', n')$$

- π © ρ is the column concatenation of π and ρ and it is defined only if

m = m'



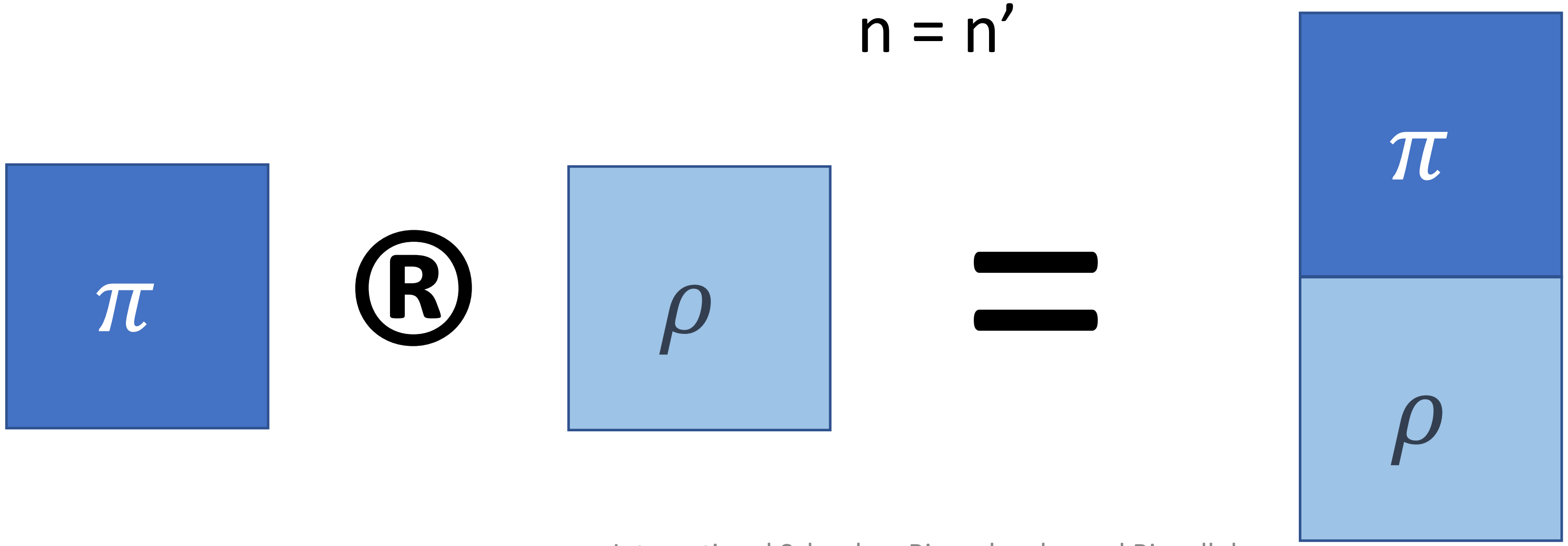
2. Evolutionary Pictures Operations

- Row and column concatenation operations

$$Let\ \pi\ and\ \rho\ \in V_*^*\ two\ pictures\ over\ V$$
$$size(\pi) = (m, n), size\ (\rho) = (m', n')$$

- π ® ρ is the row concatenation of π and ρ and it is defined only if

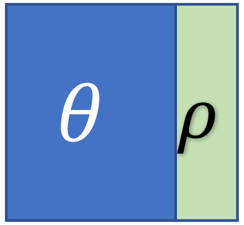
n = n’



2. Evolutionary Pictures Operations

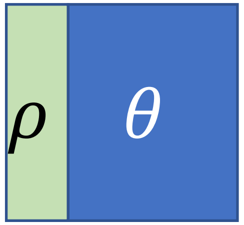
- Column right-quotient of π with ρ

$$\pi / \leftarrow \rho = \theta \Leftrightarrow \pi = \theta \copyright \rho$$



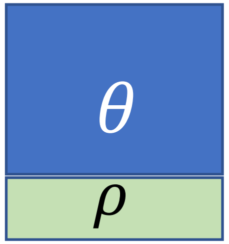
- Column left-quotient of π with ρ

$$\pi / \rightarrow \rho = \theta \Leftrightarrow \pi = \rho \copyright \theta$$



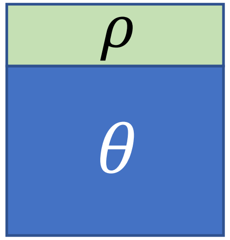
- Row down – quotient of π with ρ

$$\pi / \downarrow \rho = \theta \Leftrightarrow \pi = \theta \circledR \rho$$



- Row up – quotient of π with ρ

$$\pi / \uparrow \rho = \theta \Leftrightarrow \pi = \rho \circledR \theta$$



2. Evolutionary Pictures Operations

- Evolutionary rules

$$\sigma \equiv a \rightarrow b, \text{ is a:}$$

- substitution rule iff $a \neq \varepsilon \wedge b \neq \varepsilon$
- deletion rule iff $a \neq \varepsilon \wedge b = \varepsilon$
- insertion rule iff $a = \varepsilon \wedge b \neq \varepsilon$

2. Evolutionary Pictures Operations

- Definitions:

$$Sub_V = \{\sigma \mid \sigma \text{ is a substitution rule}\}$$

$$Del_V = \{\sigma \mid \sigma \text{ is a deletion rule}\}$$

2. Evolutionary Pictures Operations

- Actions for evolutionary rules when are applied on pictures

Given a rule σ an a picture π $\in V_m^n$ we defined the following actions:

$$\sigma^{\leftarrow}(\pi)$$
$$\sigma^{\rightarrow}(\pi)$$
$$\sigma^{\uparrow}(\pi)$$
$$\sigma^{\downarrow}(\pi)$$

$$\sigma^{|}(\pi)$$
$$\sigma^{-}(\pi)$$
$$\sigma^{+}(\pi)$$

2. Evolutionary Pictures Operations

- If $\sigma \equiv a \rightarrow b \in Sub_V$ *then*
  - *If the first column of contains an occurrence of a, then* $\sigma^{\leftarrow}(\pi)$ *is the set of all pictures* π' *such that the following conditions are satisfied:*
    i. $\exists\, 1 \leq i \leq m, \pi(i,1) = a \wedge \pi'(i,1) = b,$
    ii. $\pi(k,l) = \pi'(k,l)\ for\ all\ (k,l) \in [m]x[n] - \{(i,1)\}.$
  - *If this column does not contain any occurrence of a, then* $\sigma^{\leftarrow}(\pi)$ *=*$\{\pi\}$

2. Evolutionary Pictures Operations

$$\sigma^{\leftarrow}\left(\begin{bmatrix} a & b & b & c \\ c & b & b & a \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\right) = \left\{\begin{bmatrix} b & b & b & c \\ c & b & b & a \\ c & c & b & c \\ a & b & c & b \end{bmatrix}, \begin{bmatrix} a & b & b & c \\ c & b & b & a \\ c & c & b & c \\ b & b & c & b \end{bmatrix}\right\}$$

$$\sigma^{\leftarrow}\left(\begin{bmatrix} c & b & b & c \\ c & b & b & a \\ c & c & b & c \\ b & b & c & b \end{bmatrix}\right) = \left\{\begin{bmatrix} c & b & b & c \\ c & b & b & a \\ c & c & b & c \\ b & b & c & b \end{bmatrix}\right\}$$

2. Evolutionary Pictures Operations

- In an analogous way can be defined

$$\sigma^{\rightarrow}(\pi), \sigma^{\uparrow}(\pi), \sigma^{\downarrow}(\pi), \sigma^{+}(\pi)$$

as the set of all pictures obtained by applying σ to the right most column, to the first row, to the las row, and to any column/row of π.

2. Evolutionary Pictures Operations

$$\sigma^{\rightarrow}\left(\begin{bmatrix} a & b & b & c \\ c & b & b & a \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\right) = \left\{\begin{bmatrix} a & b & b & c \\ c & b & b & b \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\right\}$$

$$\sigma^{\uparrow}\left(\begin{bmatrix} a & b & b & c \\ c & b & b & a \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\right) = \left\{\begin{bmatrix} b & b & b & c \\ c & b & b & a \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\right\}$$

2. Evolutionary Pictures Operations

$$\sigma^{\downarrow}\left(\begin{bmatrix} a & b & b & c \\ c & b & b & a \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\right) = \left\{\begin{bmatrix} a & b & b & c \\ c & b & b & a \\ c & c & b & c \\ b & b & c & b \end{bmatrix}\right\}$$

$$\sigma^{+}\left(\begin{bmatrix} a & b & b & c \\ c & b & b & a \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\right) = \left\{\begin{bmatrix} b & b & b & c \\ c & b & b & a \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\begin{bmatrix} a & b & b & c \\ c & b & b & b \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\begin{bmatrix} a & b & b & c \\ c & b & b & a \\ c & c & b & c \\ b & b & c & b \end{bmatrix}\right\}$$

2. Evolutionary Pictures Operations

- If $\sigma \equiv a \to \varepsilon \in Del_V$ then

$$\sigma^{\leftarrow}(\pi) = \begin{cases} \pi / \leftarrow \rho \\ \pi, otherwise \end{cases}$$

Where ρ is the leftmost column of π, and ρ contains at least one occurrence of ‘a’

$$\sigma^{\leftarrow}\left(\begin{bmatrix} a & b & b & c \\ c & b & b & a \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\right) = \begin{bmatrix} b & b & c \\ b & b & a \\ c & b & c \\ b & c & b \end{bmatrix}$$

2. Evolutionary Pictures Operations

- Analogously are defined

$$\sigma^{\rightarrow}(\pi), \sigma^{\uparrow}(\pi), \sigma^{\downarrow}(\pi)$$

2. Evolutionary Pictures Operations

- Furthermore $\sigma^{|}(\pi)$ $(\sigma^{-}(\pi))$ is the set obtained from π by deleting an arbitrary column (row) of π containing the symbol ‘a’, then each column (row) is removed from different copies of π.
- If π does not contain a copy of ‘a’ then

$$\sigma^{|}(\pi) = \sigma^{-}(\pi) = \{\pi\}$$

2. Evolutionary Pictures Operations

$$\sigma^{|}\left(\begin{bmatrix} a & b & b & a \\ c & b & b & a \\ c & c & b & c \\ a & b & c & b \end{bmatrix}\right) = \left\{\begin{bmatrix} b & b & a \\ b & b & a \\ c & b & c \\ b & c & b \end{bmatrix}, \begin{bmatrix} a & b & b \\ c & b & b \\ c & c & b \\ a & b & c \end{bmatrix}\right\}$$

2. Evolutionary Pictures Operations

- Definitions:

i. For every rule σ, symbol $\alpha = \{\leftarrow, \rightarrow, \uparrow, \downarrow, |, -, +\}$ and $L \subseteq V_*^{*}$; we define the α-action of σ on *L* by

$$\sigma^{\alpha}(L) = \bigcup_{\pi \in L} \sigma^{\alpha}(\pi)$$

Note: + is defined only for substitution rules M, while | and – are defined only for deletion rules.

2. Evolutionary Pictures Operations

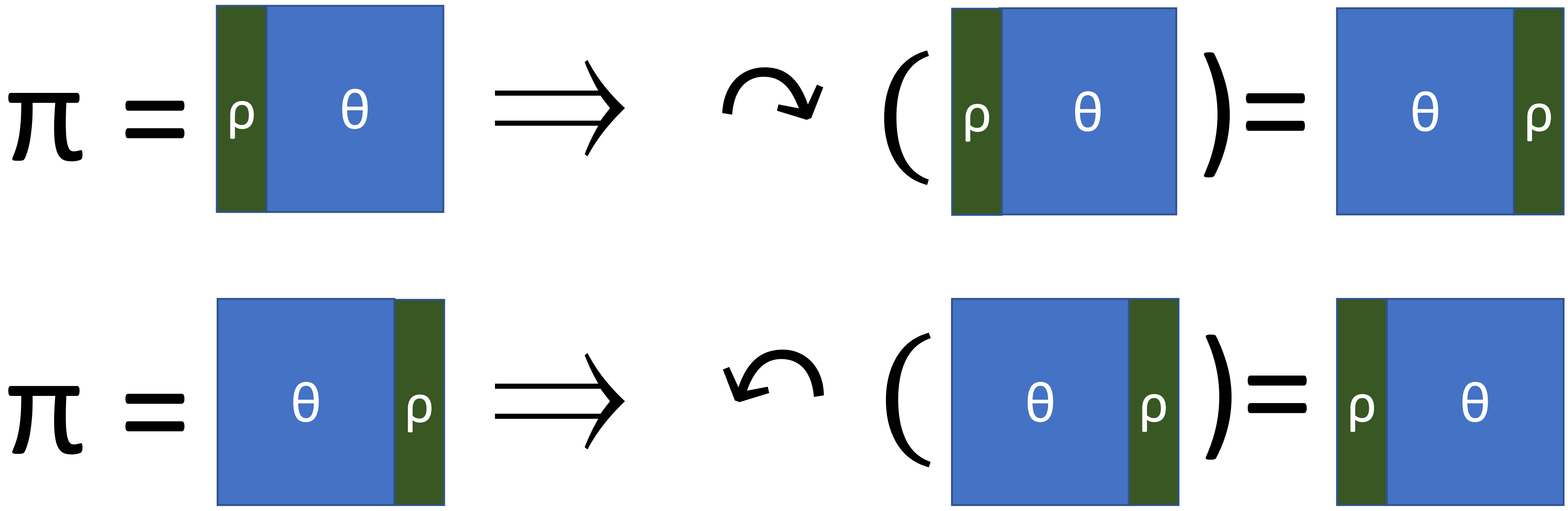
Definitions:

ii. Given a finite set of rules *M*, we define the α-action of *M* on the picture π and the language *L by*

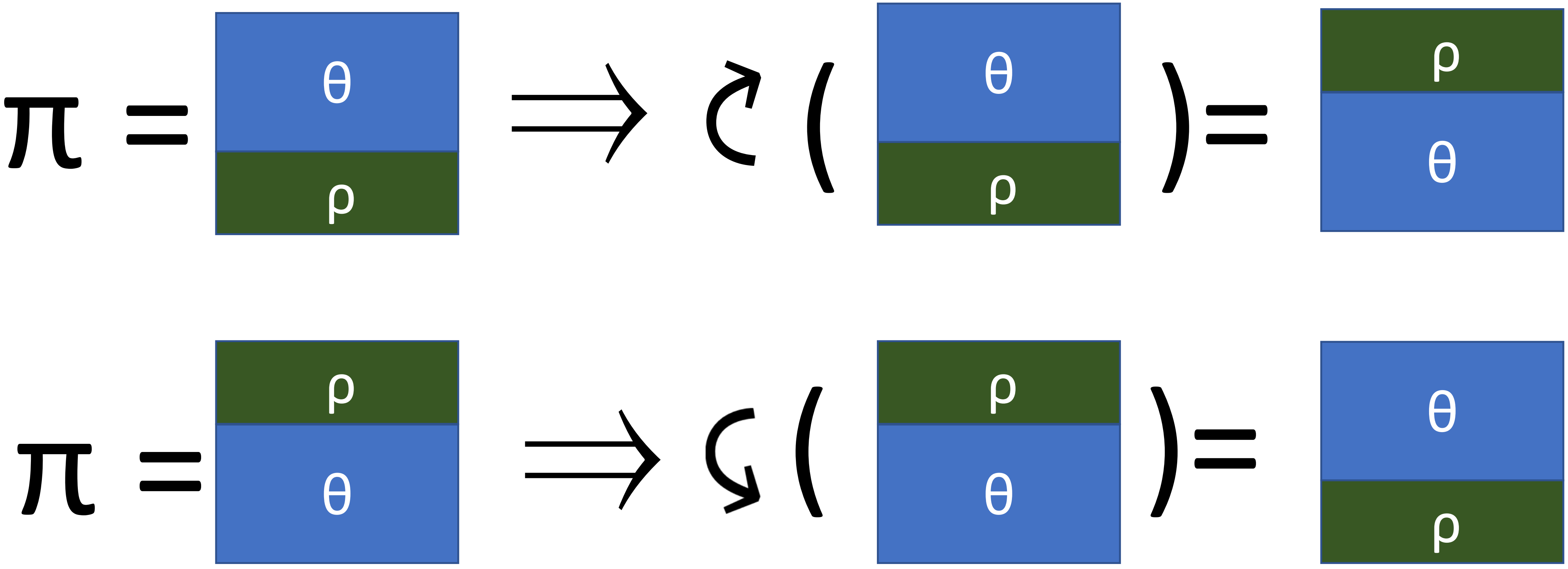
$$M^{\alpha}(\pi) = \bigcup_{\sigma \in M} \sigma^{\alpha}(\pi) \text{ and}$$

$$M^{\alpha}(L) = \bigcup_{\pi \in L} M^{\alpha}(\pi)$$

3. Circular Permutation Rules



3. Circular Permutation Rules



3. Circular Permutation Rules

- Definitions:

  For every $\circ \in \{\curvearrowleft, \curvearrowright, ⤹, ⤸\}$ and $L \subseteq V_*^*$,
  we define

$$\circ(L) = \{\circ(\pi) | \pi \in L\}$$

4. Filters Predicates

- As usual, we define two different policies for defining filters, strong and week defined by the following two predicates.
- Consider $P, F \subseteq V$ the set of permitting and forbidding symbols respectively

$$rc_s(\pi; P, F) \equiv P \subseteq alph(\pi) \land F \cap alph(\pi) = \emptyset$$
$$rc_w(\pi; P, F) \equiv P \cap alph(\pi) \neq \emptyset \land F \cap alph(\pi) = \emptyset$$

4. Filters Predicates

- For every picture language $L \subseteq V_*^*$ and $\beta \in \{s, w\}$ we define:

$$rc_\beta(L; P, F) = \{\pi \in L | rc_\beta(\pi; P, F) = true\}$$

5. Picture Processors

- Evolutionary Picture Processor (EPP)

An evolutionary picture Processor (EPP) over *V* is a 5-tuple

(M,PI,FI,PO,FO)

where:

- $M \subseteq Sub_V$ *or* $M \subseteq Del_V$ *(specialization)*
- $PI, FI \subseteq V$ *(permitting / forbidding input symbols)*
- $PO, FO \subseteq V$ *(permitting / forbidding outpt symbols)*

5. Picture Processors

- Circularly Permuted Picture Processor (CPPP)

A CPPP over *V* is a 5-tuple

$$(M, PI, PO, FI, FO)$$

where:

- ∘ ∈ {↶, ↷, ⤾, ⤿}
- The rest of parameters are identical to those defined for evolutionary processors.

6. Accepting Networks of Picture Processors (ANPP)

- An ANPP is a 9-tuple

$$\Gamma=(V, U, G, N, \alpha, \beta, \underline{In}, \underline{Halt}, \underline{Accept})$$

where:

- *V*, *U* are the input and network alphabet respectively $V \subseteq U$,
- $G = (X_G, E_G)$ is an undirected graph without loops,
- *N* is a mapping which associates each node $x \in X_G$ the picture processor

$$N(x) = (M_x, PI_x, PO_x, FI_x, FO_x)$$

6. Accepting Networks of Picture Processors (ANPP)

- $\alpha: X_G \rightarrow \{\rightarrow, \leftarrow, \uparrow, \downarrow, |, -, *\}$ gives the action mode of the rules of node *x* on the pictures existing in that node. If *x* is a CPPP then $\alpha(x) = *$
- $\beta: X_G \rightarrow \{s, w\}$ defines the type of input and output filters

$$\text{Input filter: } \rho_x(.) = rc_{\beta(x)}(.; PI_x, FI_x)$$

$$\text{Output filter: } \tau_x(.) = rc_{\beta(x)}(.; PO_x, FO_x)$$

- *In*, *Halt*, *Accept* $\in X_G$ are the input node, the halting node and the accepting node of *Γ* respectively.

6. Accepting Networks of Picture Processors (ANPP)

- <u>Definitions:</u>

1. We say that $Card(X_G)$ is the size of Γ.
2. A configuration of an ANPP Γ is a mapping

$$C: X_G \to 2^{U_*^*}$$

which associates a finite set of pictures with every node in the graph.

6. Accepting Networks of Picture Processors (ANPP)

- Given a picture $\pi \in V_*^*$, the initial configuration of Γ on π is defined by

$$C_0^{(\pi)}(\underline{In}) = \{\pi\}$$

$$C_0^{(\pi)}(x) = \emptyset \; for \; all \; x \in X_G \backslash \{\underline{In}\}$$

A configuration can change via a processing step or a communication step

6. Accepting Networks of Picture Processors (ANPP)

- Processing step

$$C \Rightarrow C' \; iff$$

$$C'(x) = M_x^{\alpha(x)}\big(C(x)\big) \forall x \in X_G \backslash \{\underline{In}\}$$

6. Accepting Networks of Picture Processors (ANPP)

- Communication step

$$C \vdash C' iff$$

$$C'(x) = (C(x) \backslash \tau_x(C(x)) \cup \bigcup_{\{x,y\} \in E_G} \left(\tau_y(C(y)) \cap \rho_x(C(y))\right), \forall x \in X_G$$

6. Accepting Networks of Picture Processors (ANPP)

- <u>Definition:</u>

Let Γ be an ANPP; the configuration of Γ on an input picture $\pi \in V_*^*$ is a sequence of configurations

$$C_0^{(\pi)} \Longrightarrow C_1^{(\pi)} \vdash C_2^{(\pi)} \Longrightarrow \; \dots C_n^{(\pi)}$$

where $C_0^{(\pi)}$ is the initial configuration of Γ on π

6. Accepting Networks of Picture Processors (ANPP)

- A computation halts if there exits a configuration such that the set of pictures existing in the halting node is non-empty.
- The picture language decided by Γ is:

$$L(\Gamma) = \{\pi \in V_*^{\,*} | the\ computation\ of\ \Gamma\ on\ \pi\ halts\ with\ a\ non-empty\ accepting\ node\}$$

7. Problems

- 2-Dimensional pattern matching

Given a picture π one can decide whether π is a subpicture of a given picture

SOLVED

What about 2-disjoint or overlapping occurrences?

7. Problems

- **Th. [BBLM – TPNC 2014]**

1. *Let π be a picture of size (k; n) for some $1 \leq k \leq 3$ and $n \geq 1$. The language $\{\pi\}$ can be accepted by an ANEPP.*

2. *Given a finite set F of patterns of size (k; l) and (l; k) for all $1 \leq k \leq 3$ and $l \geq 1$, the pattern matching problem with patterns from F can be solved by ANEPPs in O(n+m+kl+k) computational (processing and communication) steps.*

H. Bordihn, P. Bottoni, A. Labella and V. Mitrana: "Solving 2D-Pattern Matching with Networks of Picture Processors", Proceedings of Theory and Practice of Natural Computing - Third International Conference,
157--168, 2014.

7. Problems

- Let F be a finite set of pictures; the picture language F_*^* is the minimal set of pictures such that:

i. $F \subseteq F_*^*$,

ii. If $\pi, \rho \in F_*^*, then\ \pi®\rho \in F_*^*\ and\ \pi©\rho \in F_*^*$ provided that both of them exist

For a given picture π, one can decide whether $\pi \in F_*^*$?

- If all picture of *F* are of the same size, THE PROBLEM IS SOLVED

7. Problems

- **Th. [BBLM – Soft Computing 2016]**
- *Let (k; l) be two positive integers and F be a finite set of pictures of size (k; l). The language* F_*^* *can be decided by ANEPPs in O(n + m + kl) computational (processing and communication) steps.*

H. Bordihn, P. Bottoni, A. Labella, V. Mitrana: "Networks of picture processors as problem solvers", Soft Computing, 2016.

7. Problems

- A picture π is row/column scattered in a picture ϑ if ϑ contains a subpicture ρ such that all the row/column of π appears as row/column of ρ in the same order.

For a given pattern π and an arbitrary picture ϑ one can decide whether π is row/column scattered in ϑ

6. Problems

- The Hamming distance between 2 pictures of the same size is the number of positions at which the corresponding symbols are different.

For a given pattern π, a positive integer k, and an arbitrary picture ϑ one can decide whether ϑ contains a subpicture that is at the Hamming distance of at most k from π?

Thank you very much