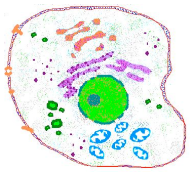
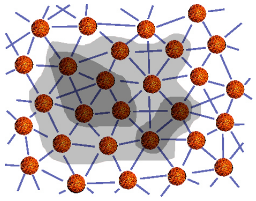
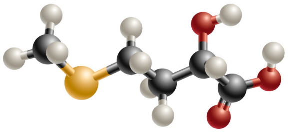
Christof Teuscher

# Chemical Blending with Particles, Cells, and Artificial Chemistries







University of California, San Diego (UCSD)
Department of Cognitive Science
Natural Computation Group
christof@teuscher.ch | www.teuscher.ch | www.ucsd.edu

## “For every complex problem, there is a solution that is simple, neat, and wrong.”

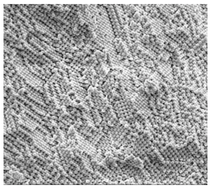
Henry Louis Mencken
1880 - 1956
(The most prominent newspaperman, book reviewer, and political commentator of his day.)

# Disclaimer and Warnings

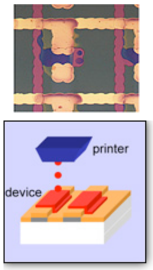
- “…wild and crazy ideas…” (UPP description)
- Provocative visions
- Experimental playground and framework

- Continuation of some of the ideas as first expressed in my PhD thesis, (*Amorphous Membrane Blending: From Regular to Irregular Cellular Machines)*
- Ongoing “high-risk” research with many many open questions

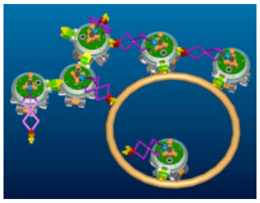
# Some Key Challenges



**Nanotech Molecular electronics**



**Printable circuits**



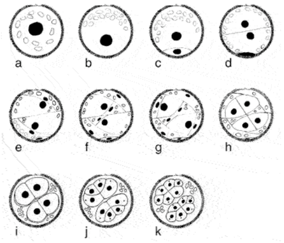
**Collective robotics Swarm intelligence**

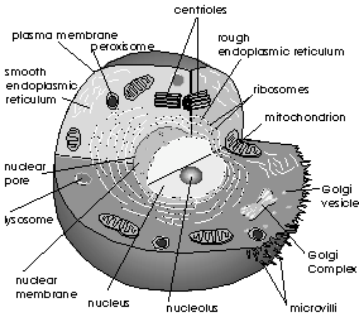
**Sensor networks Smart dust**

**Man-machine trade-off:** **machines perform poor where biological organisms perform well and vice versa.**

# Problem Areas

- Learning, adaptation
- Programming fine-grained massively parallel systems
- Scalability (hardware and software)
- Dynamical hierarchies
- Developmental systems

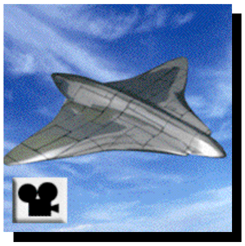




# Why New Paradigms?

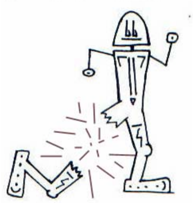
- Little hope in traditional approaches. Where is the progress?
- Brooks, *Nature*, 2001: “Something must be wrong”
- Example: The von Neumann architecture is not appropriate for biologically-inspired computing approaches.
  - Separation in passive memory and active processor
  - Sequential paradigm

Traditional approaches



# Goals

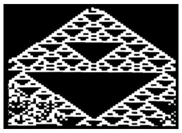
- Unconventional, reconfigurable, adaptable, cellular, biologically-inspired computing paradigm
  - Support hierarchical organizations
  - Highly robust
  - Scalable
  - Unconventional “programming” and ”adaptation” paradigm
  - Ultimately:
    - Low cost
    - Low power
    - Dense



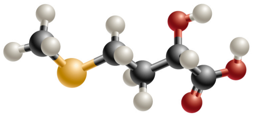
# Concepts and Metaphors

- Massive parallelism
  - Divide and conquer
  - Redundancy
- Locality and simplicity
  - Scalability, inexpensive
- No topological constraints
  - Scalability, inexpensive
  - Easy to build
  - Interesting properties
- Asynchrony
  - Scalability
- Chemical computing and membranes
  - Redundancy and hierarchies

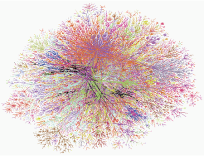
# Related Work and “Idol” Projects



Cellular automata

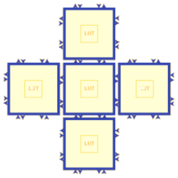


Artificial chemistries



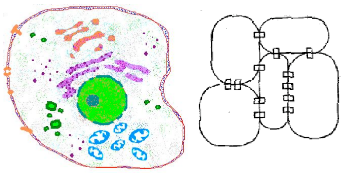
Complex networks

CellMatrix FPGAs

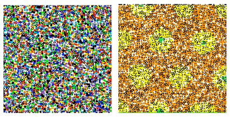


Blob computing

Membrane computing



Amorphous computing



# Learning and adaptation

- I don’t believe in “programming” when it comes to smart, adaptive machines.
  - It’s unlikely that someone will “program” a vision system
  - It’s unlikely that someone will “program” an excellent GO player
  - Too complex and too many unconsidered cases
- Programming ⇔ **Learning**
- Learning exists in many forms
- Alternatives
  - Self-organization
  - Evolutionary algorithms, etc.
- “Unconventional learning method”

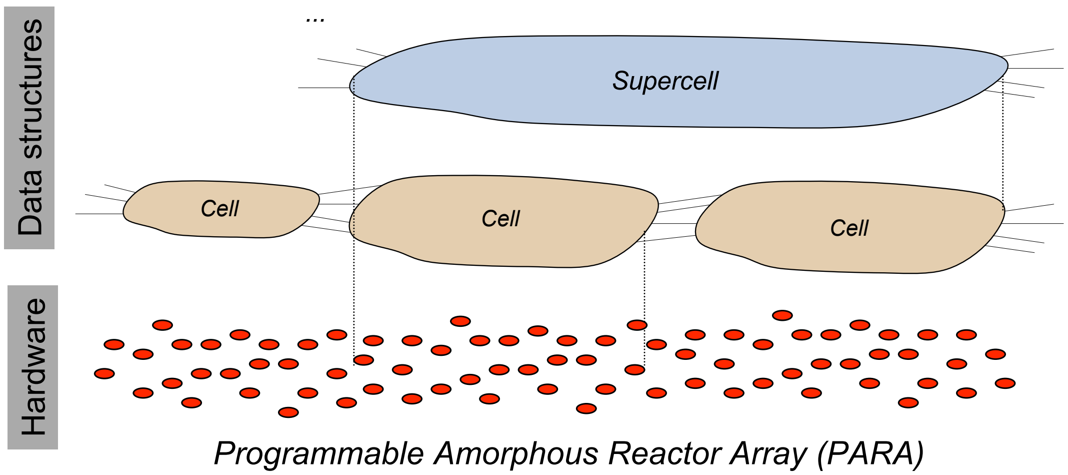
# The Hardware

# Programmable Amorphous Reactor Array (PARA)

- **Cellular computing paradigm**: simplicity, vast parallelism, locality
- **SMASH**: Simple, Many, And Self-Healing

- Open questions:
  - Share the same program?
  - Developmental processes?

# Overview



Data structures

Hardware

...

Supercell

Cell

Cell

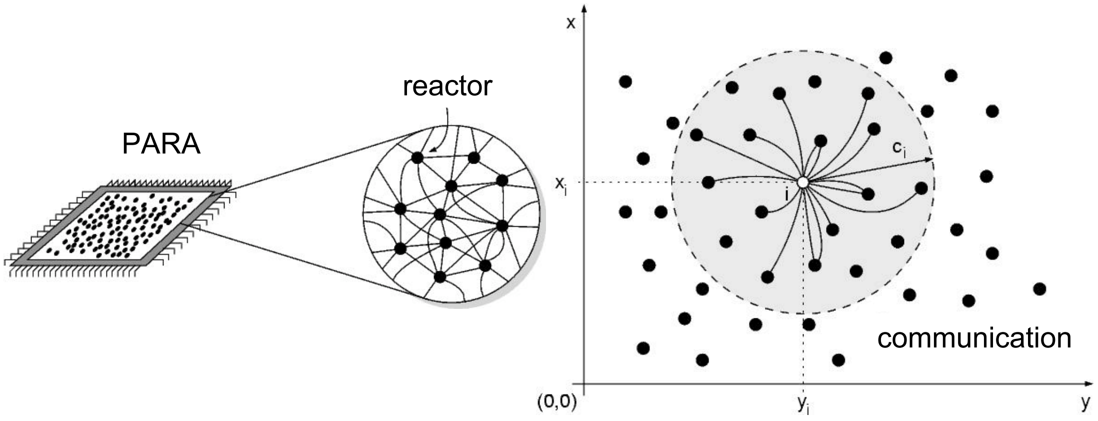
Cell

*Programmable Amorphous Reactor Array (PARA)*

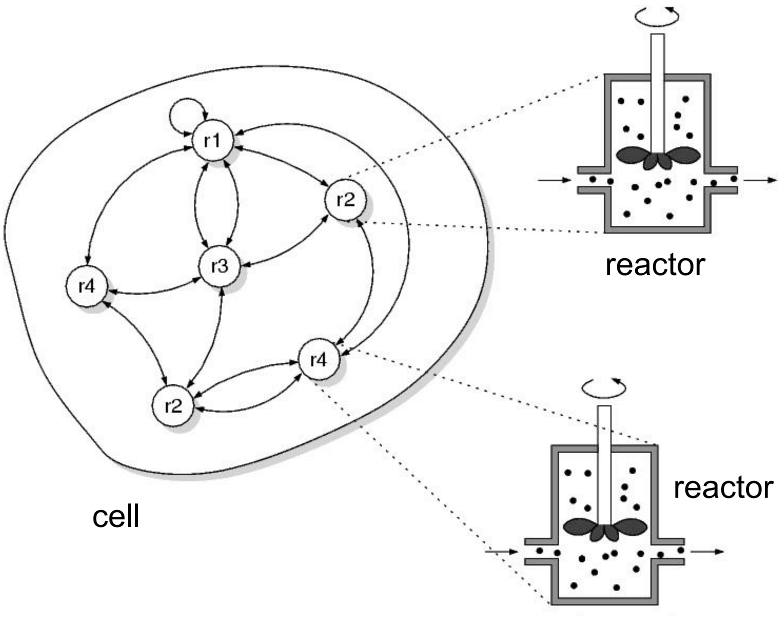
Imperfect computational substrate

# Programmable Amorphous Reactor Array

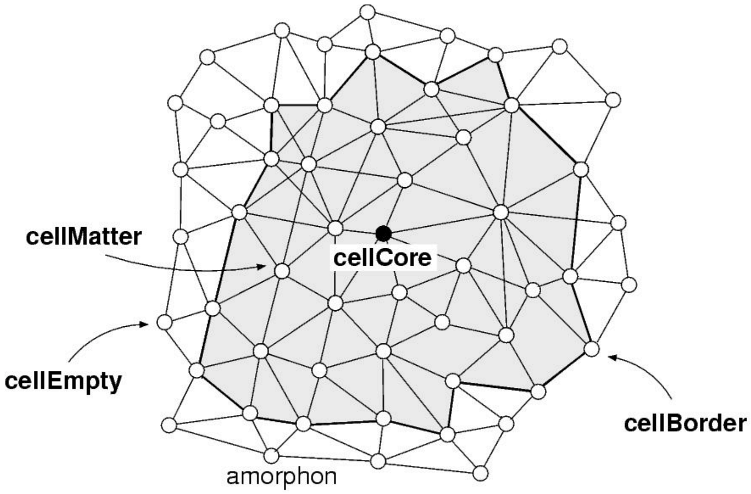
- Example:
  - 2D particle arrangement
  - Random neighborhood with average (redundant and dense) connectivity
  - Particles and links don't move, but they are unreliable



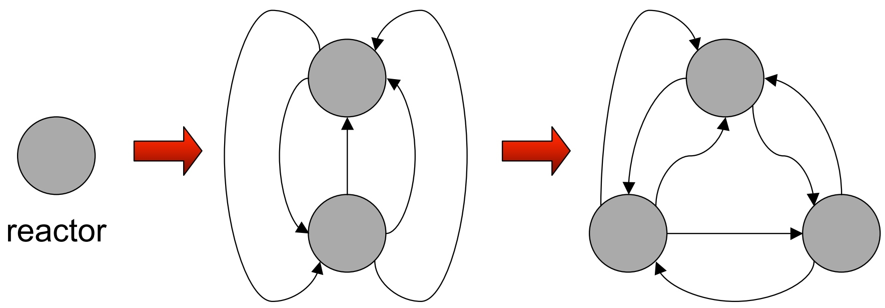
# Programmable Amorphous Reactor Array



# Programmable Amorphous Reactor Array



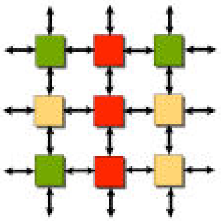
# Adding New Resources



Reactors have a limited capacity

# Two Extreme Network Topologies

Regular Networks

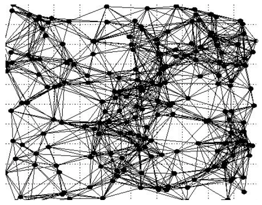


Highly **clustered** and **long path lengths**

Small-World Nets

Random Networks

(Erdös-Rényi Networks)

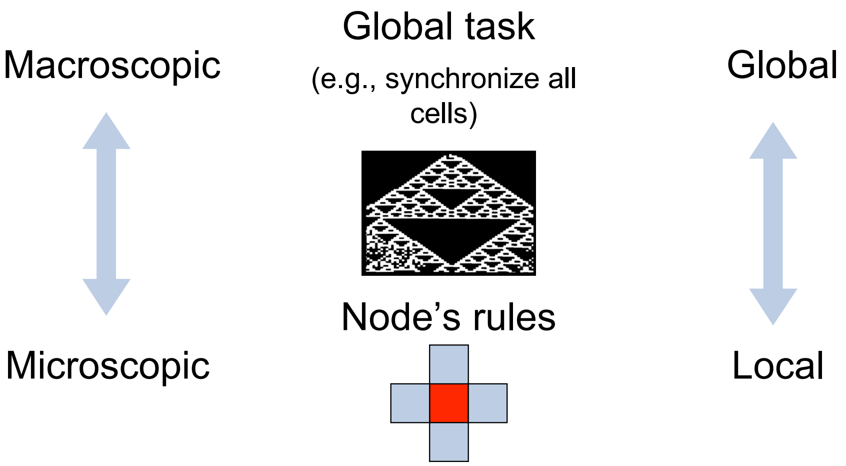


Highly **un-clustered,** and **short path lengths**

Increasing Randomness

C(p): ratio between the actual number of edges between each set of neighbors and the maximum number c, averaged over all units in the system.

# Microscopic versus Macroscopic



Macroscopic

Microscopic

Global task

(e.g., synchronize all cells)

Node's rules

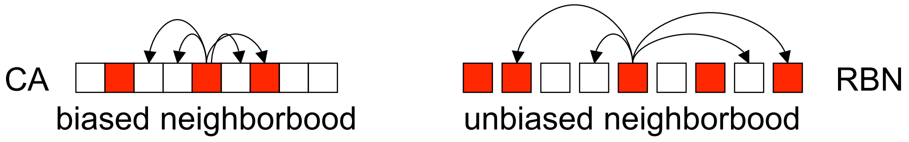
Global

Local

Deducing the local rules from a desired global behavior is a nontrivial task!

# Random Boolean Nets: From Global to Local

- Analytical approach to find the rules of a RBN for the "global" density and the synchronization task.
- Random Boolean nets perform much better than CAs
- In contrast to other work, we can make convergence and quality estimates.
- The network can be randomly rewired during operation.
- Key point: relationship between the global system state and the cell's neighborhood.



CA

biased neighborborood

unbiased neighborborood

RBN

# PARA Research Questions

- An arbitrary interconnection topology is possible.
- What is the best topology (regular…small-world…random) for a given problem and environment?
- Average network interconnectivity K?
- Number of nodes (reactors) required?

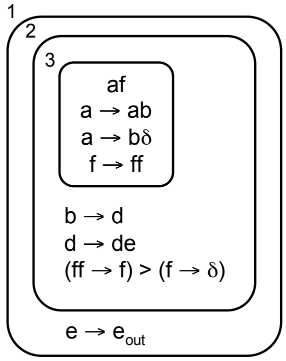
- Constraints (contradicting):
  - Minimize traffic
  - Maximize robustness (against node and link failures)
  - Minimize resources used

# Cells and Membranes

# Cells and Membrane Systems

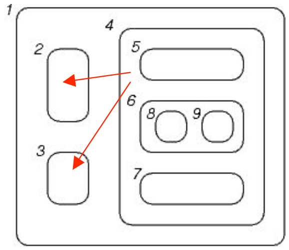
- Represent data structures
- Abstract computing machine
- Move data structures around
- Create hierarchical structures

- Get inspiration from membrane systems and make use of the existing theory



# Artificial Chemistry Inspired by P Systems

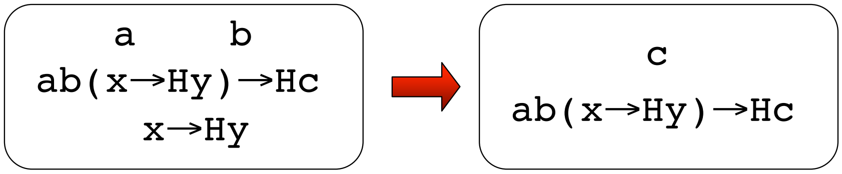
- Implementation and hardware friendly (not guided by theory)
- Well-stirred stochastic reactor
- Rules can also be rewritten
  - Change “program”
  - More flexibility
- Wide variety of “special operators”
- Different communication structures



# Artificial Chemistry

- Molecules: $\mathbf{V} = \{a,b,c,d,\ldots\} \cup \mathbf{R}$
- Rules: $\mathbf{R} = u \rightarrow S v$
- Special symbol: S = {H=Here, L=Leave,...}
- Algorithm: Well-stirred stochastic reactor

- Example:
  - V = {a,b}
  - R = {x→Hy, ab(x→Hy)→Hc}



a b

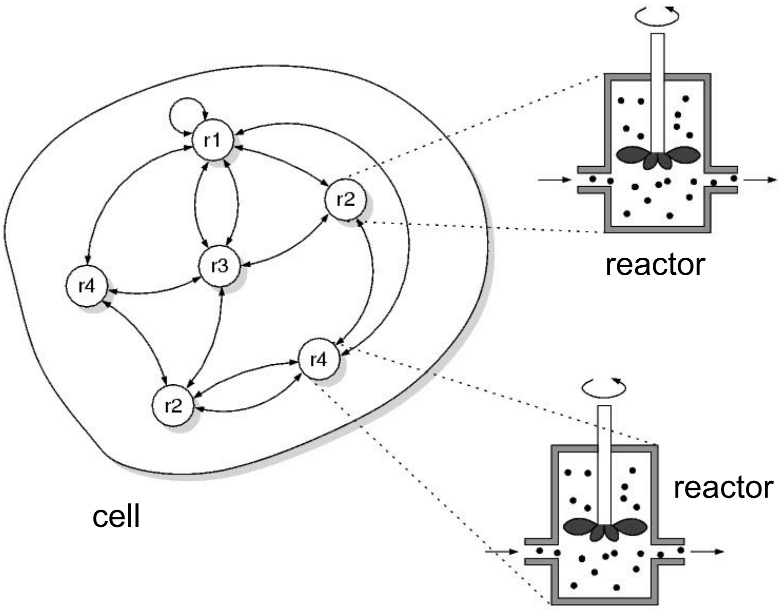
ab(x→Hy)→Hc

x→Hy

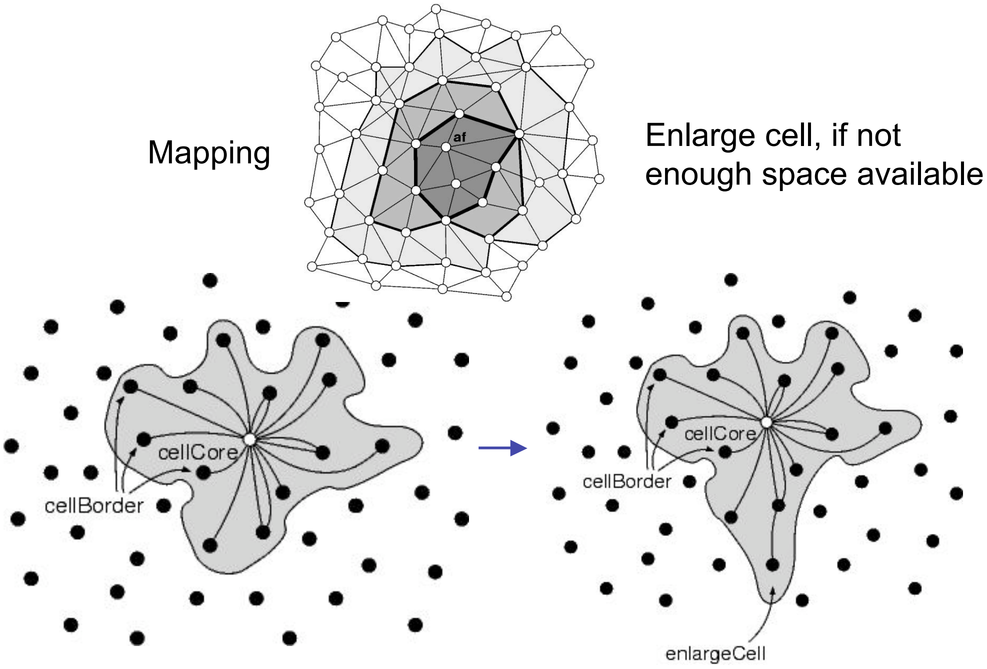
c

ab(x→Hy)→Hc

# Programmable Amorphous Reactor Array



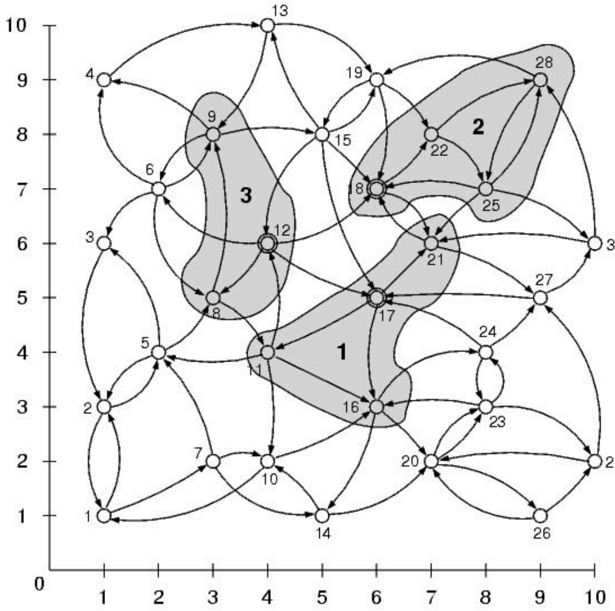
# Example: Enlarge Cells

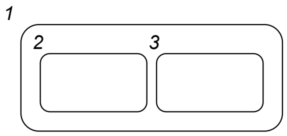


Mapping

Enlarge cell, if not enough space available

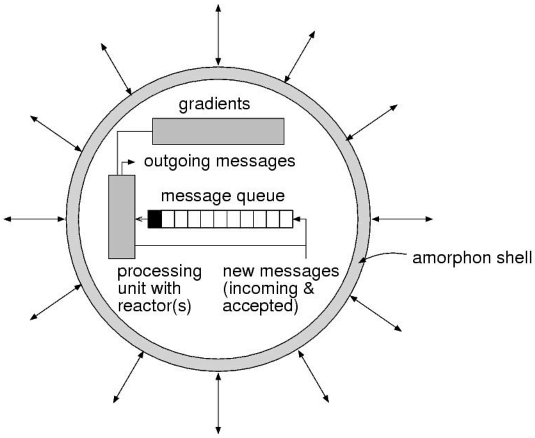
# “Flat” Cells





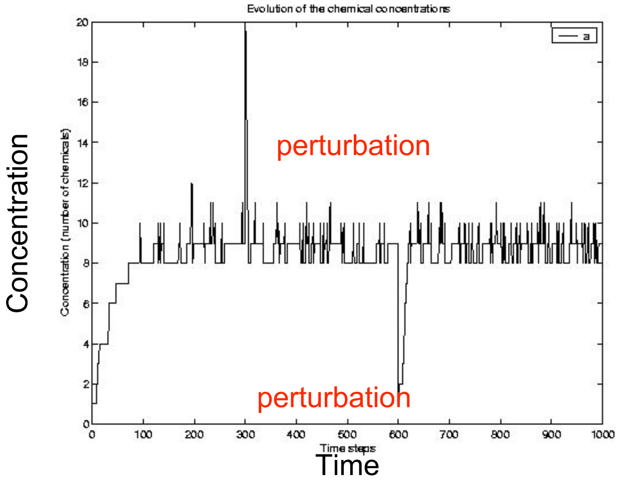
# Communication

- Distributed reactor network
- Message-based communication (message queue, FIFO)
- Messages are guided (self-routed) by gradients:
  - Broadcast message to neighbors
  - Follow gradient

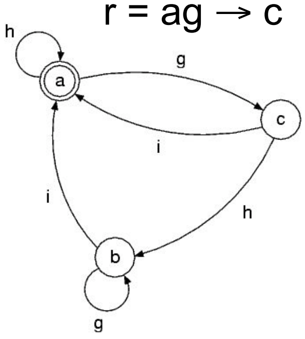

gradients
outgoing messages
message queue
processing unit with reactor(s)
new messages (incoming & accepted)
amorphon shell

# Chemical State Machine

- Can me made very robust!



$r = ag \rightarrow c$



m

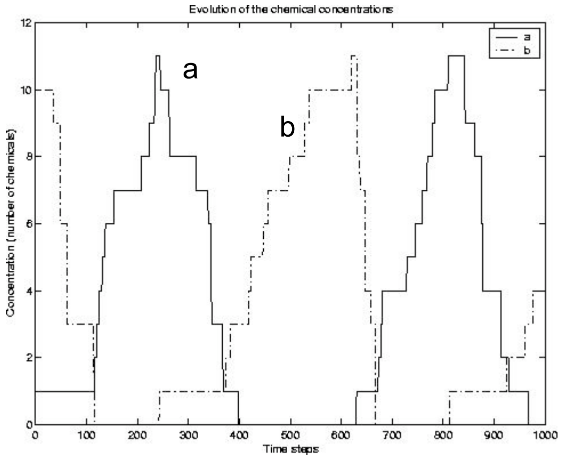
n

$r_1 = a \rightarrow a^2$

$r_2 = a^m \rightarrow a^n$

# Chemical Oscillator

Concentration



Time

$$r_1 = xb^2 \rightarrow x$$
$$r_2 = xb \rightarrow x$$
$$r_3 = ya^2 \rightarrow y$$
$$r_4 = ya \rightarrow y$$
$$r_5 = xa \rightarrow xa^2$$
$$r_6 = yb \rightarrow yb^2$$
$$r_7 = xa^{10} \rightarrow ya^{10}$$
$$r_8 = yb^{10} \rightarrow xb^{10}$$

x,y catalysts

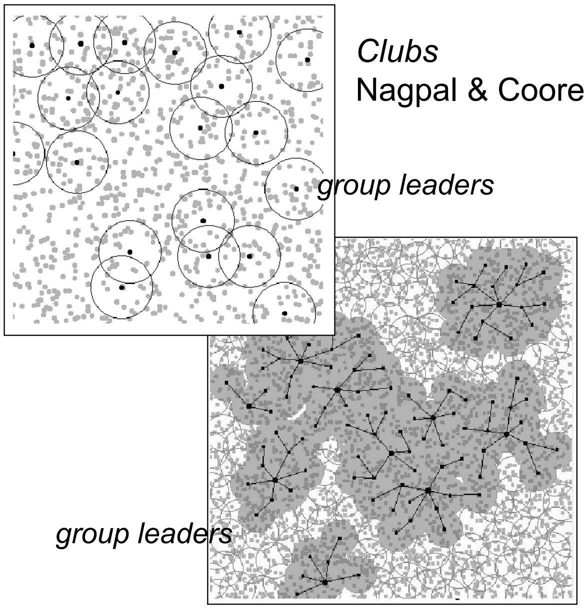
# Groups and Hierarchies

# Groups and Hierarchies

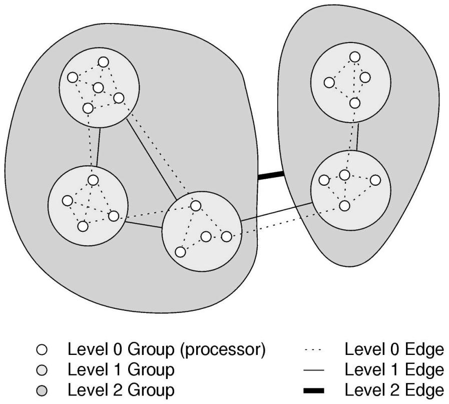
- Optimize limited resources
- Reduce complexity
- Minimize the overhead of control
- Redundancy
- Abstraction (simplification for programming, analysis, etc.)

- Amorphous computer, Blob computing, etc.: close relationship between the spatial structure and data structures.

# Groups and Hierarchies

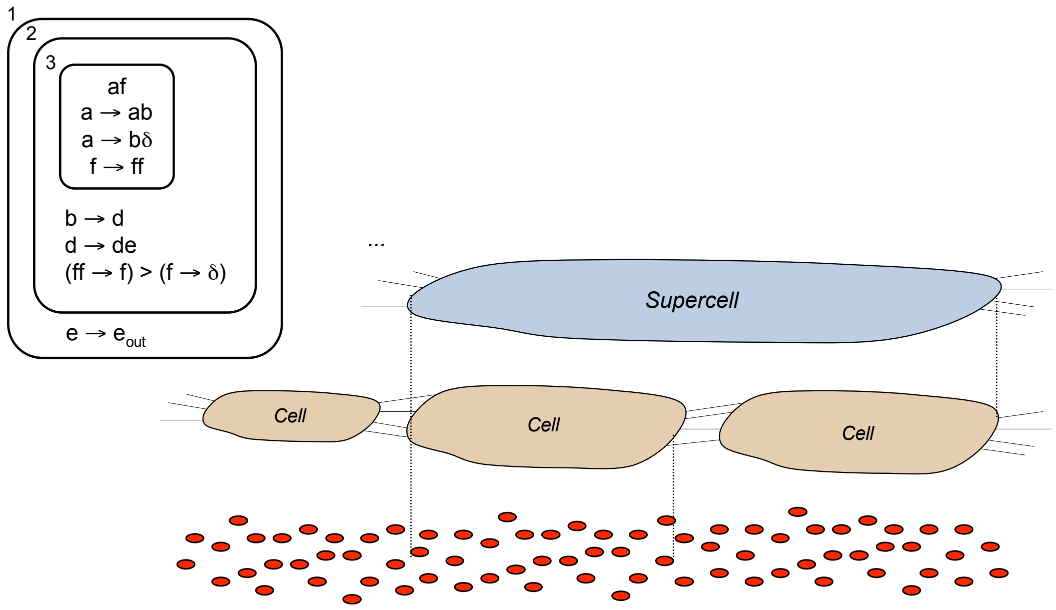


*Clubs*
Nagpal & Coore



*AC hierarchy*
D. Coore et al., 1997

# Cellular Hierarchy



1

2

3

af

a → ab

a → bδ

f → ff

b → d

d → de

(ff → f) > (f → δ)

e → $e_{out}$

# Chemical Blending

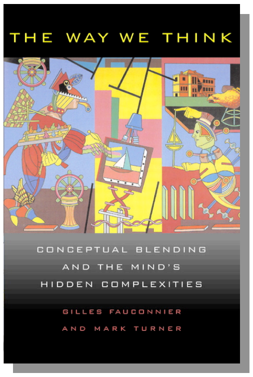
# How to Program or «Create» an Artificial Chemistry

- By hand?
- Compiler?
- Evolutionary algorithms?
- …?

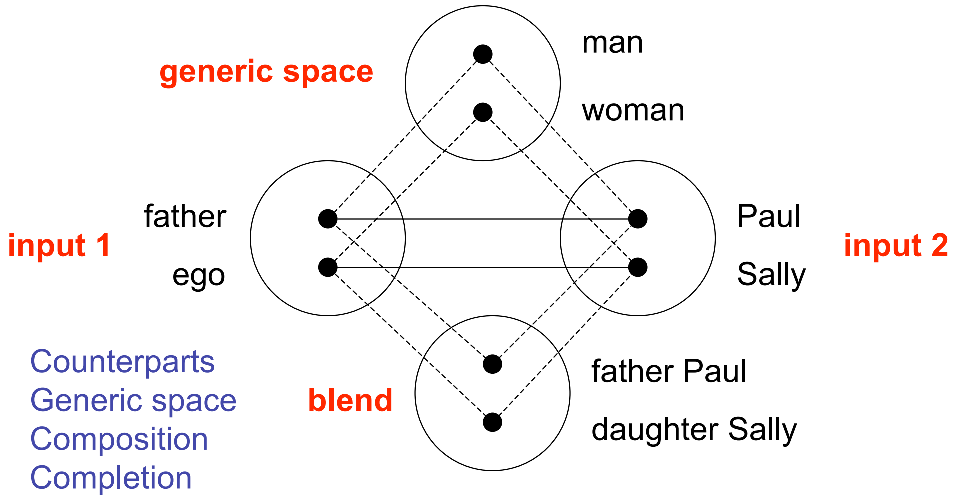
# Conceptual Integration (Blending)

**Gilles Fauconnier & Mark Turner**

- Theoretical framework for exploring human information integration.
- Main questions: How do we creatively think? How do we get new ideas?
- Set of unifying processes, central, uniform, and pervasive.
- No explicit computational framework.
- Chemical blending does NOT copy blending!
  - Does not deal with concepts (math.: set of elements with relations among them)
  - Process oriented (instead of knowledge oriented)
- “Blending-Inspiration”
- Computational blending:
  - Goguen, Harrell, Pereira, etc.



# Blending Example



1. Counterparts
2. Generic space
3. Composition
4. Completion
5. Elaboration (“mental” simulation)
6. Optimality principles

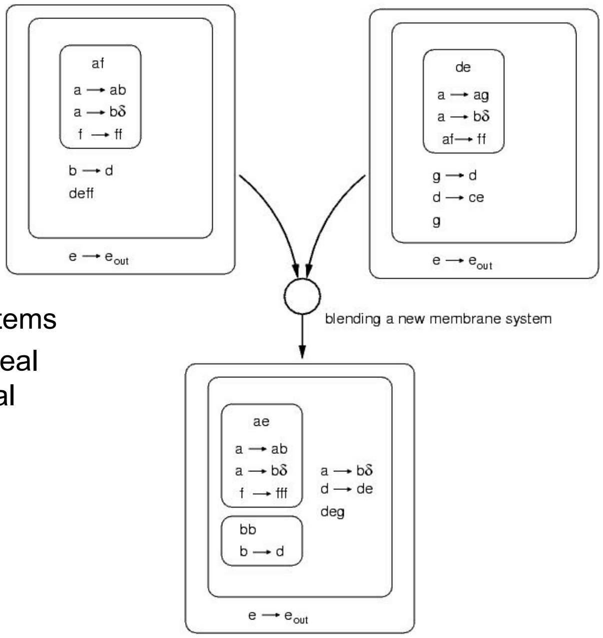
# Questions About «Novelty»

- What is “novelty” (computer science)?
  - An unexplored point in the state space?
- How can we “create” novelty (on the fly)?
  - Random process?
  - In a “smarter” way?
  - Useful for machine learning, games, language, active poetry, etc.
  - New words: houseboat, roadkill, etc.
- Computational creativity
- Learning = exploring new “spaces” = it’s all about novelty and creativity!

# Learning versus Evolution versus Blending

- Evolution:
  - Random and possibly large changes to individuals
  - Uses selection to drive the population towards a global goal.
- Learning:
  - Makes small changes in individuals
  - Many variants
- Blending:
  - Fauconnier and Turner claim that evolution is a form of blending as well (…but they also claim that blending is *everywhere).*
  - Works on the individual level and there is no population

# Chemical Blending Vision



af

$a \rightarrow ab$

$a \rightarrow b\delta$

$f \rightarrow ff$

$b \rightarrow d$

deff

$e \rightarrow e_{out}$

de

$a \rightarrow ag$

$a \rightarrow b\delta$

$af \rightarrow ff$

$g \rightarrow d$

$d \rightarrow ce$

g

$e \rightarrow e_{out}$

blending a new membrane system

ae

$a \rightarrow ab$

$a \rightarrow b\delta$

$f \rightarrow fff$

bb

$b \rightarrow d$

$a \rightarrow b\delta$

$d \rightarrow de$

deg

$e \rightarrow e_{out}$

- Adaptation of membrane systems
- An artificial chemistry is an ideal environment for computational blending
  - Composition ✔
  - Completion ✔
  - Elaboration ✔

# Analogies Used

| **Blending** |  | **Chemical Blending** |
|---|---|---|
| Mental space | ⇔ | Membrane system (P system) |
| Object | ⇔ | Molecule |
| Rule | ⇔ | Reaction |
| Composition | ⇔ | Compose chemicals |
| Completion | ⇔ | Complete chemicals |
| Elaboration | ⇔ | Run the membrane system |

- Mathematically, a mental space is a set of elements with relations among them

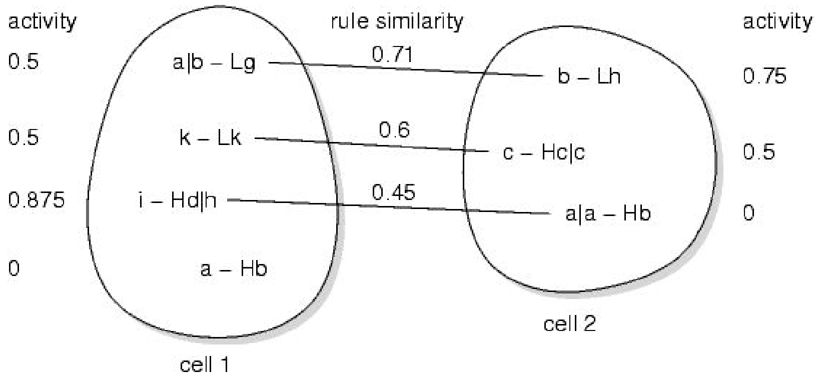
# Chemical Blending

- house + boad = houseboat
- road + kill = roadkill
- dhhsdkjfk + jkkkaula = ??

- We need external measures and evaluation!

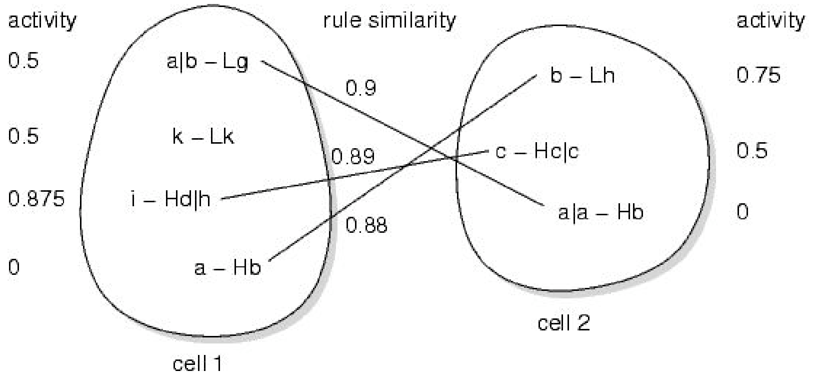
- Hypothesis (example):
  - **Activity**: Active reactions and molecules are considered more useful. Less active elements are selected with lower probability.
  - **Similarity**: The more similar/opposite two objects are, the more likely they are combined together

# Example: C-Blending Initial Mapping

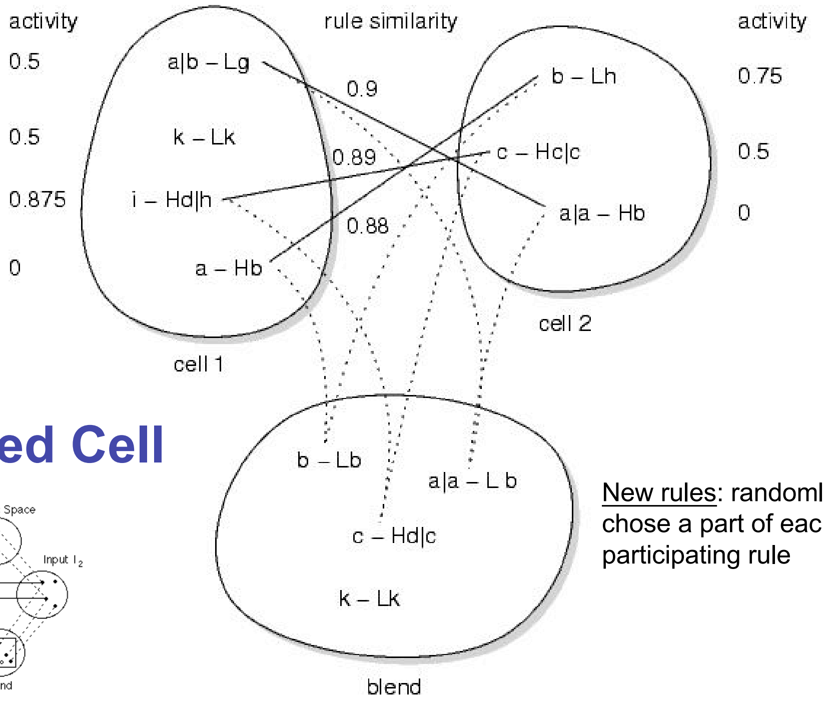


**Input space 1** **Input space 2**

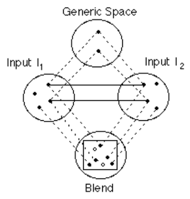
# Example: C-Blending Improved Mapping



Can be easily implemented in stochastic reactor network



# Blended Cell



New rules: randomly chose a part of each participating rule

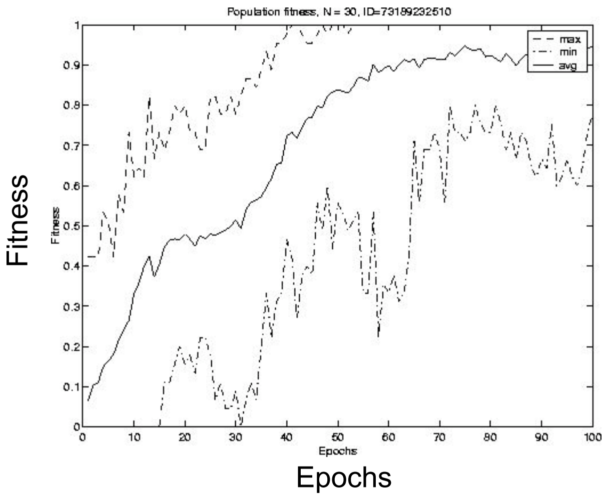
# Survival of the Fittest?

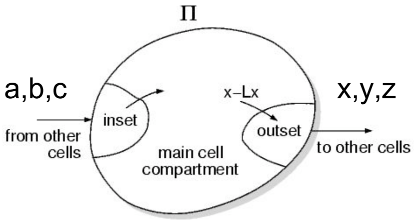
- How do we evaluate new molecules/rules and cells?
- Less active elements are selected with lower probability
- Introduce a “fitness” measure. Credit assignment problem?

- Replace crossover and mutation in a GA by blending?
- Make “smooth” and “smart” changes

## Blending Toy Example: Pattern Recognition

- Alphabet = {a,b,c,x,y,z}
- Mapping: a → x, b → y, c → z (toy example!)
- Inputs: {$a^5$, $b^5$, $c^5$}, Outputs: {$a^3$, $b^3$, $c^3$}

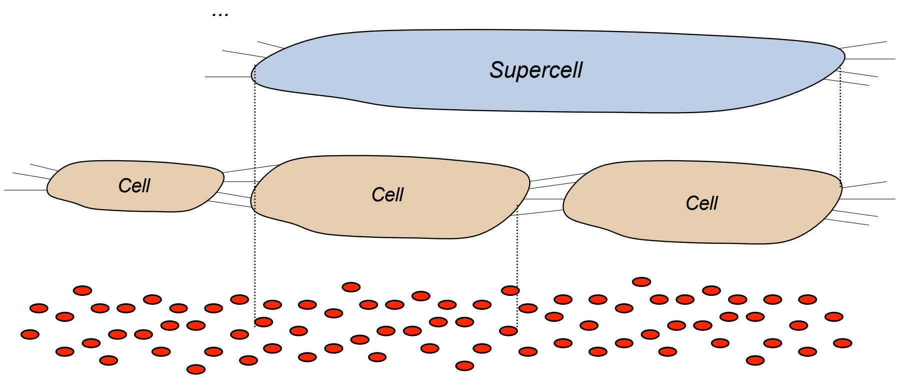




Membrane system

## Integration and Implementation

- Chemical blending is implemented by means of special symbols in the P systems rules
- Putting all together
- …future work…



# Conclusions

## “The computer revolution hasn’t happen yet!”

Alan Kay
1998

## “The important thing is that you have lots of ideas and that you learn most are going to be wrong.”

Francis Crick