Institute of Computer Science

Academy of Sciences of the Czech Republic

Computability and Non-Computability Issues in Amorphous Computing

Jiří Wiedermann

Academy of Sciences
Charles University
Prague, CZ

IFIP TCS 2012

Partially supported the Czech NSF grant P202/10/1333

Outline

1. Introduction – why amorphous computing systems
2. Informal definition
3. Examples of possible applications
4. Flying amorphous computer
5. Molecularly communicating nano-machines
6. What is the simplest universal computing device?
7. Can Turing machines simulate amorphous computers?
8. Conclusions

Introduction

Why it is interesting to speak about computational and non-computational issues in amorphous computing:

- Amorphous computing systems probably present the simplest universal computing devices
- In order to operate as envisaged some systems make use of non-computational operations
- It appears that some amorphous systems cannot be faithfully simulated by classical Turing machines

Amorphous computing systems differ from classical computing systems almost in almost every aspect:

- They lack any concrete architecture (hence their name)
- Their myriads of processors communicate wirelessly in an aerial or aqueous environment
- The processors' computing and communicating abilities are stripped down to essentials
- The processors can move

A huge collection (of order 10^6 or 10^12 or more, say) of computational particles dispersed irregularly on a surface or throughout a volume creating a self-organizing computing network

Particles:

- have no a priori knowledge about their positions, no identifiers
- are possibly faulty, contain sensors and effectors, might be mobile
- are not synchronized but operate at the same speed
- are identically programmed

Each particle:

- has energy source, modest computing power and modest amount of memory, random number generator, plus sender/receiver apparatus
- communicates on the same channel, to a limited small distance
- is "blind" – does not know, whether there are some neighbors, to communicate with
- can communicate only with nearby neighbors (via radio, chemically, optically, acoustically,…)
- can receive a signal if exactly one sender is on air

Motivation

Scientific fiction:

- The Black Cloud by Fred Hoyle, 1957
- A Deepness in the Sky by Vernor Vinge, 1999

Device Miniaturization

- Technology allows producing very small energy efficient processors (MEMS, NEMS) equipped by sensors, transmitters and a power source
- They can be built in large numbers

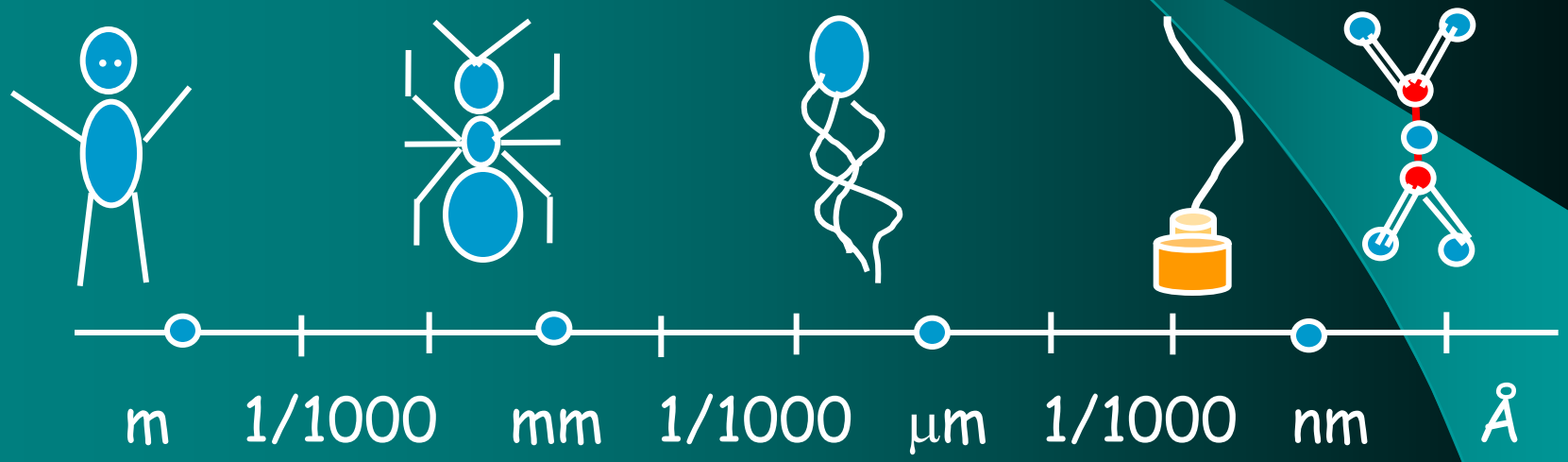
→ Wireless (sensor) networks

→ Amorphous computers (Smart Dust)

Cellular engineering

- Synthetic biology (Biots) – can produce "programmable" bacteria

The Scale



Possible applications:

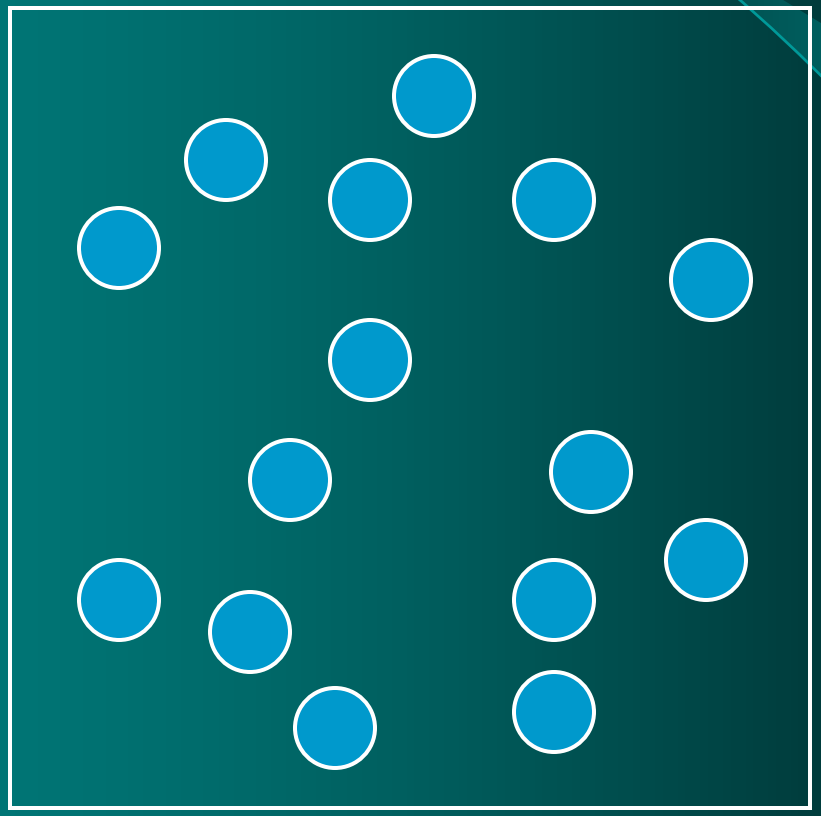
- Traffic control,
- People and item tracking in buildings,
- Monitoring plant yield in agriculture,
- Environment monitoring,
- Weather forecast,
- Ocean, atmosphere and underwater flow,
- Intruder detection,
- Battle field surveillance,
- space prospecting, etc.
- Artificial immune systems in the blood stream
- Gene therapy for non-reproductive cells
- Reversing degenerative disease
- Combating heart disease
- Overcoming cancer
- Slowing-down diseases and aging (living forever?)
- Reversing aging
- Cloning
- A new way of eating
- Redesigning the digestive system
- Programmable blood
- Redesigning the human brain
- Bio-fuel production
- Global warming combat
- Environment pollution clearance
- Soil fertilization

Amorphous Computer

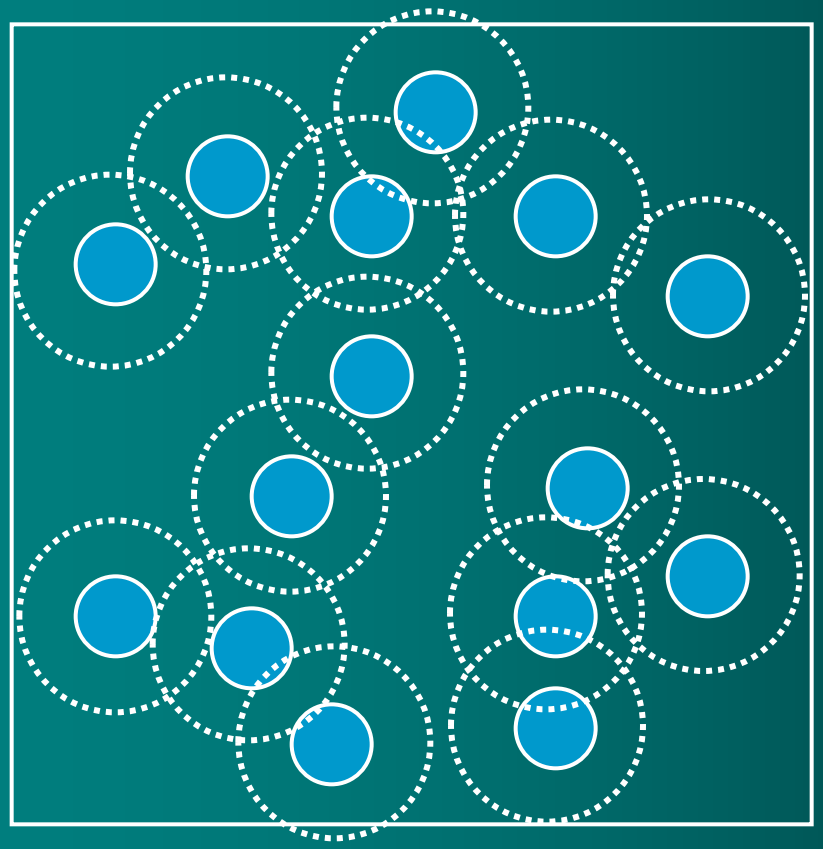
A bag containing a large number of computing elements

Amorphous Computer

Amorphous Computer



Amorphous Computer



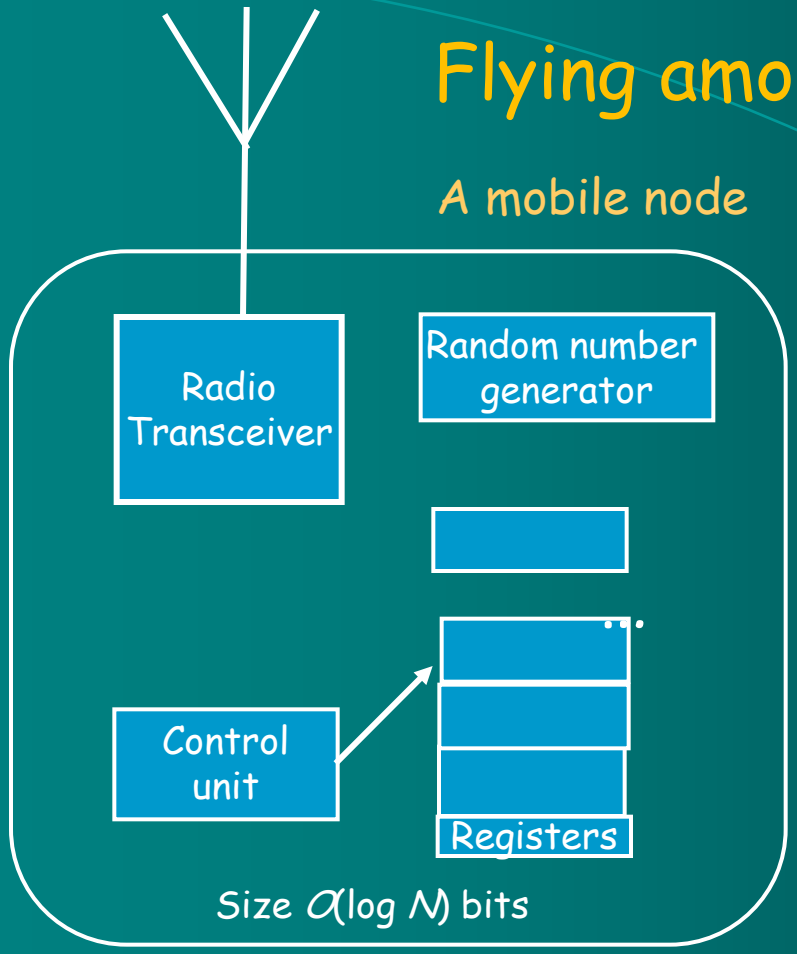
Airborn medium:

"Flying amorphous Computer"

Aqueous medium:

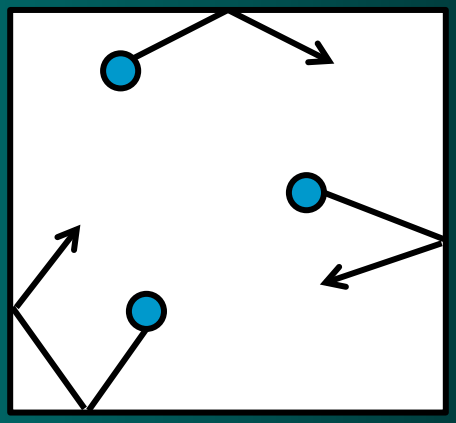
Molecularly communicating Nano-machines

Flying amorphous computer



- Local clock, not synchronized with other nodes
- No identifiers
- Radio transceiver:
  - single channel
  - limited range r
  - No collision detection (a node cannot distinguish the case of no broadcasting from the case of two or more nodes broadcasting simultaneously)

The nodes are mobile and move in a square area in random directions, with a constant speed, bouncing when reaching a boundary:



Two problems:

- a blind communication
- temporarily inaccessible nodes

Solving the problem of blind communication:

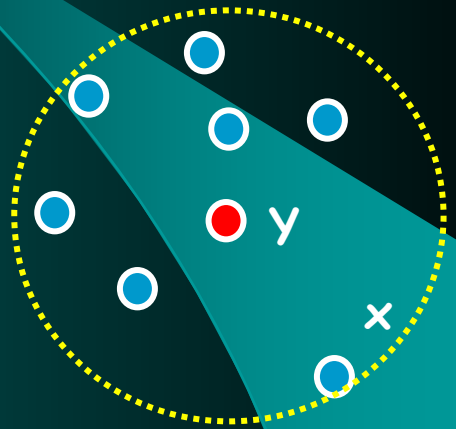
The key ideas of a protocol transmitting a message to a node's neighbours:

- Send sporadically (in order to avoid broadcast conflicts)
- Repeat the sending a few times (this will increase the probability of error-free message delivery)

Analysis: the probability p of sending should depend inversely on the number Q of a node's neighbors and should be repeated k times to handle the case of more processors in a node's neighborhood , with probability ε of failure

$p=1/(Q+1)$

$k= O(Q \log(1/\varepsilon))$



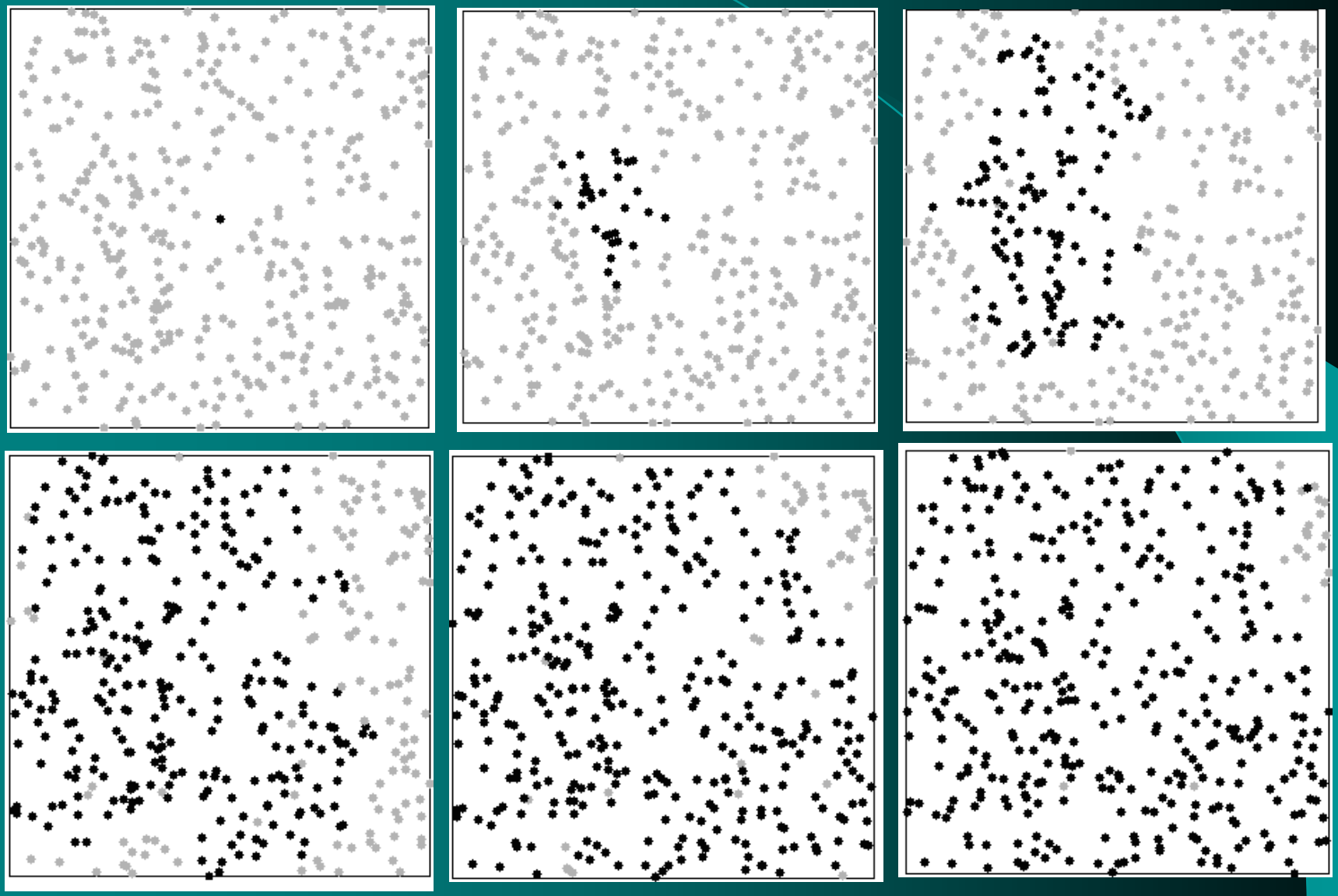
Flooding the network by a message:

After receiving a message from the leader, each nodes keeps re-sending this message until all nodes in the network get flooded by this message

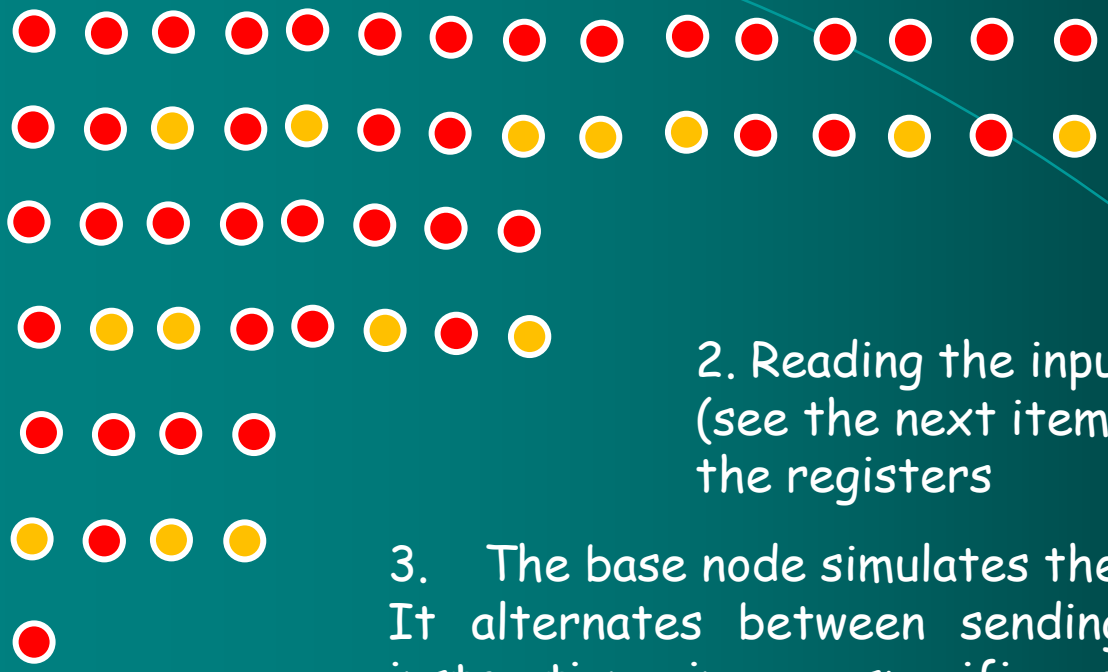
Solving the problem of temporarily inaccessible nodes:

by assumption: we assume that no node remains inaccessible forever

Broadcasting in a flying amorphous computer



Determining a leader by randomized halving of a set



Simulating a RAM

1. Address assignment: repeatedly use the leader selection strategy for assigning addresses 1,2,…N.

2. Reading the input into the first registers (see the next item) and initializing the rest of the registers

3. The base node simulates the individual RAM instructions: It alternates between sending a request to perform an instruction in a specific register and expecting the acknowledgement from it until such an acknowledgement is received (this must happen in a finite time)

Theorem: If the address assignment is successful (and this can be guaranteed with an arbitrary large probability) then the simulation terminates within a finite time and always delivers a correct result.

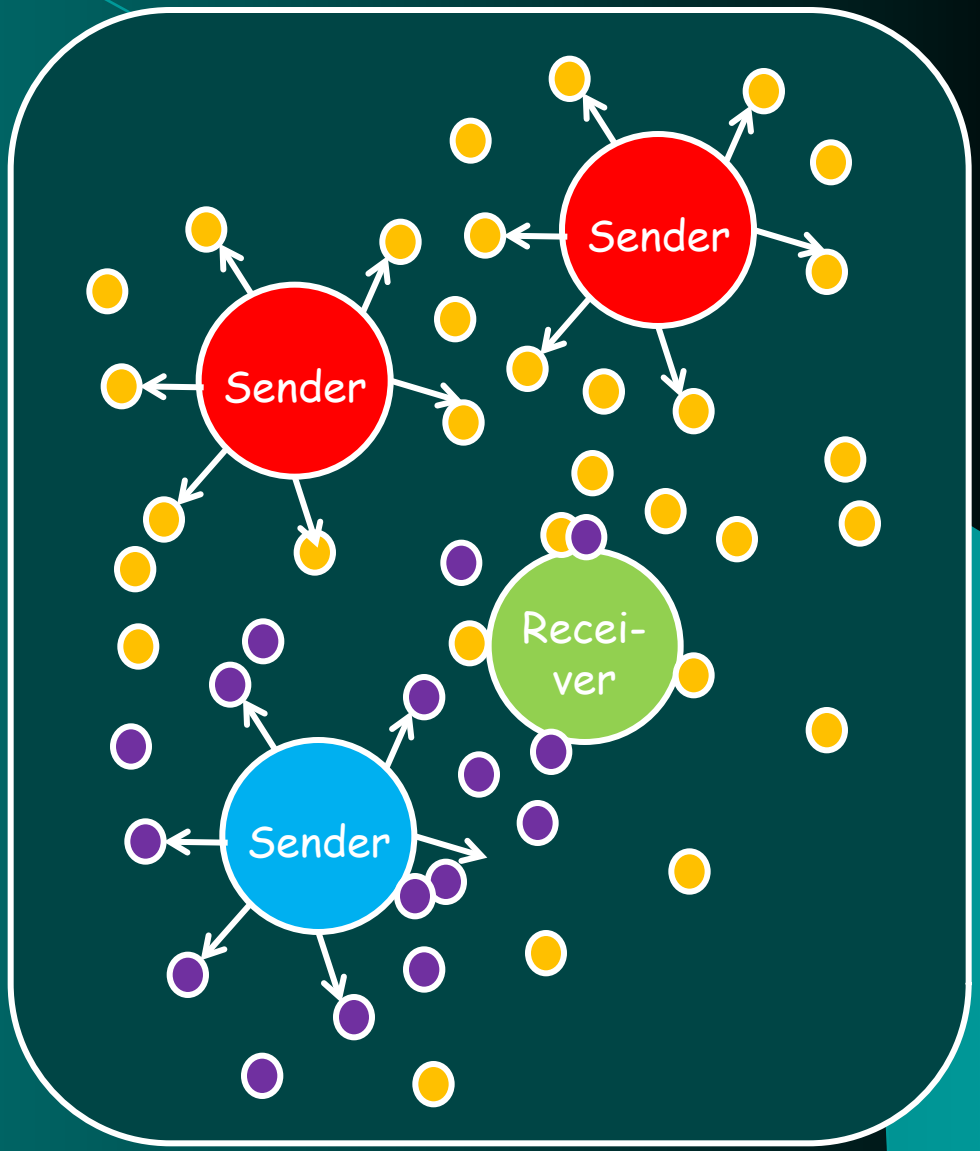
Molecularly communicating mobile nano-machines

- A collection (typically of order at least millions) of embodied computational units freely floating in a closed liquid environment
- molecular cell-sized devices or engineered organisms
- produced by self-assembly or self-replication
- capable of performing simple tasks such as actuation and sensing
- Nano-machines "talk" to each other via molecular communication creating thus a self-organizing computing network called nano-net

Communication	Molecular	Conventional
Carrier	Molecule	Electromagnetic waves
Signal type	Chemical	Digital
Propagation	Slow (by diffusion)	Fast (light speed)
Environment	Aqueous (liquid)	Airborne or cable
Energy costs	Low	High
Reliability	Low (stochastic)	High accuracy

Computational/Communication Scenario:

- A finite number of self-reproducing nanomachines freely moving (actively or passively) in a closed liquid environment
- no external control
- interaction via a special type of molecular communication – quorum sensing

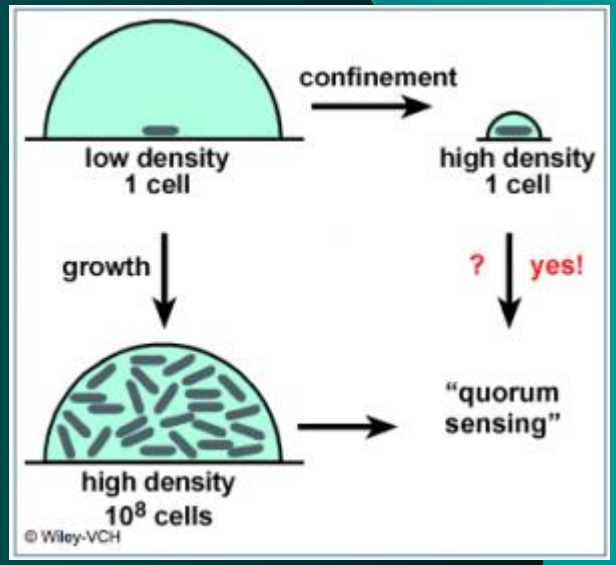


Quorum sensing

- is a type of decision-making process used by decentralized groups to coordinate individual behavior;

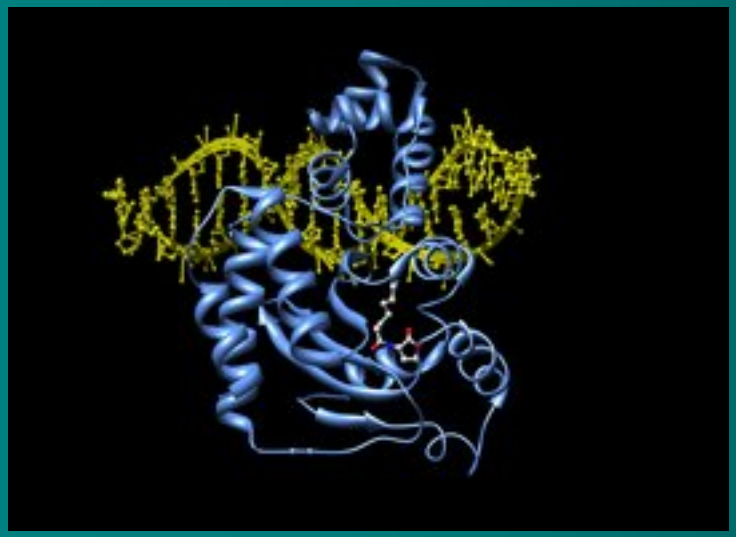
- in our case, quorum sensing is based on individually estimated density of nano-machines in the environment;

- the density of nano-machines is indirectly inferred from the density of signal molecules secreted by the machines into the environment

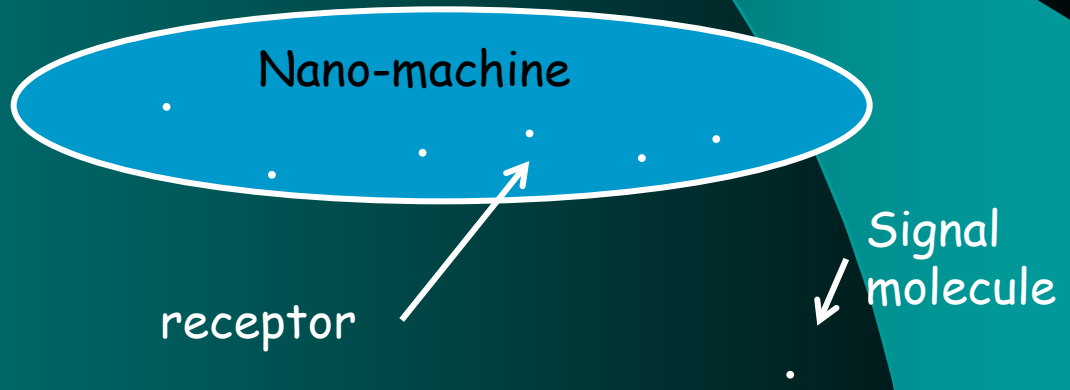


Properties of sensors and signal molecules

- Each sensor is specialized to recognize signal molecules of a certain type
- a nanomachine has hundreds of sensors of each type
- A signal molecule is approx. 1000 times smaller than a nanomachine
- after a certain time, signal molecules disintegrate



A quorum sensing autoinducer bound to its receptor

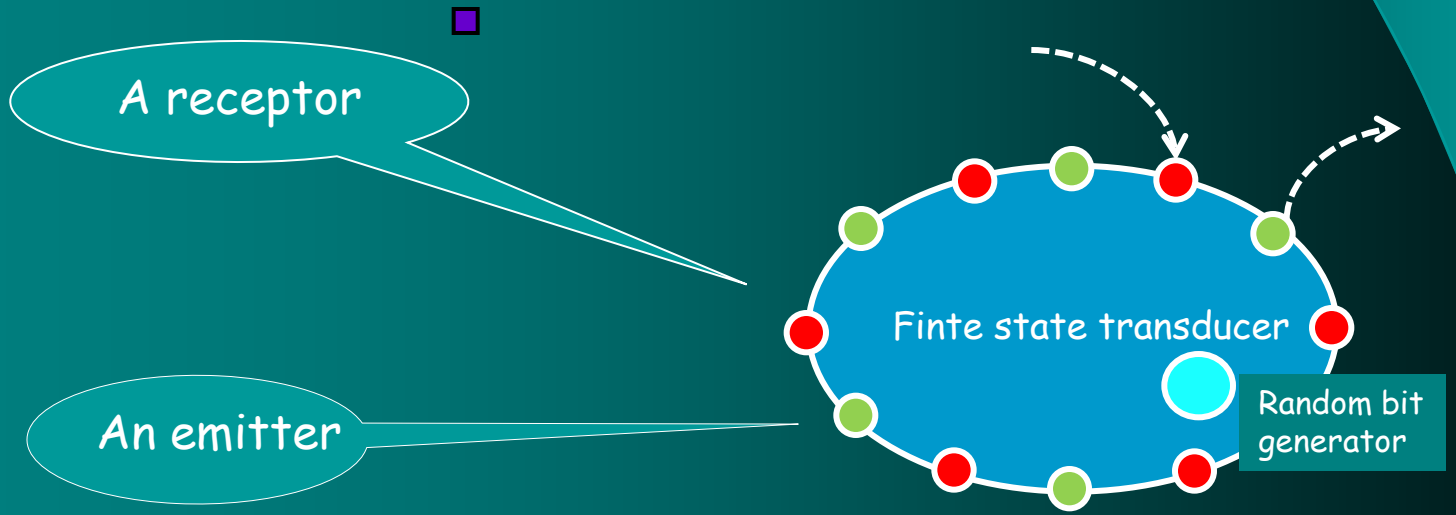


A stochastic contact model

A nano-machine model

1. Body: sensors, receptors, emitters (pores), self-reproducing organs, random bit generator, timers, locomotive organs

2. Computational part – a finite multi-input and multi output (Mealy) automaton
   - Inputs arrive from sensors, receptors and timers
   - Outputs are sent to receptors, effectors (signal molecules emitters, flagella,…) and timers



Growing the population of nano-machines

- n - environment volume
- V – nano-machine volume
- N=n/V is the maximal number of machines in the environment
- v - signal molecule volume
- prior to its self-reproduction, each machine emits O(V/v) signal molecules
- after g generations, there will be ~2^gV/v signal molecules
- if the environment is full of signal molecules, then

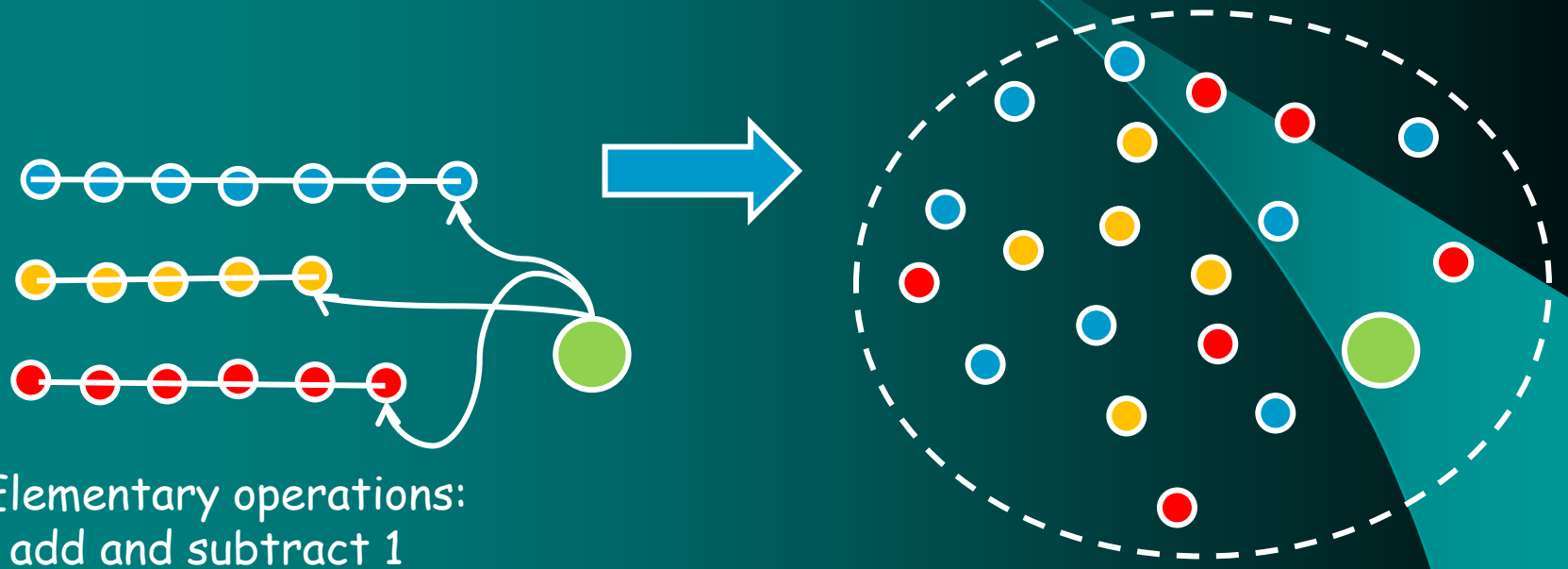
$$2^g V/v = n/v, \text{ i.e., } g = \log N$$

- at that time, with probability approaching 1 each machine detects signal molecules at all its receptors ("quorum sensing") and will stop reproducing

Proposition: in the environment of volume *n* the self-reproduction process of nano-machines, each of volume *V*, will stop with a great probability after *g=log n* generations, for *N=n/V*. At that time there will be approx. *N* nano-machines and *n/v* signal molecules

Distributed computing through nano-machines

1. Raise-up a population of a nano-machines
2. represent the input in unary, i.e., each machines reads a symbol from {0,1} via a special sensor and stores it in its state
3. simulate a counter machine:



Elementary operations:
- add and subtract 1
- test for zero

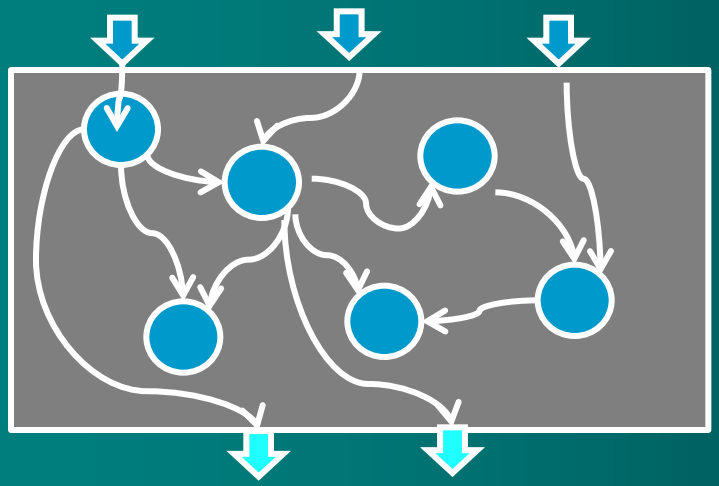
Each operation is done by a series of quorum sensing processes

Theorem: a family of non-uniform nano-nets can correctly simulate a counter machine with a high probability

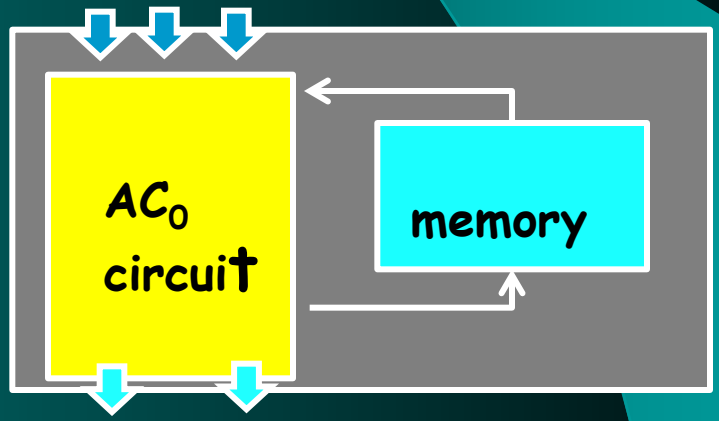
Implementing finite control via circuits

$\Sigma^n \times Q \rightarrow \Gamma^m \times Q$ $\quad f(x_1,\ldots,x_n, q)=(y_1,\ldots,y_m,r)$

Finite state device

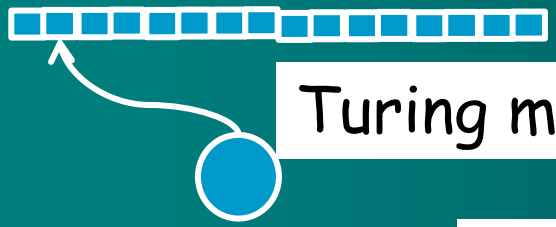


Circuit + memory

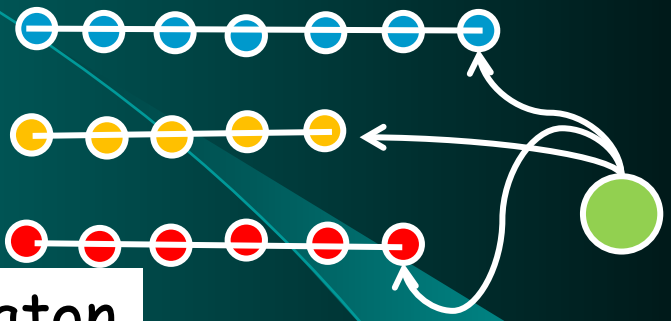


Embodied circuit

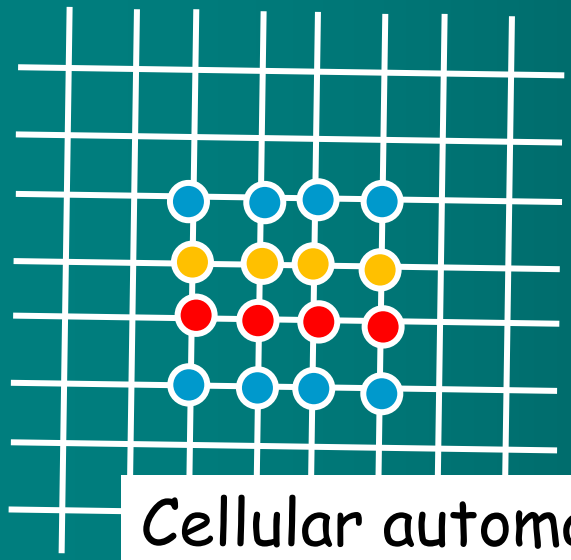
What is the simplest universal computing system?



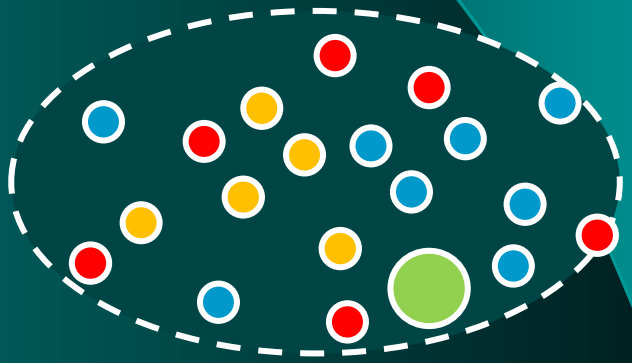
Turing machine



Counter automaton



Cellular automaton

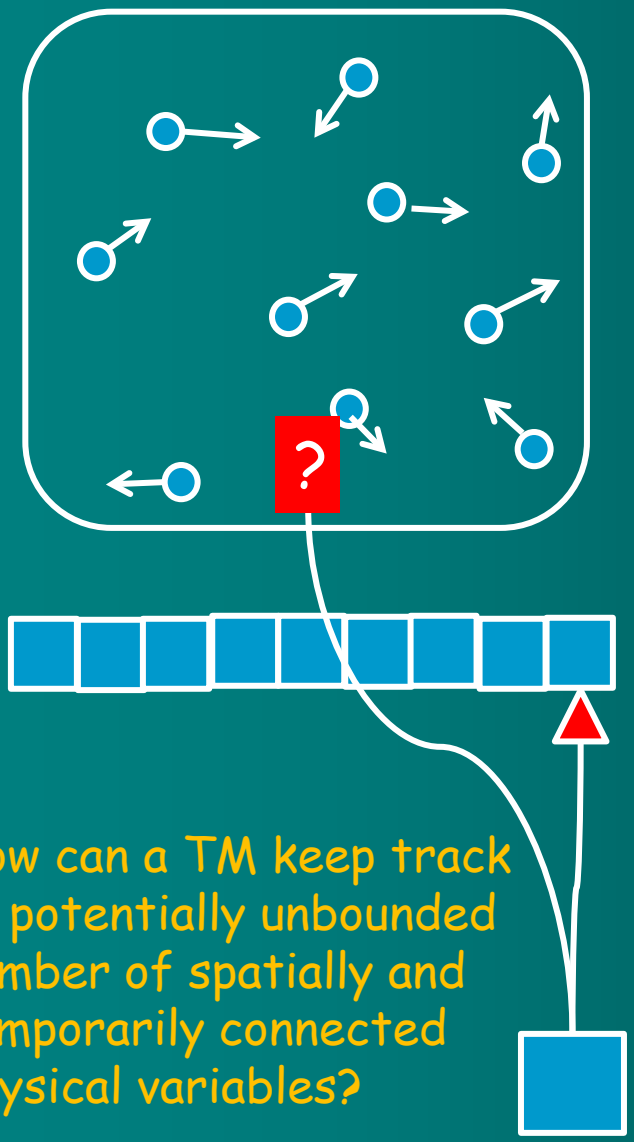


A set of nano-machines

Their computational part is given by a simple circuit, the rest is delegated to the embodiment:

the circuit controls the body actions

Can a Turing machine simulate a flying amorphous computer? A nanonet?



How can a TM keep track of potentially unbounded number of spatially and temporarily connected physical variables?

Scenario: the amorphous computer's processors move around, independently and randomly, read the (changeable) input data from the environment, communicate randomly and compute some function of the data

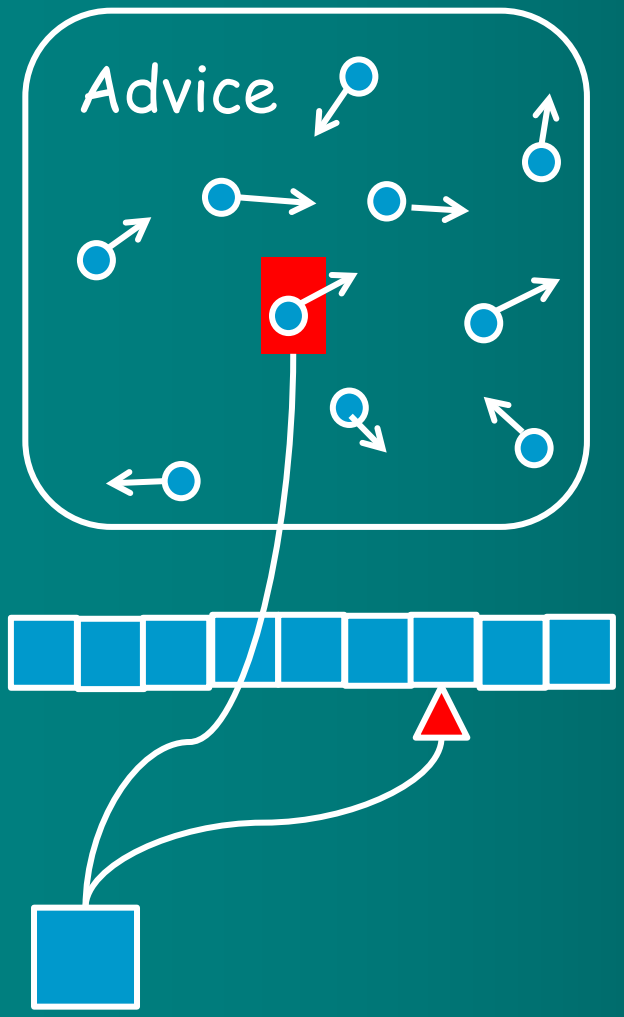
The simulation has to bridge the gap between the physical system and a formal mathematical model

What about non-computational operations:

- self-reproduction
- signal molecules desintegration
- interaction of molecules with their environment

Is this a non-Turing computation?

Simulating a flying amorphous computer with the help of a TM with interactive advice



Scenario: Upon request, the advice returns the configuration of a processor and the message to be heard, if any, and the TM makes the necessary update and sends the new configuration, and a message to be broadcasted, if any, back to the advice. The advice inspects the processors in the order in which the clocks in the processor tick (for simplicity we assume that no ticks occur simultaneously).

Theorem:

1. a flying amorphous computer can be simulated by a TM with interactive advice with high probability.
2. A flying amorphous computer has a super-Turing computing power

Conclusion

- We have presented two models of amorphous computing systems: flying amorphous computers and embodied nano-machines
- We have shown the universal computing power of families of such computers
- Embodied nano-machines are among the simplest computational devices possessing universal computing power
- In order to operate as envisaged, embodied amorphous computing systems make use of classically non-computational operations; hence they cannot be simulated by classical Turing machines; is this a counterexample to the Church-Turing Thesis?
- However, they can be simulated by TMs with advice – they have super-Turing power

Further research is needed in order to better understand the paradigm of amorphous computing and their computational power

The End