NETWORKS OF PICTURE PROCESSORS

Victor Mitrana

joint work with

Henning Bordihn (Potsdam)

Paolo Bottoni (Rome)

Anna Labella (Rome)

CONTENTS

GENERAL IDEA

Is the classic architecture sufficient?

Make the computation in parallel

Distribute the computation

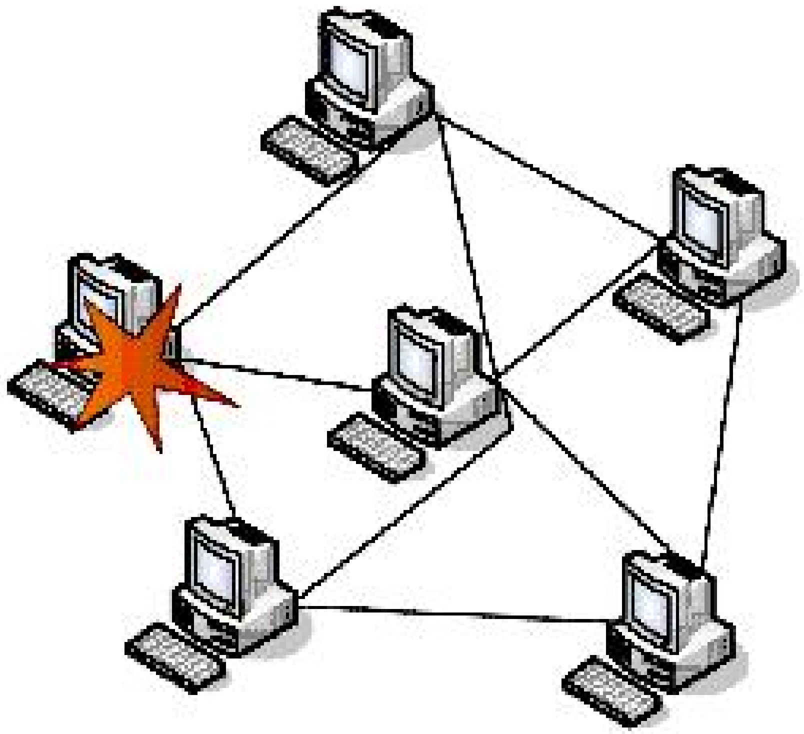
Approximate the solution

Change the "ingredients": natural computing
-Genetic algorithms, neural networks
-Quantum computing
-Molecular computing
and many others

GENERAL IDEA

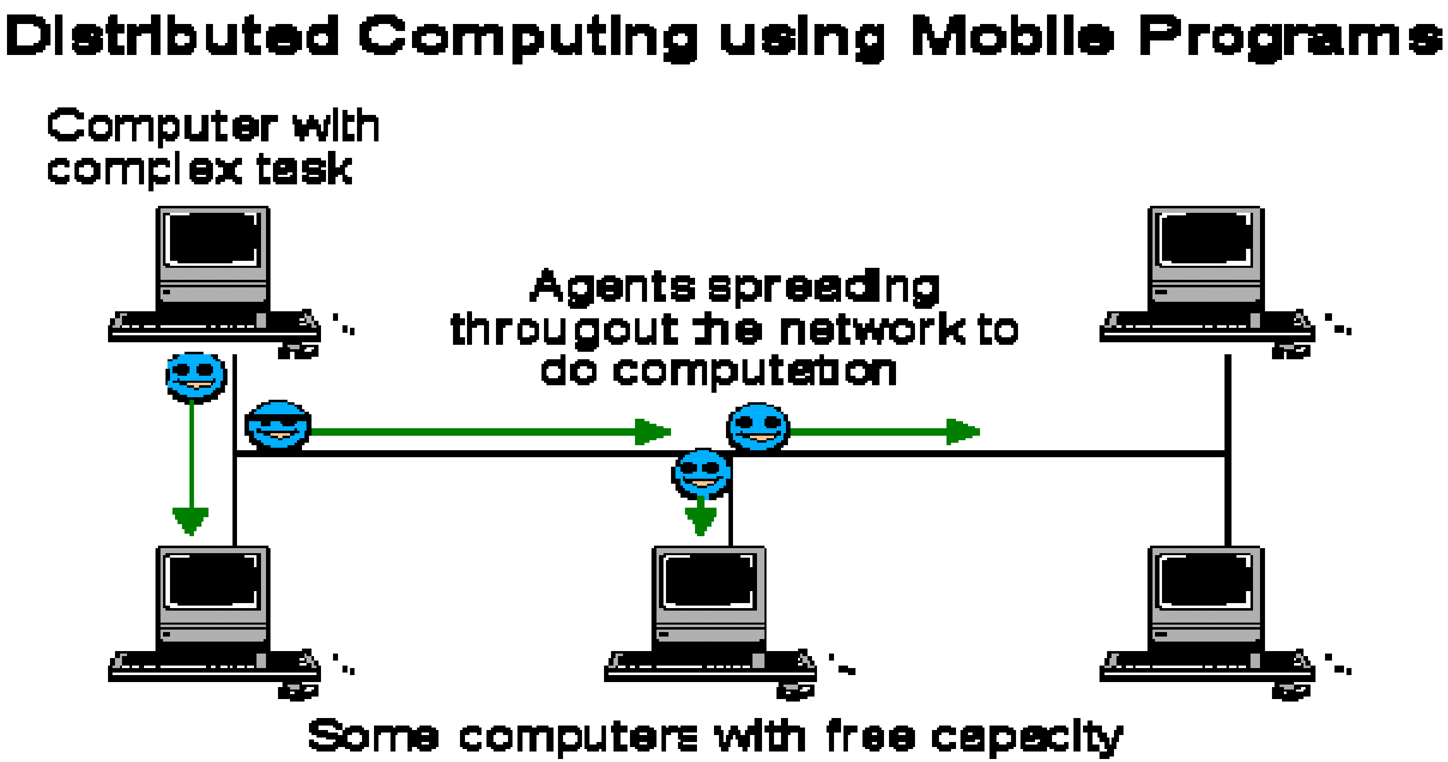
Distributed computing: networks of computer-like devices that can exchange large messages with their neighbors and perform arbitrary local computations.

Supercomputers are more powerful, but errors or break downs are disastrous



Distributed computing systems are made up of many systems; so if one crashes, others are unaffected

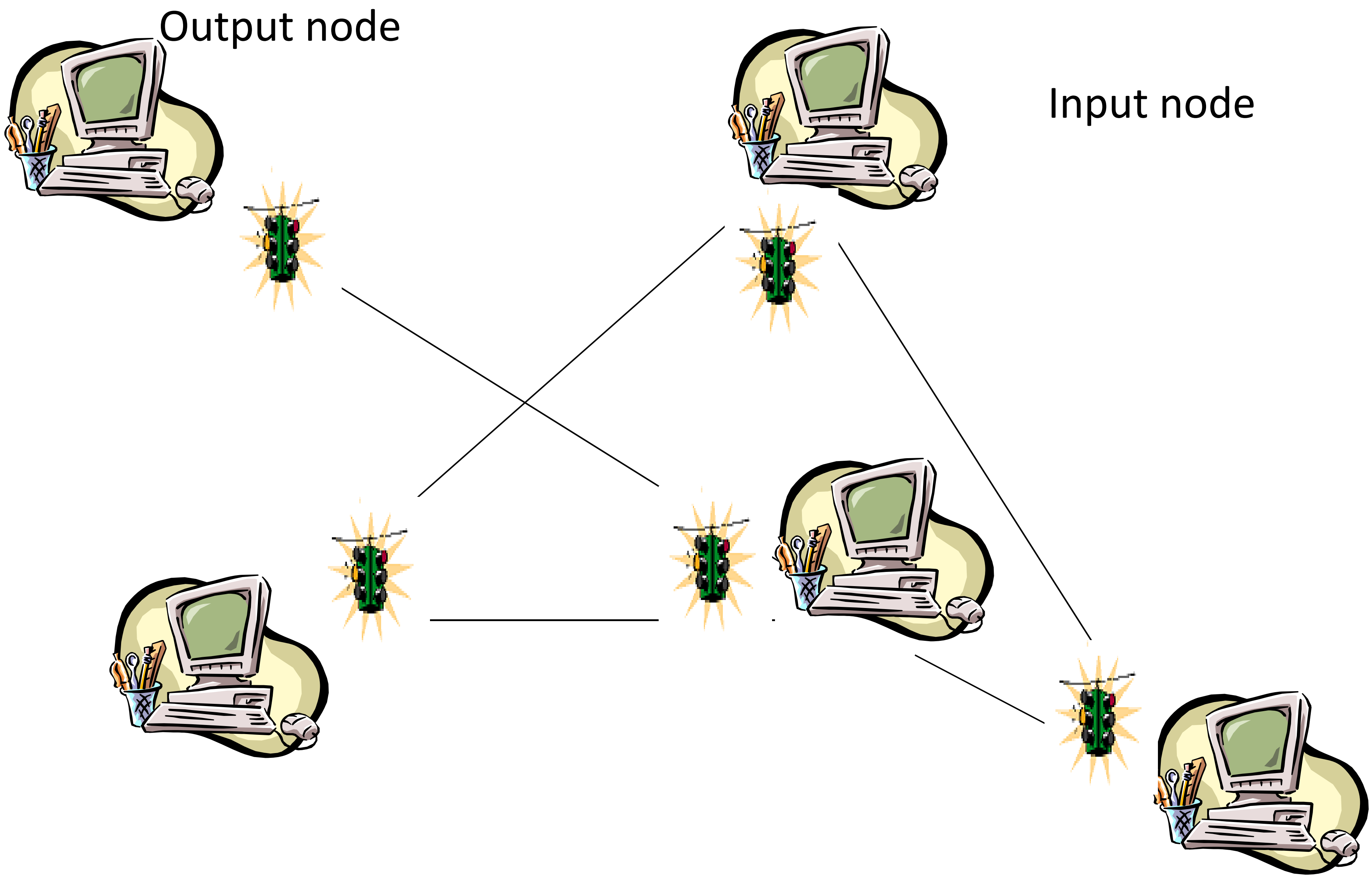
GENERAL IDEA



-Processors having limited memory and computational abilities.
-Processors having no a priori knowledge of their location.
-Each processor has only a limited, incomplete view of the system.

AMORPHOUS COMPUTING

A FUNNY (IM)POSSIBLE WAY OF SOLVING PROBLEMS



GENERAL IDEA

A set of nodes that are connected: a **network**
Each node: **simple processor**
Three distinguished nodes: **In, Halt** and **Accept**

The edges: **bidirectional communication channels**
Information: **strings, *pictures*, graphs**
Restricted communication by: **input/output filters.**

Input: a picture in ***In***.
Computation: Processing, Communication, ...
Halting: a picture enters ***Halt***
Acceptance: when the computation halts, ***Accept*** is nonempty

GENERAL IDEA

Goal: To apply distributed computing methods to networks of sub-microprocessor devices, e.g., biological cellular networks or networks of nano-devices.

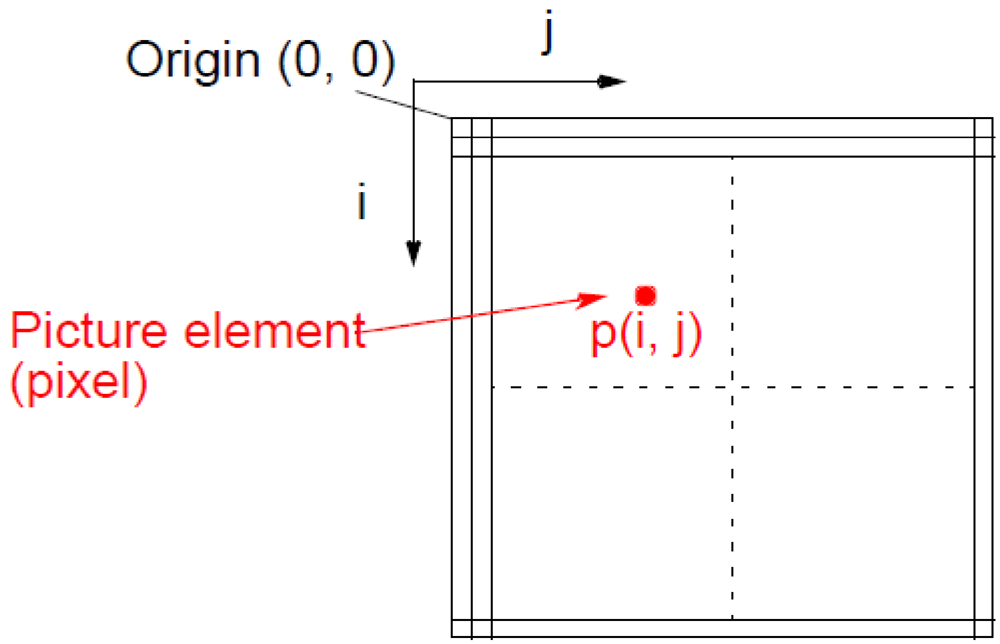
Question: do tiny bio/nano nodes "compute" and/or "communicate" essentially the same as a computer?

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: ?

IMAGES AS RECTANGULAR PICTURES

Stored image consists of two-dimensional array of *pixels* (picture elements):



Many low-level image-processing operations assume monochrome images and refer to pixels as having *gray level* values or *intensities*.

PICTURES

π =

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

ρ =

a	b	b	b	b	a	b
b	b	a	b	a	a	b
c	b	a	b	a	a	c
b	b	a	b	a	a	b

π © ρ

a	b	b	a	c	a	b	b	b	b	a	b
b	b	a	a	b	b	b	a	b	a	a	b
c	b	a	a	a	c	b	a	b	a	a	c
b	b	a	a	a	b	b	a	b	a	a	b

π ® ρ

OPERATIONS ON PICTURES

EVOLUTIONARY OPERATIONS AND ACTIONS:

CSubstitution: a $\rightarrow$ b (|)

$\sigma^*(\pi)$

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

a	b	b	a	c
b	b	a	a	b
c	b	a	b	a
b	b	a	a	a

b	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

OPERATIONS ON PICTURES

EVOLUTIONARY OPERATIONS AND ACTIONS:

CSubstitution: a $\rightarrow$ b (|) $\sigma^{\leftarrow}(\pi)$

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

b	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

OPERATIONS ON PICTURES

EVOLUTIONARY OPERATIONS AND ACTIONS:

CSubstitution:a $\rightarrow$ b (|)

$\sigma^{\rightarrow}(\pi)$

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

a	b	b	a	c
b	b	a	a	b
c	b	a	a	b
b	b	a	a	a

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	b

OPERATIONS ON PICTURES

EVOLUTIONARY OPERATIONS AND ACTIONS:

RDeletion: a $\rightarrow \lambda$ (-)

$\sigma^*(\pi)$

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

a	b	b	a	c
b	b	a	a	b
b	b	a	a	a

b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

OPERATIONS ON PICTURES

EVOLUTIONARY OPERATIONS AND ACTIONS:

RDeletion: a $\rightarrow$ λ (-) $\sigma^{\uparrow}(\pi)$

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

OPERATIONS ON PICTURES

EVOLUTIONARY OPERATIONS AND ACTIONS:

RDeletion:a $\rightarrow \lambda$ (-) $\sigma^{\downarrow}(\pi)$

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a

FORMAL DEFINITIONS

FILTERS:

$$\varphi^{(s)}(w;P,F) \equiv P \subseteq alph(w) \quad \wedge \quad F \cap alph(w)=\emptyset.$$

$$\varphi^{(w)}(w;P,F) \equiv alph(w) \cap P \neq \emptyset \quad \wedge \quad F \cap alph(w)=\emptyset.$$

$$\varphi^{\beta}(L,P,F)=\{w\in L : \varphi^{\beta}(w;P,F)\}.$$

FORMAL DEFINITIONS

EVOLUTIONARY PICTURE PROCESSOR: (M,PI,FI,PO,FO)

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

$G=(X_G, E_G)$: **underlying graph structure**

$\mathcal{N}: X_G \rightarrow EP_V$: **associated picture processors**

$\alpha : X_G \rightarrow \{*,\leftarrow,\rightarrow,\downarrow,\uparrow\}$: **action mode**

$\beta : X_G \rightarrow \{s,w\}$: **filter type**

$\rho_x(.)=\varphi^{\beta(x)}(.; PI_x, FI_x)$: **input filter**

$\tau_x(.)=\varphi^{\beta(x)}(.; PO_x, FO_x)$: **output filter**

FORMAL DEFINITIONS

WORKING MODE

Evolutionary step:

$C \Rightarrow C'$, iff $C'(x) = M_x^{\alpha}(x)(C(x))$

Communication step:

C ⏩ C' iff

$$C'(x) = (C(x) - \tau_x(C(x))) \cup \bigcup_{\{x,y\} \in EG} (\tau_y(C(y)) \cap \rho_x(C(y)))$$

LOCAL PICTURE LANGUAGE

#	#	#	#	#	#
#	1	0	0	0	#
#	0	1	0	0	#
#	0	0	1	0	#
#	0	0	0	1	#
#	#	#	#	#	#

LOC = the class of local picture languages

$$\Theta = \left\{ \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline\end{array}, \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 1 & 0 \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & 0 \\ \hline \# & \# \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & 1 \\ \hline \# & 0 \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \# \\ \hline \# & 1 \\ \hline\end{array}, \begin{array}{|c|c|}\hline 1 & \# \\ \hline \# & \# \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \# \\ \hline 0 & \# \\ \hline\end{array}, \begin{array}{|c|c|}\hline 0 & 1 \\ \hline 0 & 0 \\ \hline\end{array}, \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & 0 \\ \hline \# & 0 \\ \hline\end{array}, \begin{array}{|c|c|}\hline 0 & \# \\ \hline 1 & \# \\ \hline\end{array}, \begin{array}{|c|c|}\hline 0 & \# \\ \hline 0 & \# \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \# \\ \hline 1 & 0 \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \# \\ \hline 0 & 0 \\ \hline\end{array}, \begin{array}{|c|c|}\hline 0 & 0 \\ \hline \# & \# \\ \hline\end{array}, \begin{array}{|c|c|}\hline 0 & 1 \\ \hline \# & \# \\ \hline\end{array} \right\}$$

COMPUTATIONAL POWER OF ANEPP

Th. [BLM – Fundamenta Informaticae 2014]

1. *The complement of every local language can be weakly accepted by an ANEPP.*
2. *There exist non-recognizable languages which can be accepted by ANEPPs.*

2-DIMENSIONAL PATTERN MATCHING

Problem. Given a pattern π and a picture θ is π a subpicture of θ?

SOLVING 2D PATTERN MATCHING

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+l) computational (processing and communication) steps.*

FURTHER OPERATIONS ON PICTURES

MASK and UNMASK:

mask↓

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a

FURTHER OPERATIONS ON PICTURES

MASK and UNMASK:

mask↓

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

FURTHER OPERATIONS ON PICTURES

mask↑

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

FURTHER OPERATIONS ON PICTURES

unmask $^{\rightarrow}$

a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

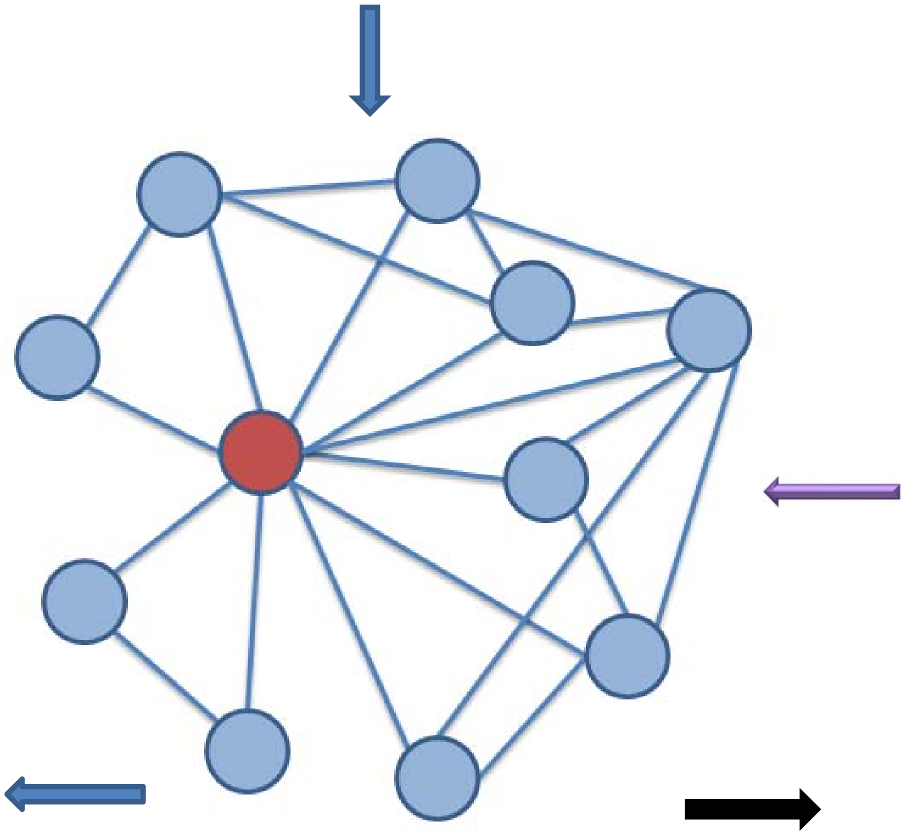
a	b	b	a	c
b	b	a	a	b
c	b	a	a	a
b	b	a	a	a

SOLVING 2D PATTERN MATCHING

Th. [BBLM – TPNC 2014]

Let π *be a picture of size (k; l) for some k,l ≥ 1. The language* $\{\pi\}$ *can be accepted by an ANPP.*

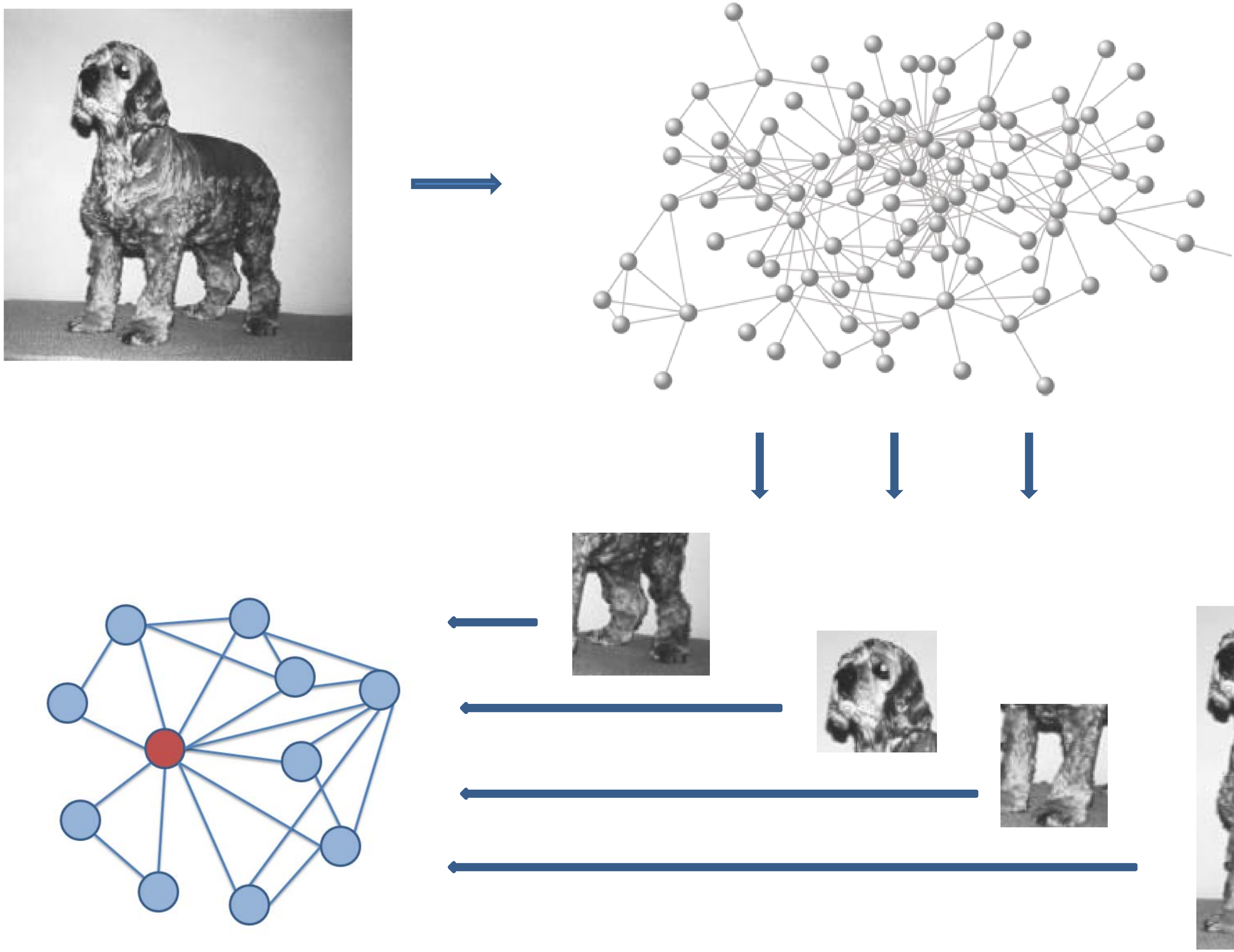
SOLVING 2D PATTERN MATCHING



ACCEPT

REJECT

SOLVING 2D PATTERN MATCHING



SOLVING 2D PATTERN MATCHING

THEOREM [BBLM, TPNC 2014]

Given a finite set *F* of patterns of size (*k*,*l*) for any k,l ≥ 1, the pattern matching problem with patterns from *F* can be solved by ANPPs in $O(n+m+kl+k)$ computational (processing and communication) steps.

CONSEQUENCES

Th. [BBLM – Soft Computing 2016]

1. *Let (k; l) be two positive integers, $1 \leq k \leq 3$ and $l \geq 1$. Every (k; l)-local language or (l; k)-local language can be decided by ANEPPs in $O(n + m + l)$ computational (processing and communication) steps.*
2. *Let (k; l) be two positive integers. Every (k; l)-local language can be decided by ANEPPs in $O(n + m + kl)$ computational (processing and communication) steps.*

CONSEQUENCES

F be a finite set of pictures of size (k, l). The picture language F_*^* is the minimal set of pictures such that:

(i) $F \subseteq F_*^*$,

(ii) If $\pi, \rho \in F_*^*$, then $\pi \circledR \rho \in F_*^*$ (provided that $\pi \circledR \rho$ exists) and $\pi \circledcirc \rho \in F_*^*$ (provided that $\pi \circledcirc \rho$ exists.)

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.*

Thank You