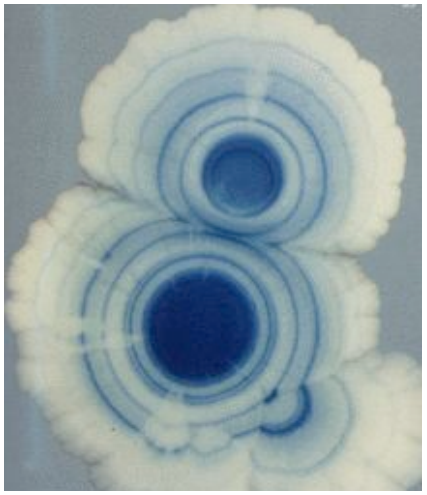
Amorphous Computing

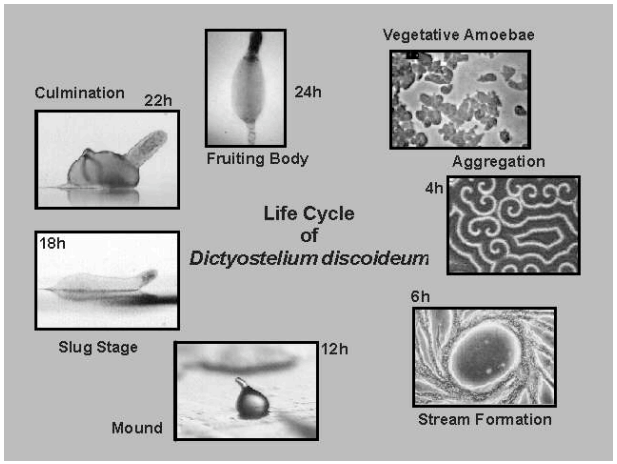
Pattern formation *in silico*

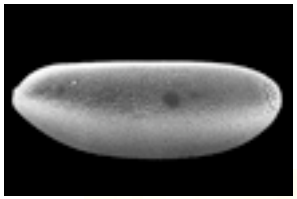
Synthetic Biology Conference
6/9/04

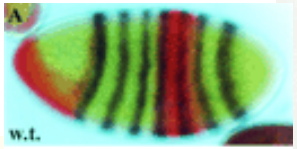
Radhika Nagpal
Harvard University

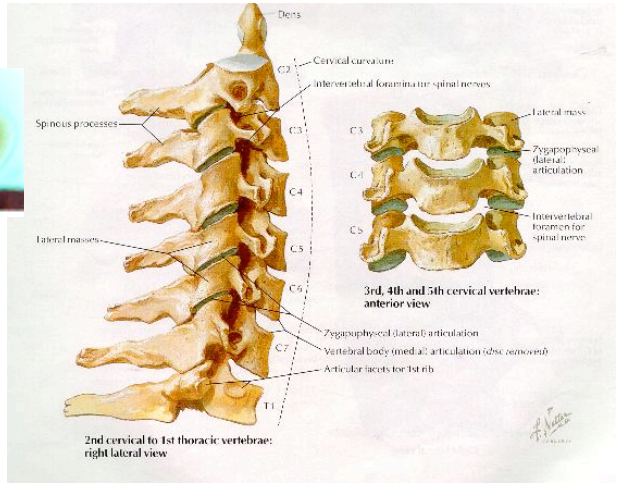
Pattern in Nature



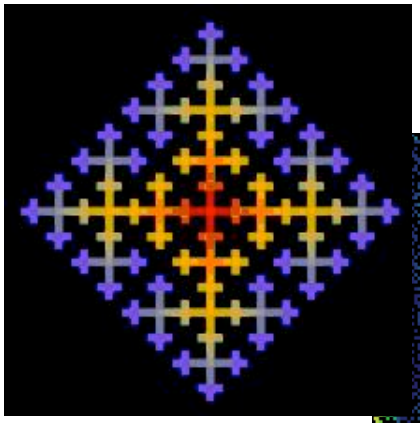


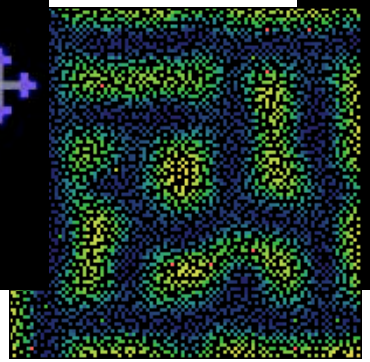


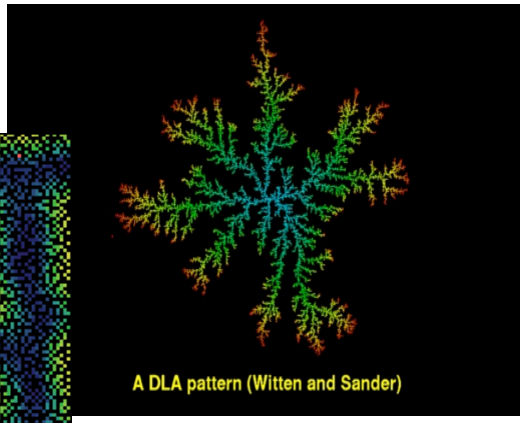


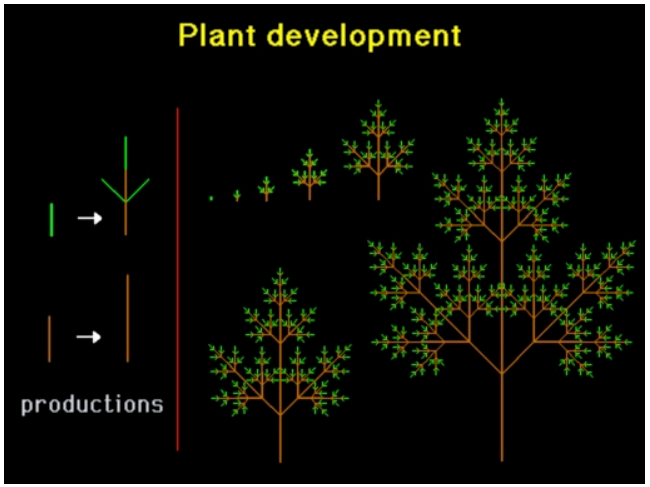


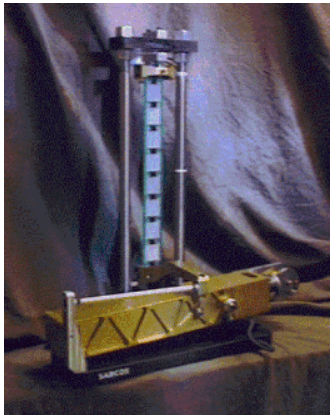
Pattern in Artificial Systems

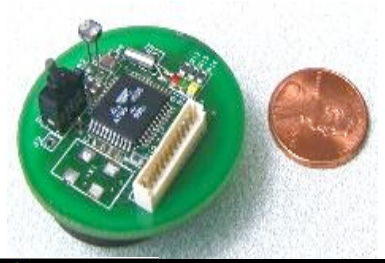


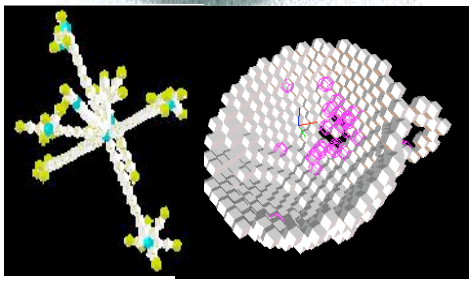


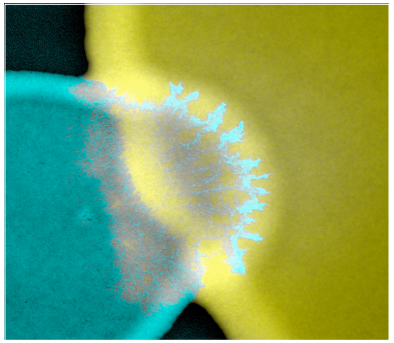










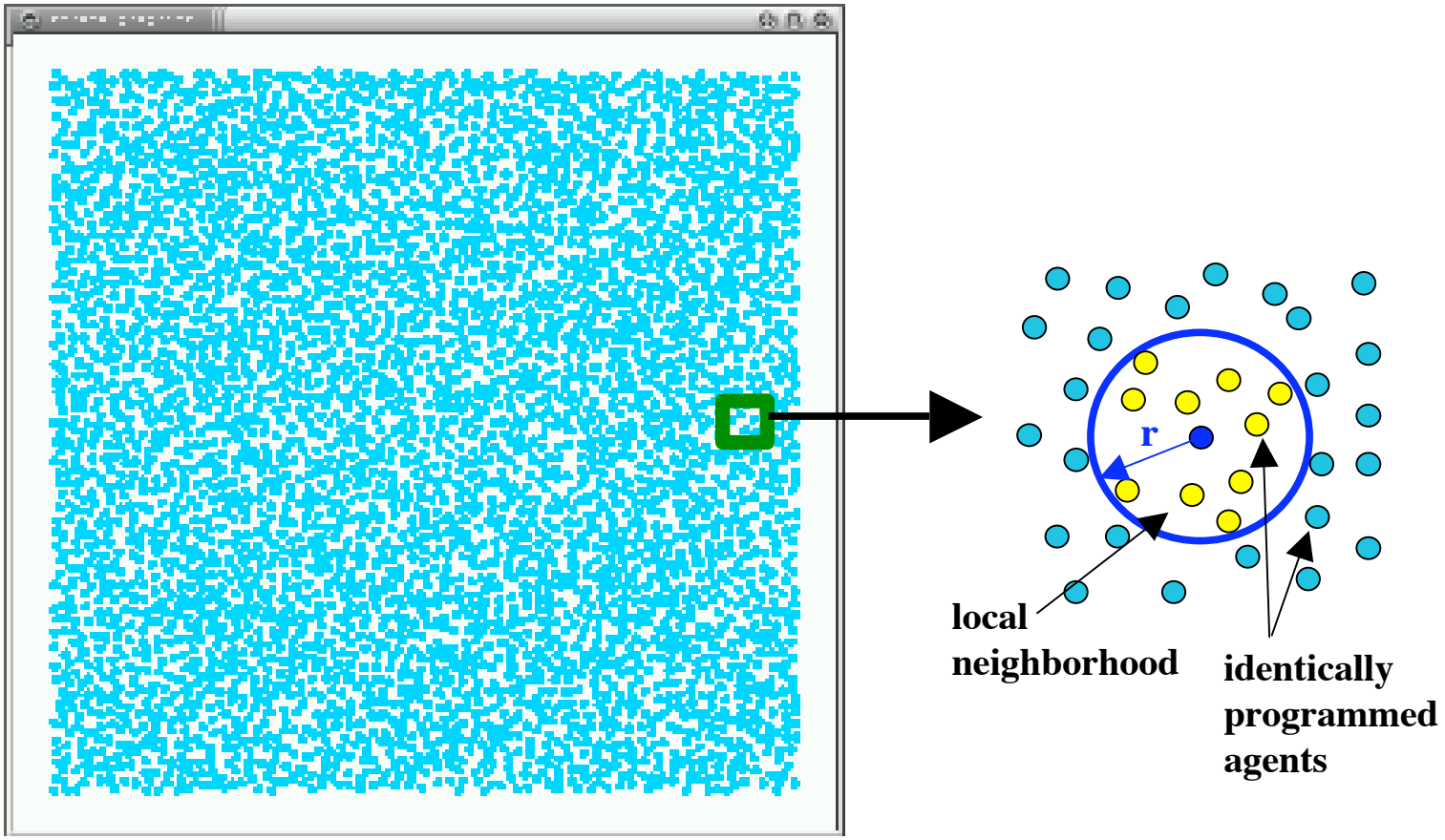


Amorphous Computing

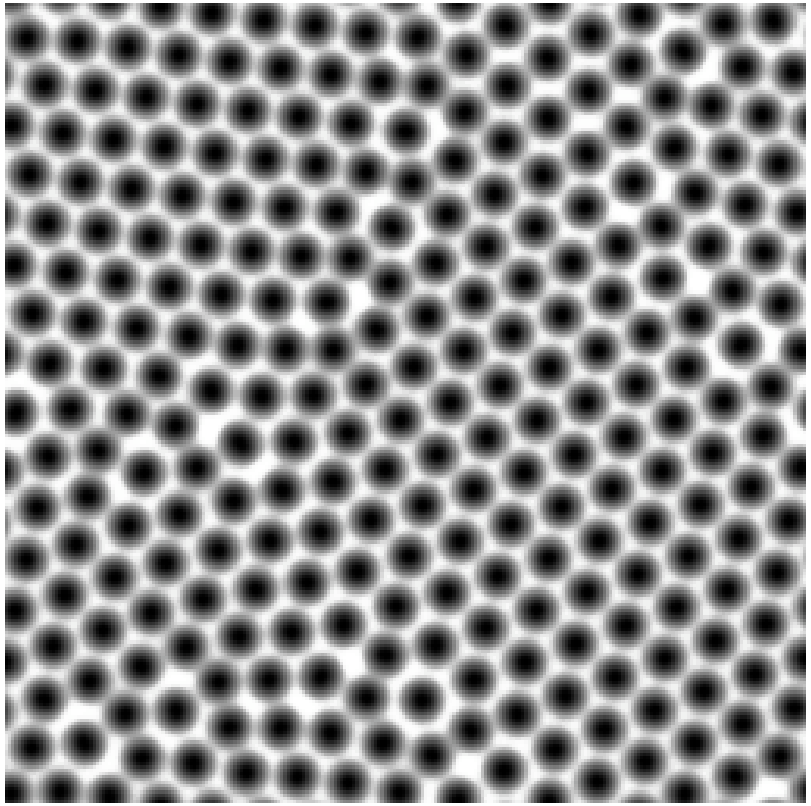
- How do we obtain a *robust behavior* from the cooperation of vast numbers of unreliable parts?
- How do we engineer *pre-specified global* behavior from local interactions?

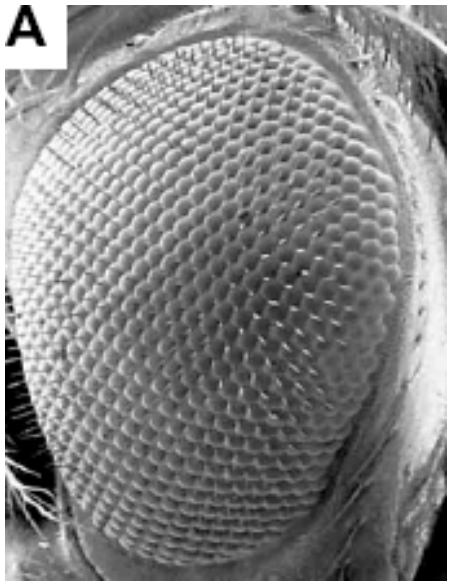
- Organizational principles
- Algorithms
  - Design and analysis
- Programming models and languages
  - Catalogues of primitives
  - Means of combination
  - Means of abstraction
  - Development tools
  - Compilation technology targeted to appropriate substrates

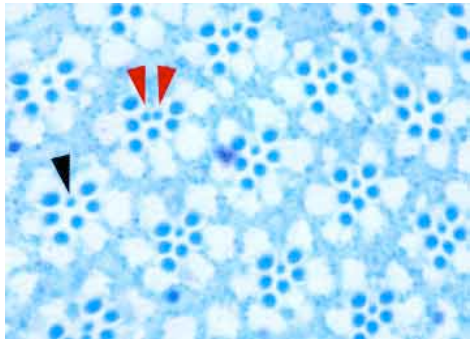
An Amorphous Computer



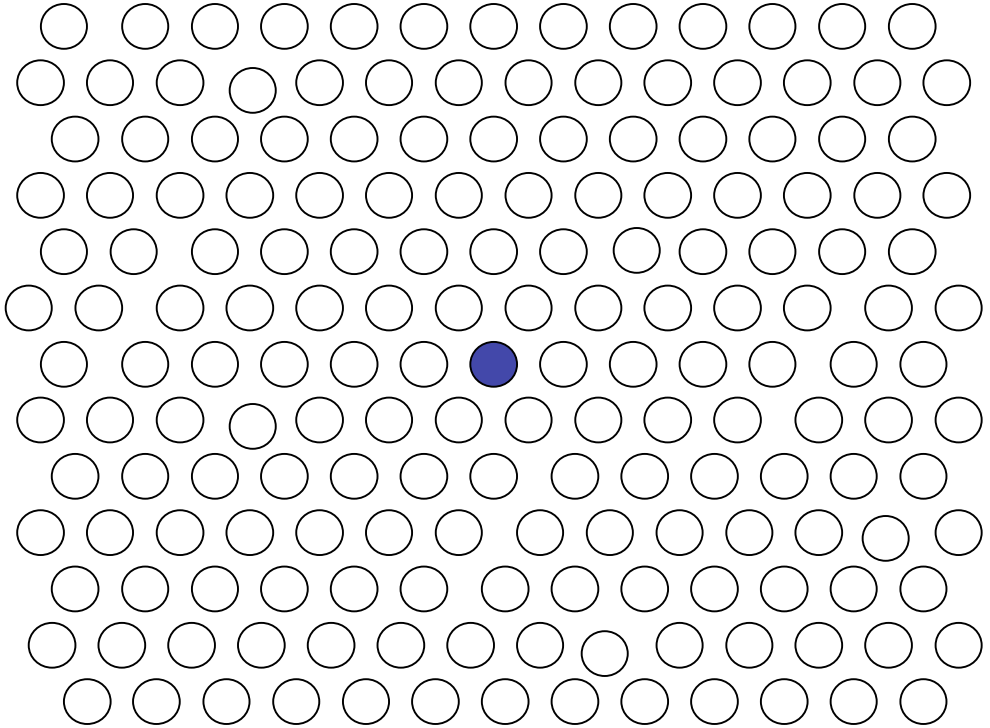
A Simple Pattern: Polka Dots



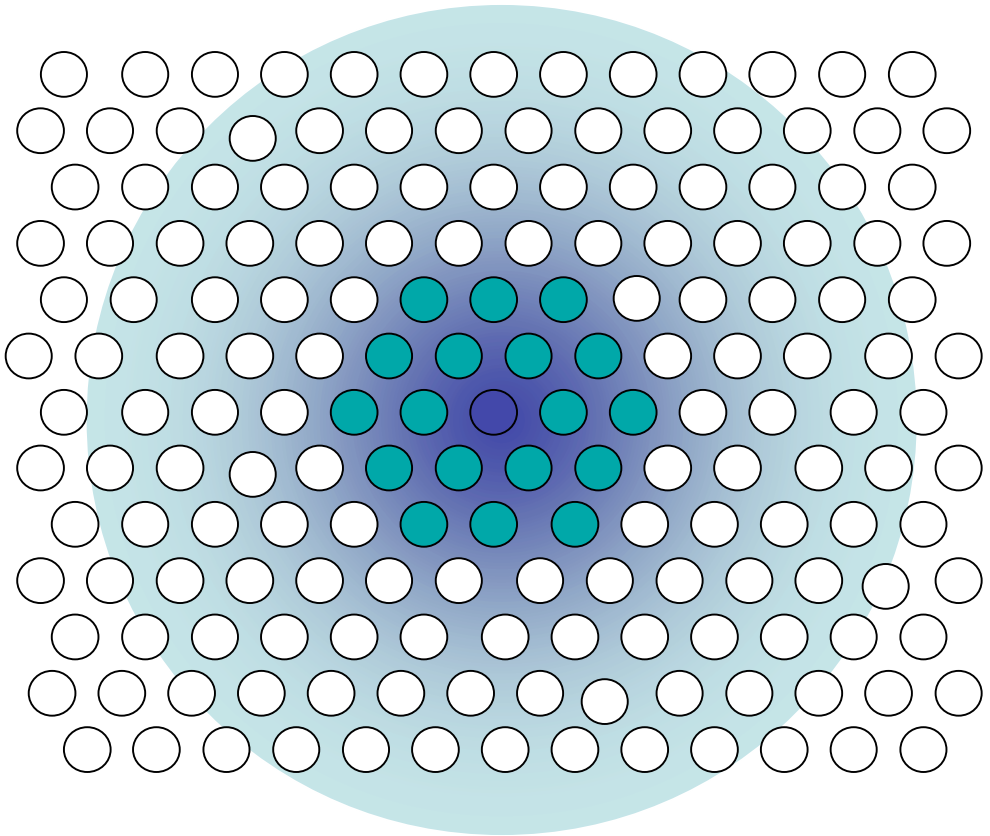




A Simple Pattern

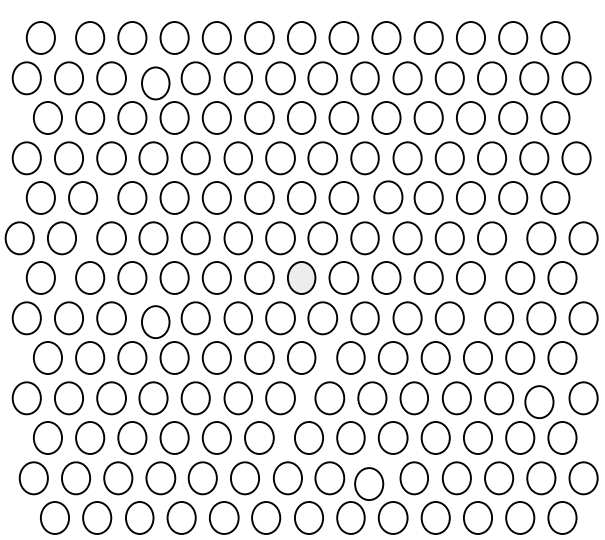


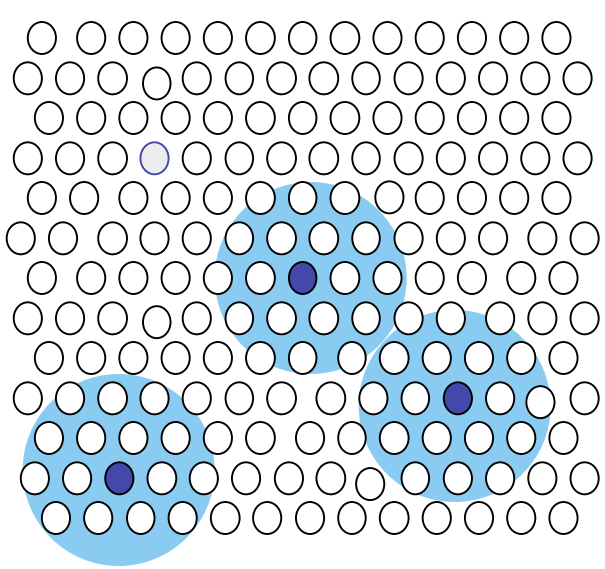
A Simple Pattern

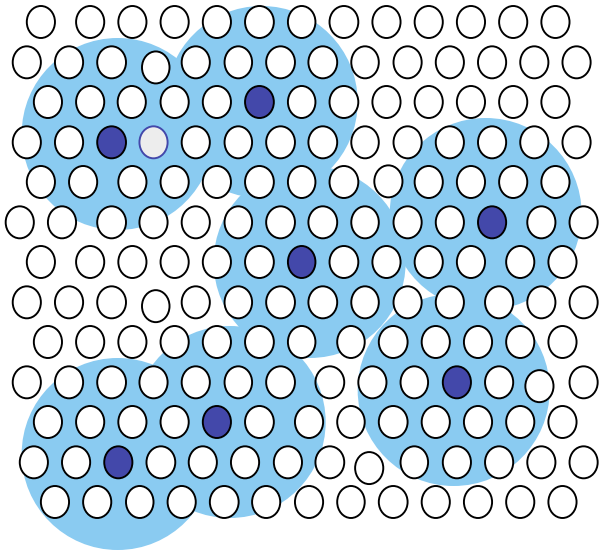


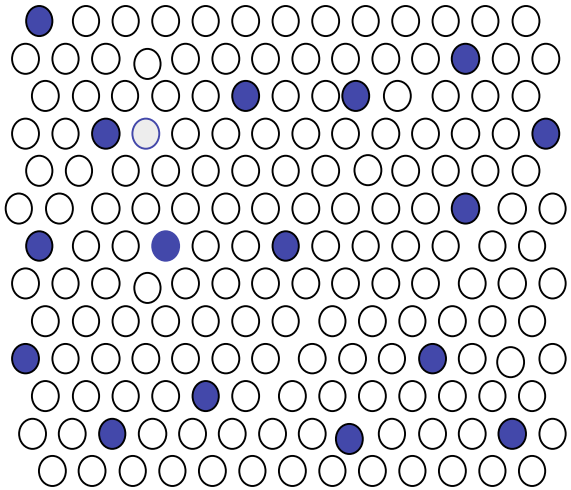
Gradient

Polka Dot Program

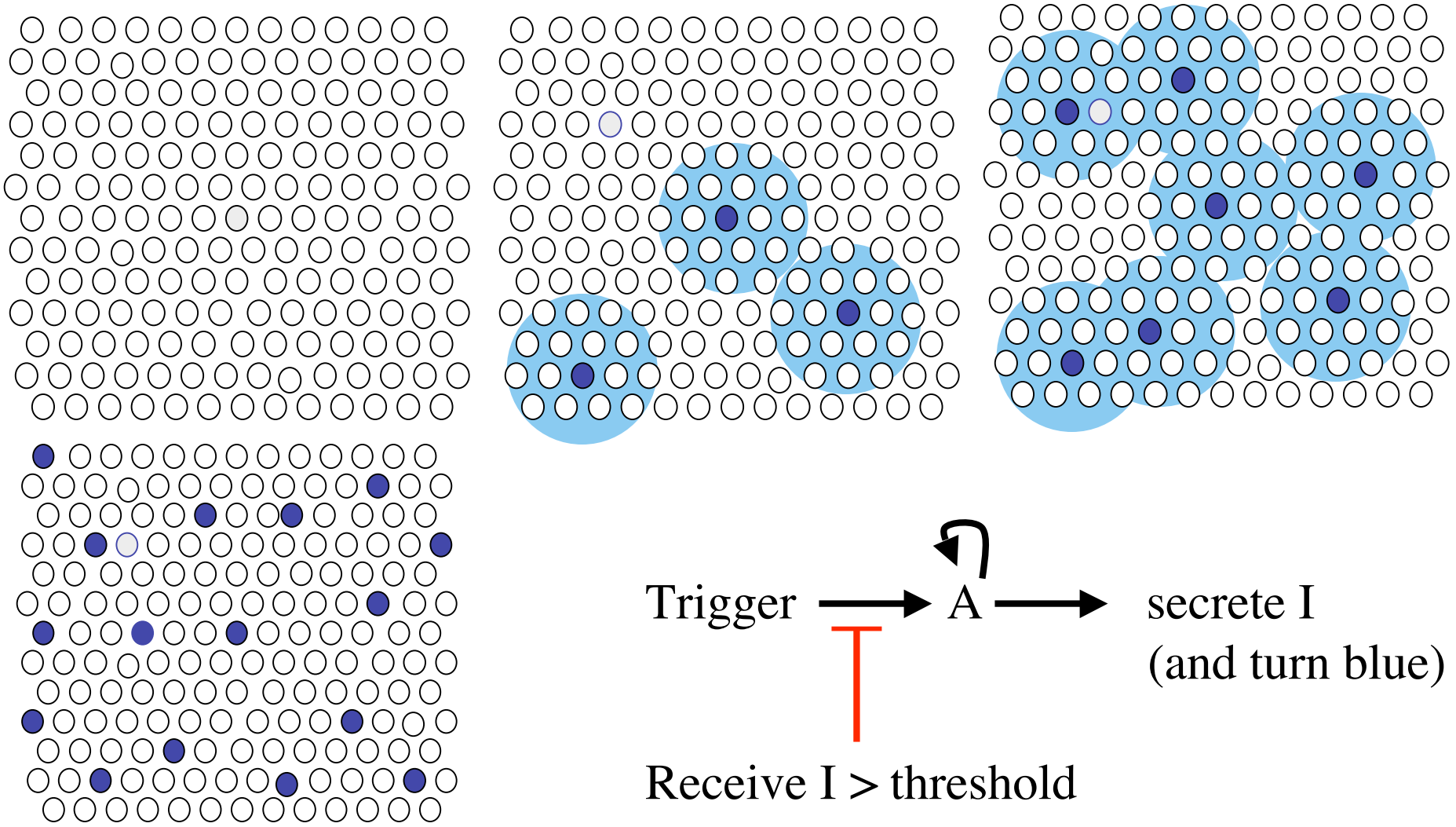




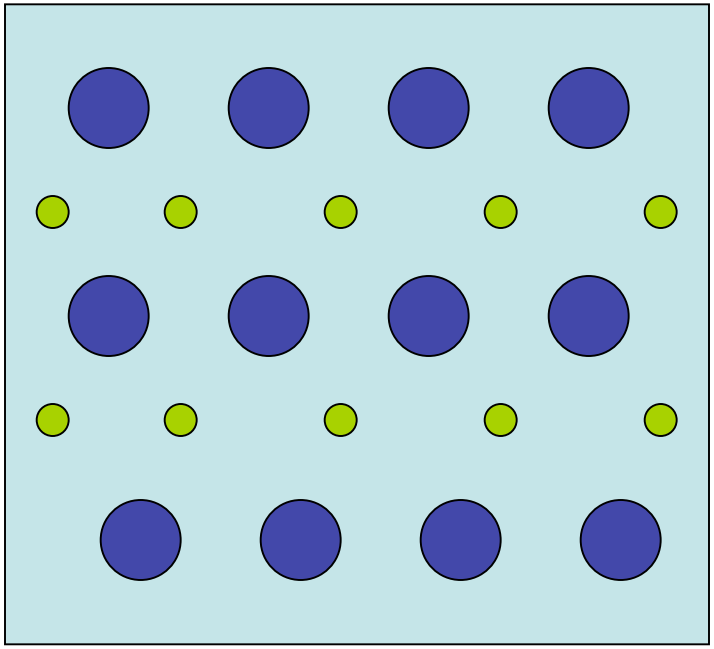


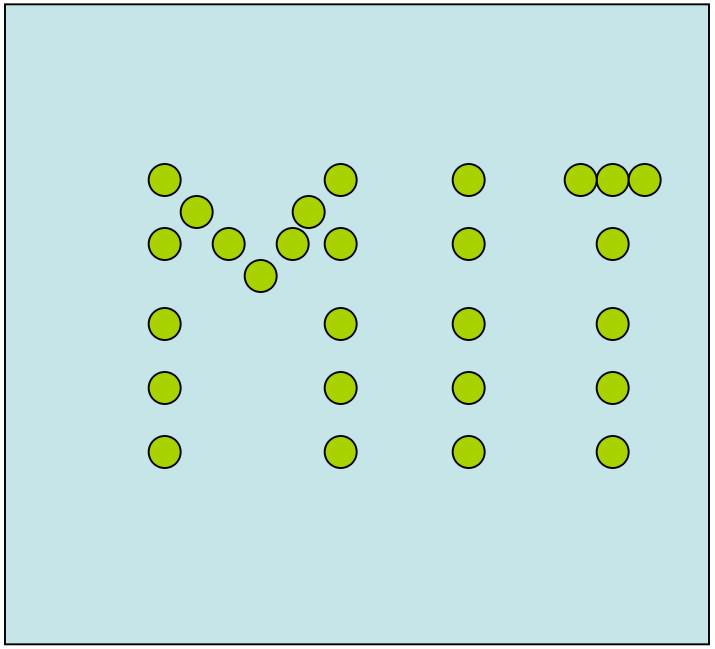


Polka Dot Program

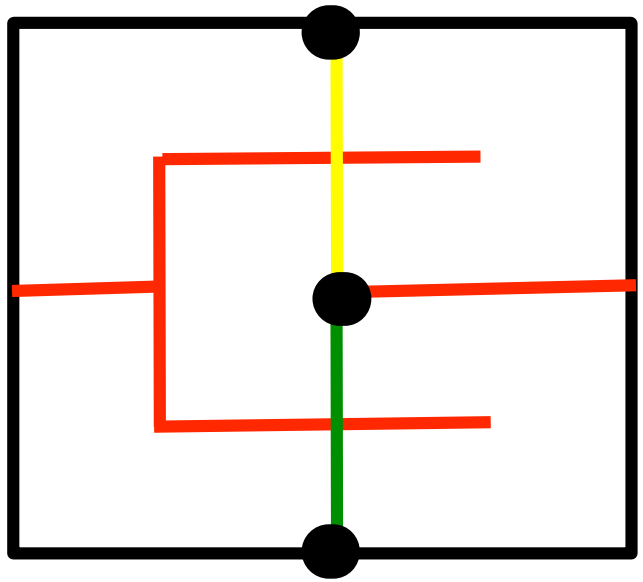


How much more complex are these?

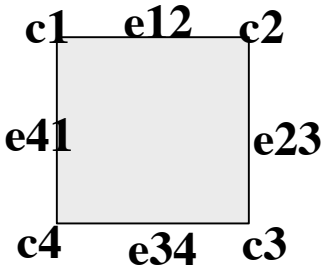


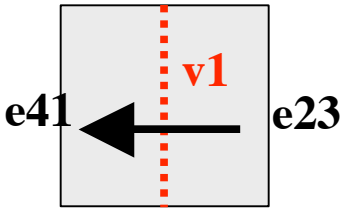


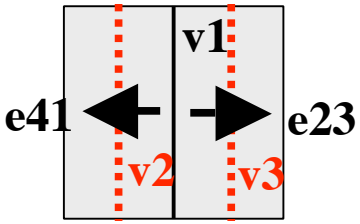
Suppose we want to create an arbitrary pattern?

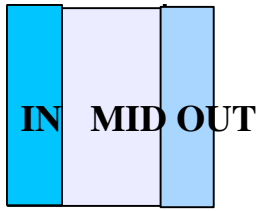


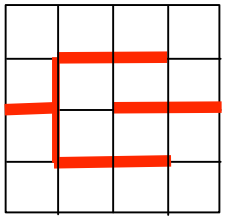
An Inverter Program

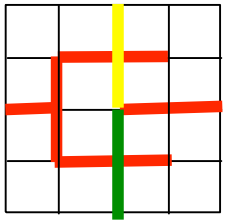


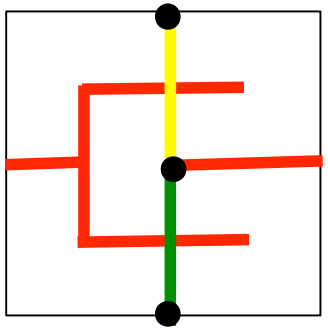






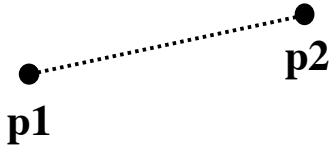




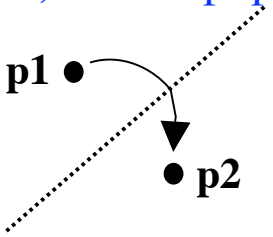


A Set of Construction Rules

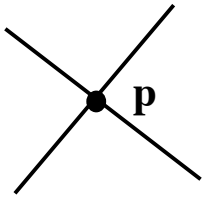
(A1) crease-lbp



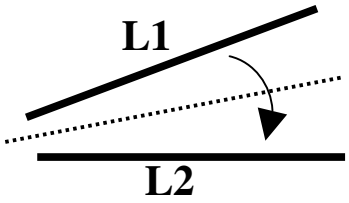
(A2) crease-p2p



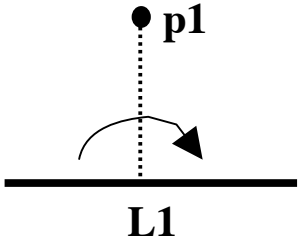
intersect



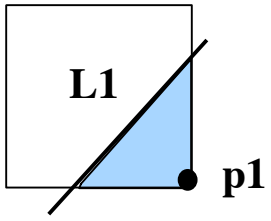
(A3) crease-l2l



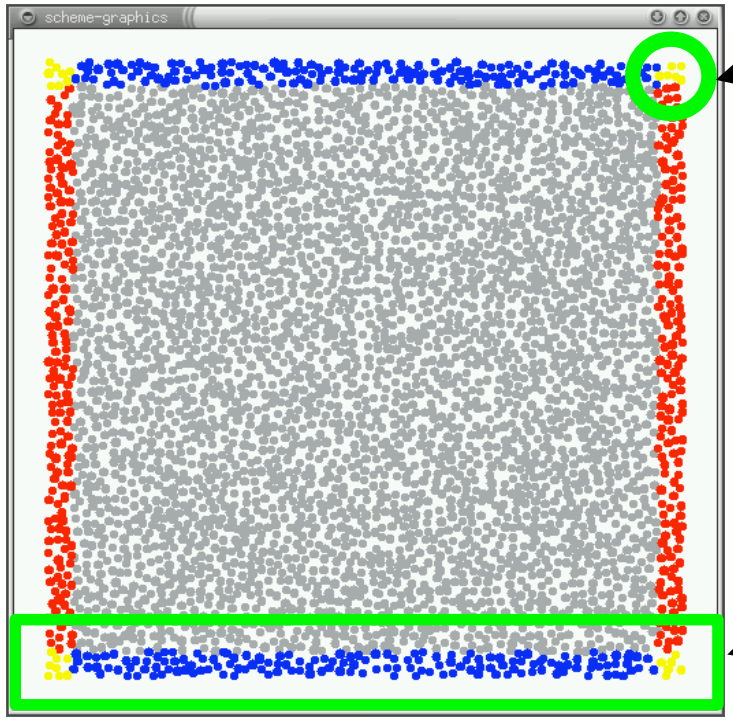
(A4) crease-l2self



region



Initial Conditions: "Determinants"



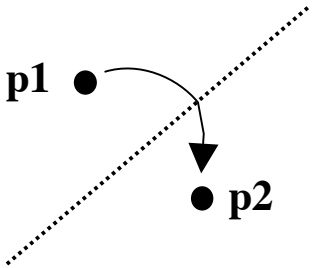
Point of agents

Line of agents

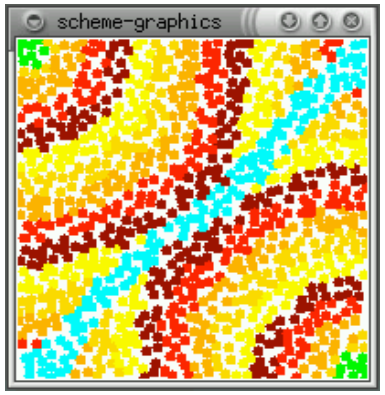
Agent Local State
c2 = #t
c3 = #f
.....

Implementing the Rules

"Balancing Gradients"

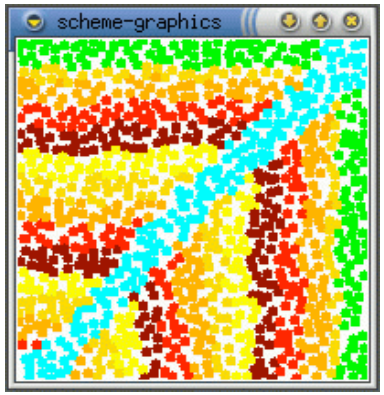


p1



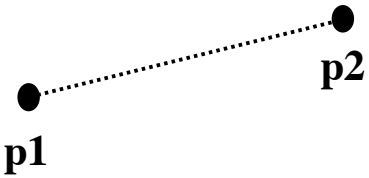
p2

L1

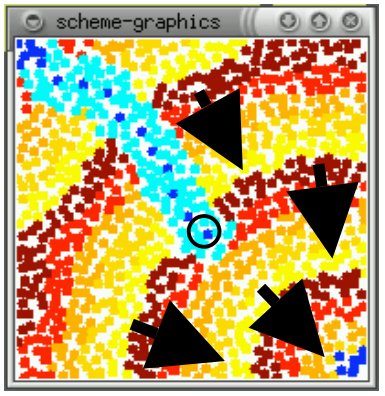


L2

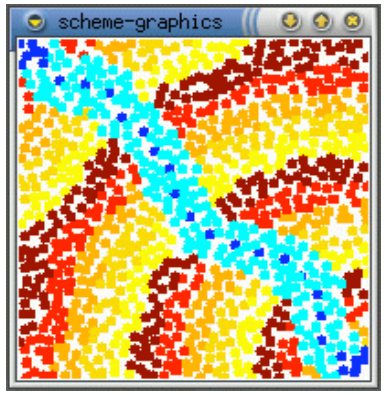
"Chemotropism"



p2



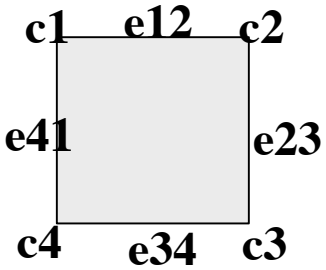
p1

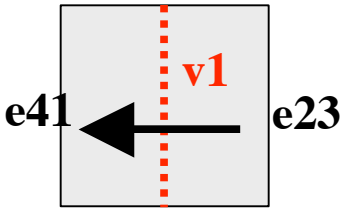


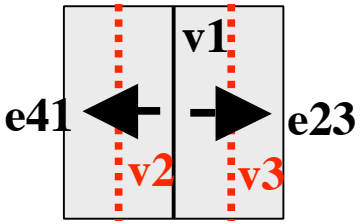
Individual Element

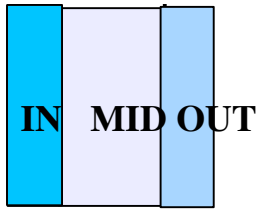
```
(define (axiom2-rule i1 i2 g1 g2 gend)
   (if i1 (create-gradient g1))
   (if i2 (begin (wait-for-gradient g1)
                 (create-gradient g2)))
   (if i1 (begin (wait-for-gradient g1)
                 (wait local-delay)
                 (create-gradient gend)))

   (wait-for-gradients gend)

   (if (<= (abs (- g1 g2)) threshold)
       #t
       #f)
 )
```

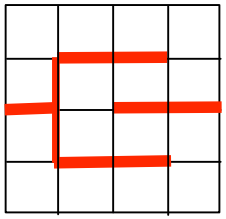
An Inverter Program

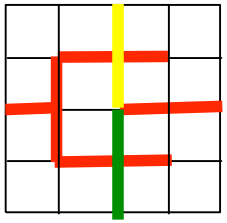


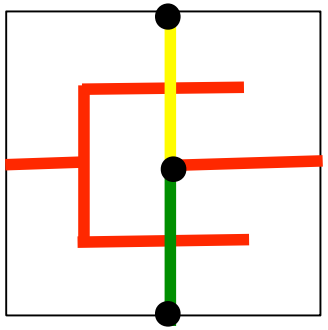




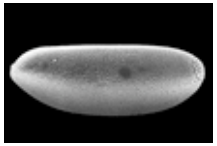


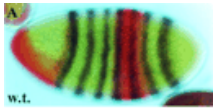


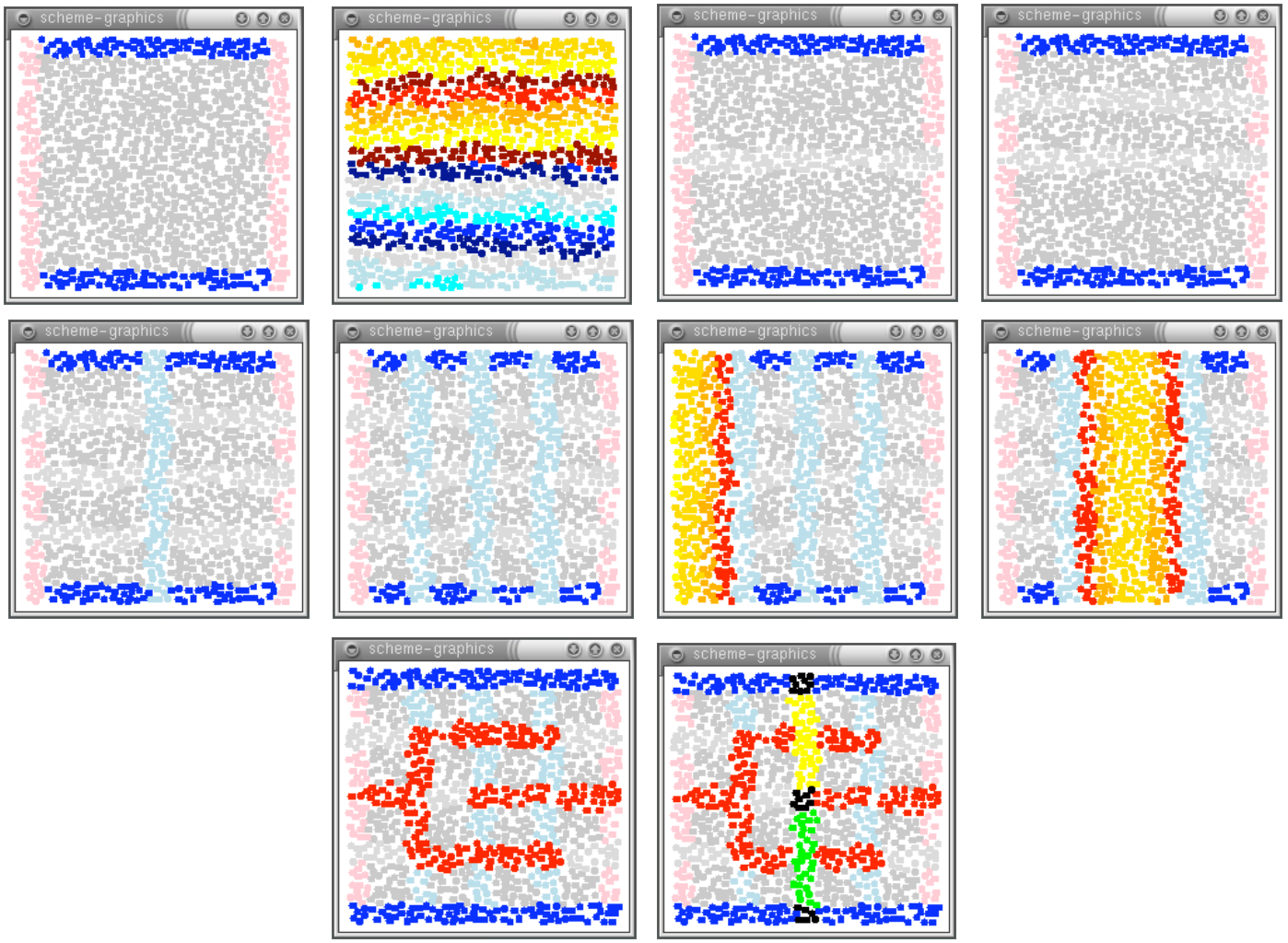


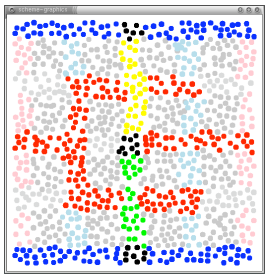


Pattern-Formation on an Amorphous Computer



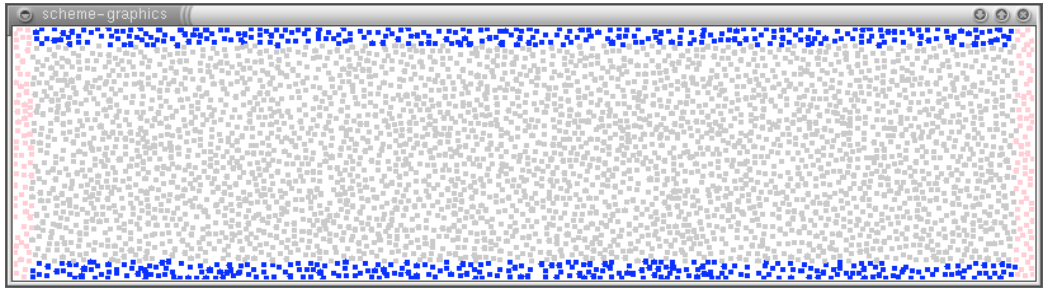




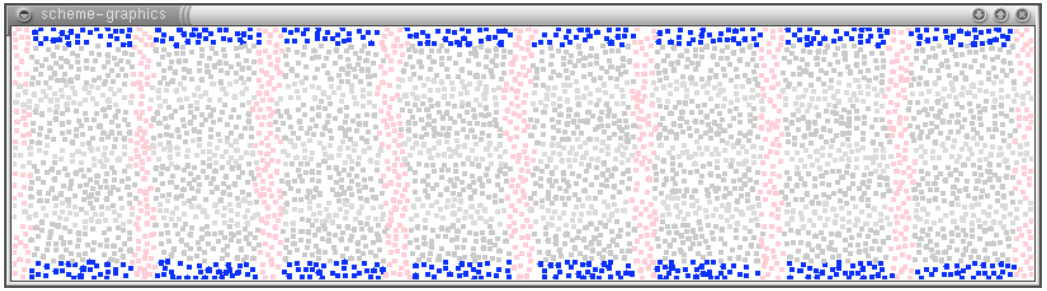


Inverter Chain Pattern

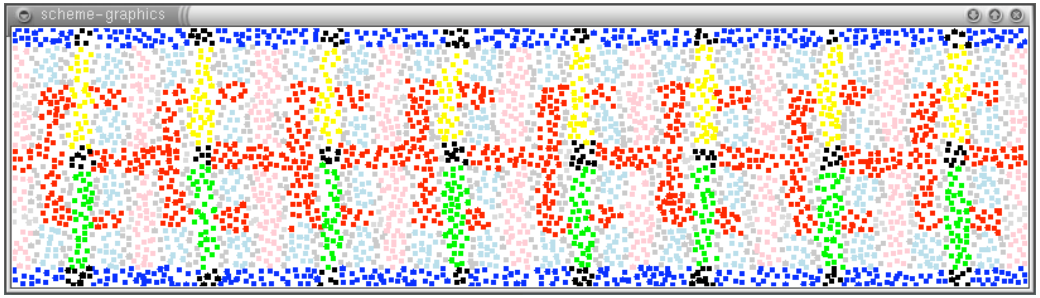
;; Segment into
;; **8 COMPARTMENTS**



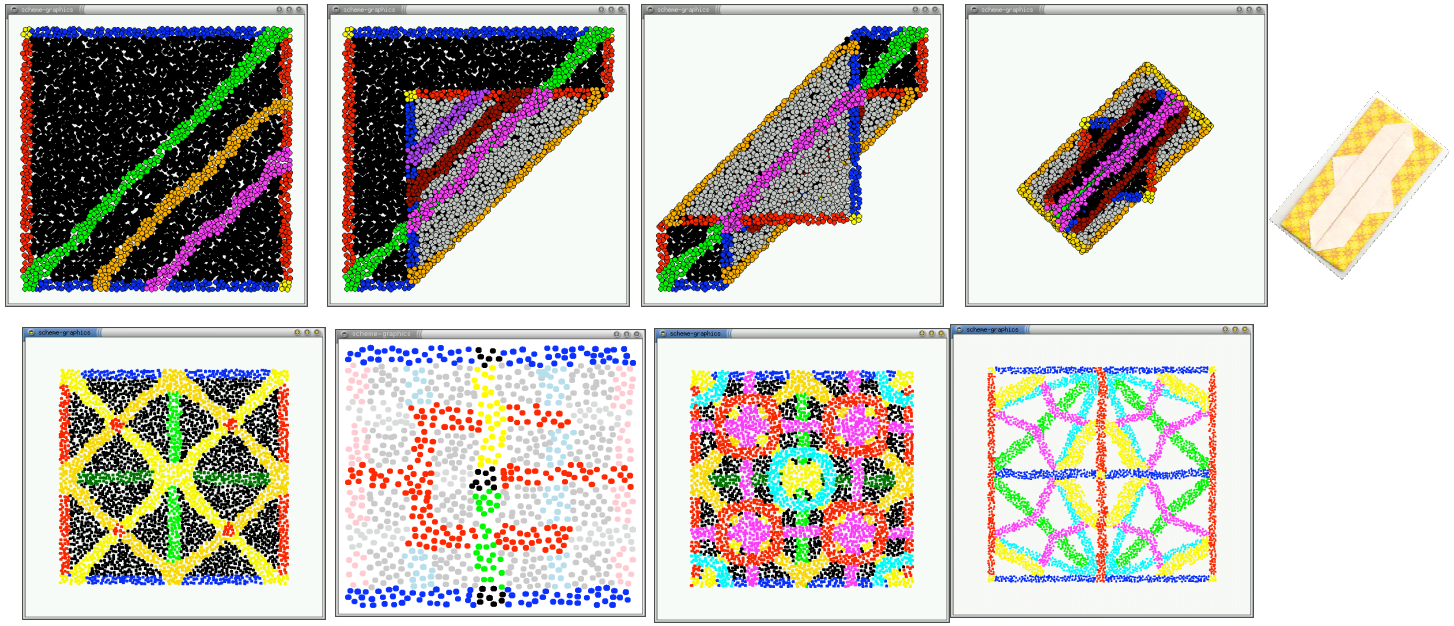
;; Execute
;; inverter pattern program
;; within a region




```
(within-region r1
    (create-inverter
        left-border1
        right-border1))
```



Many Shapes and Patterns

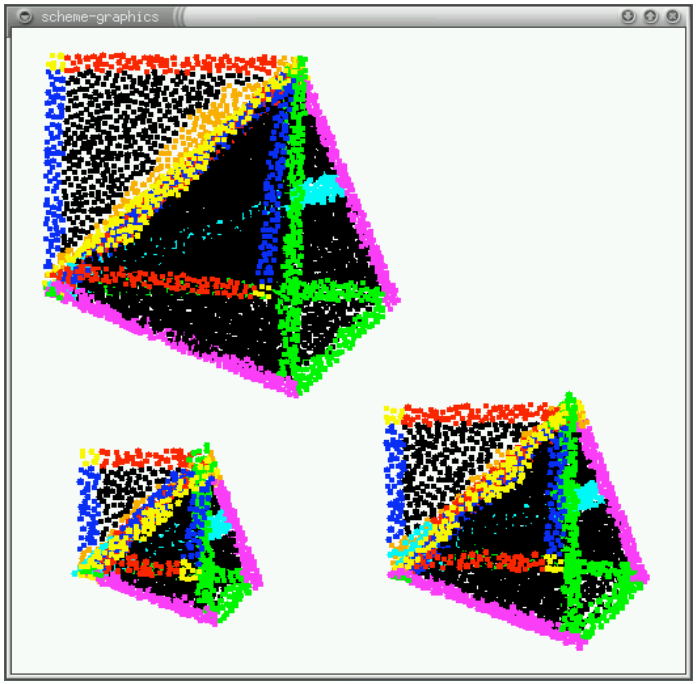


Local State: Boolean per distinct point, line, region

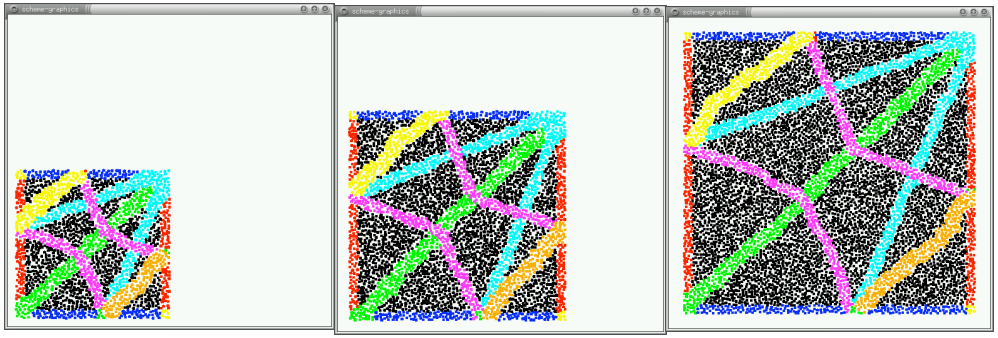
Morphogen Gradients: Many, but short-lived

Could use as few as 6

Scale-Independence

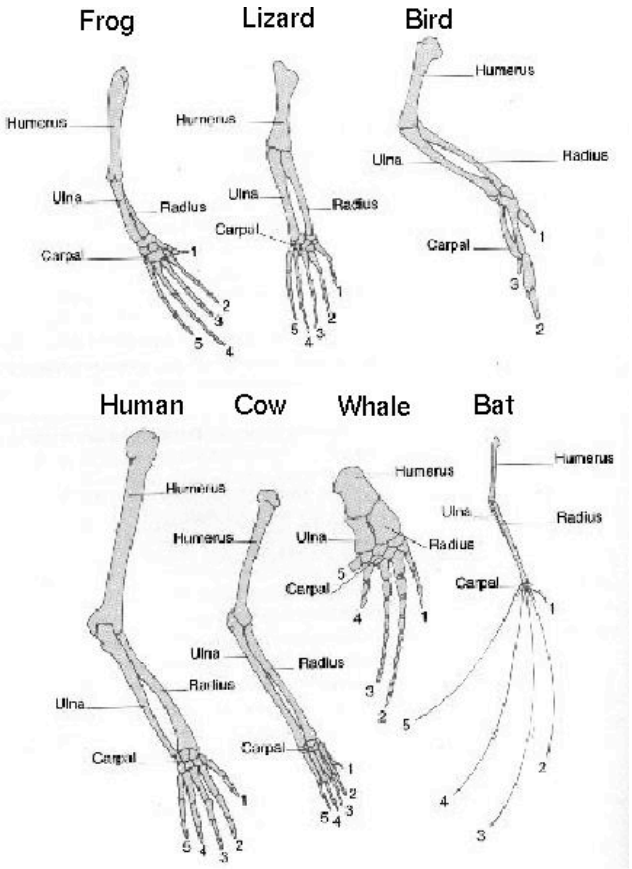


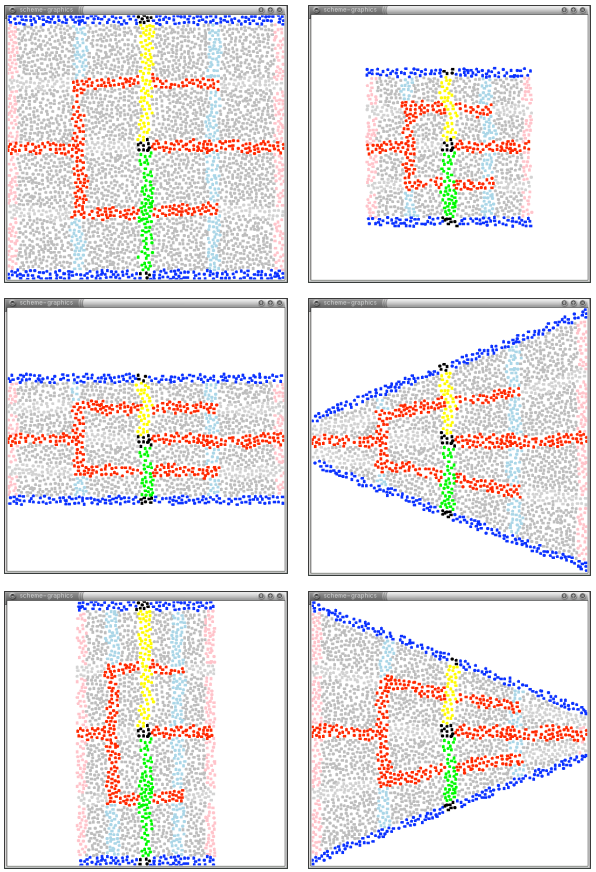
pattern *scales* with number of cells



2000 4000 8000

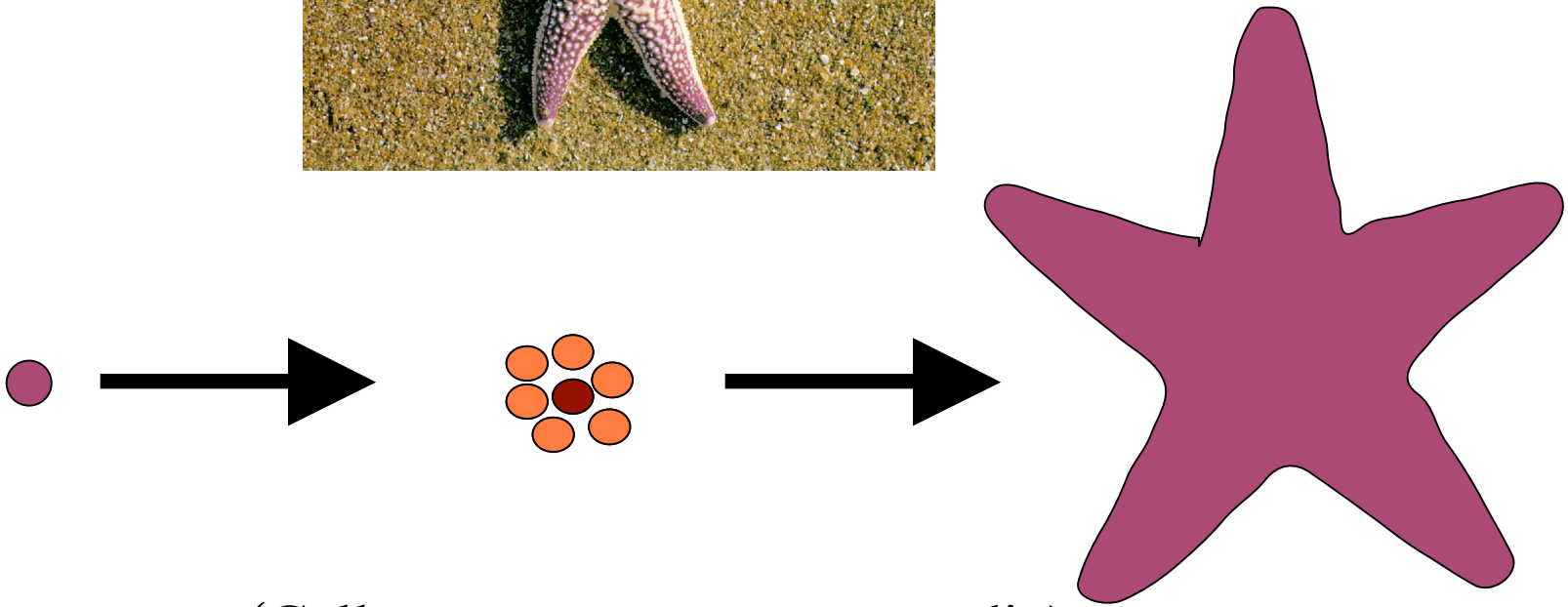
Related Structures





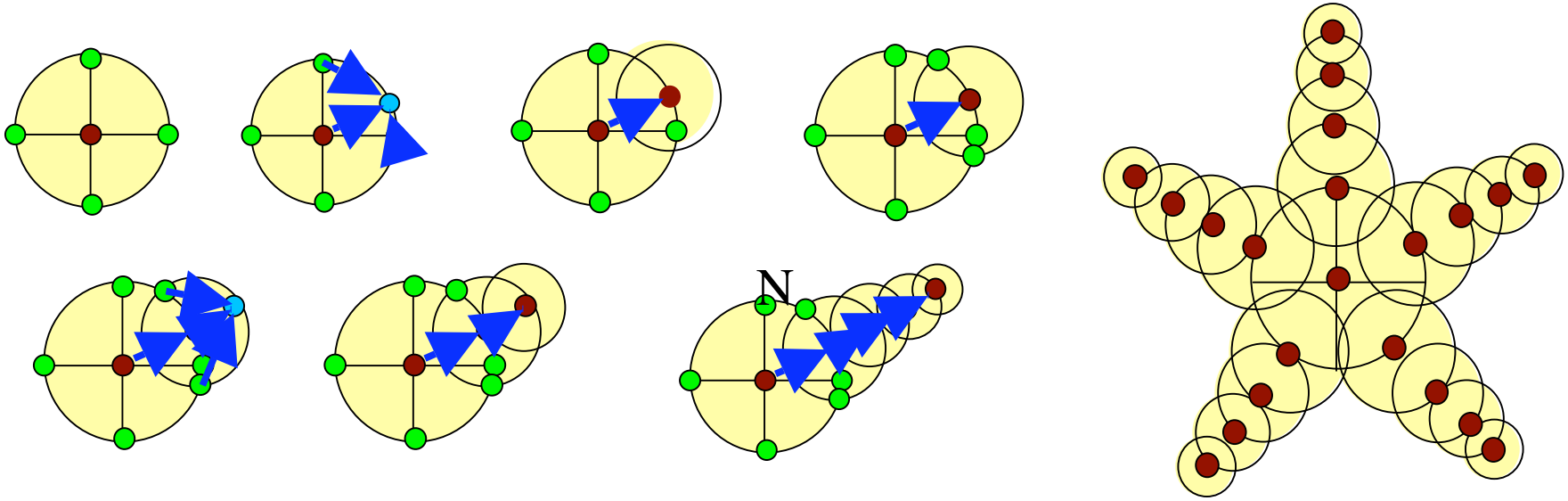
Ridley (1997) *Evolution*

Program the Cell,
to "grow" the appropriate shape



(Cell can grow, not grow, or die)

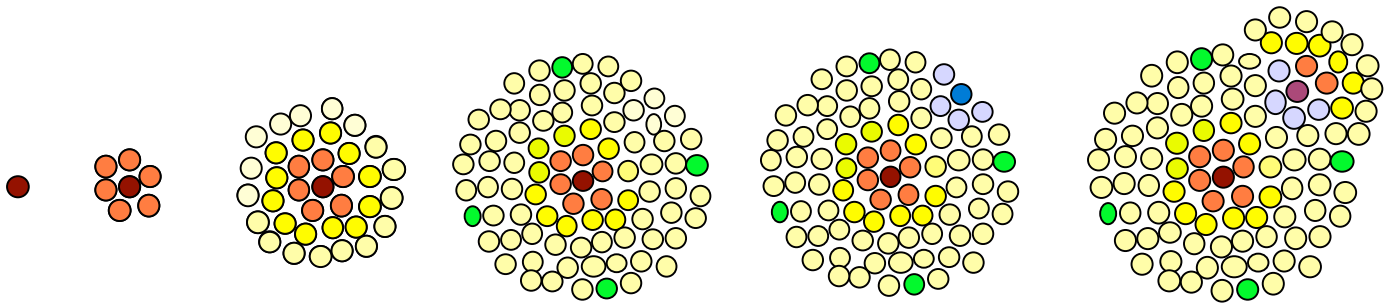
Constructing a 2D shape



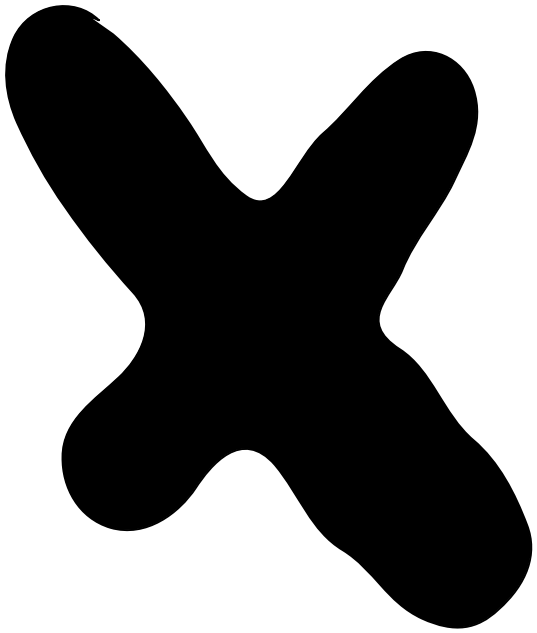
Two rules:

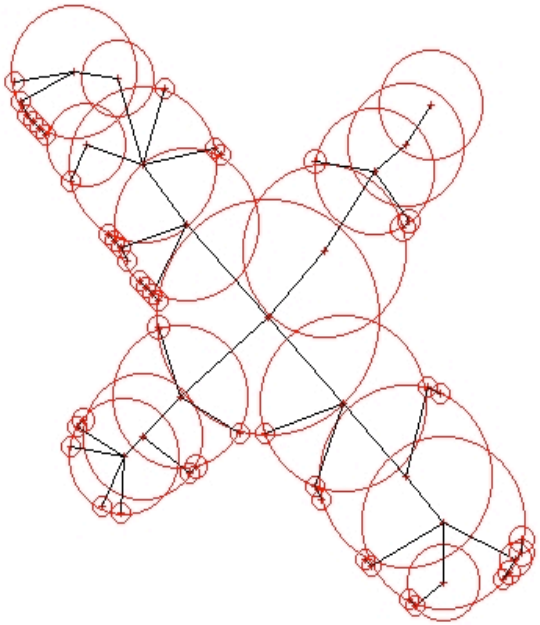
(1) grow a circle of a given radius (2) elect reference points

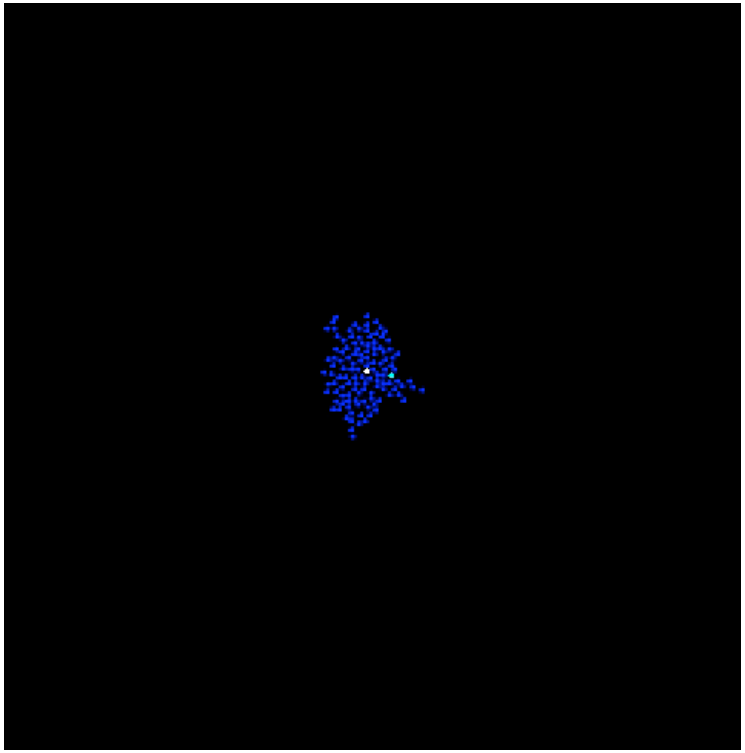
Shape == Set of rules (proportional to the number of circles)



Self-Assembly using directed growth



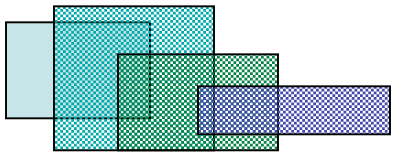




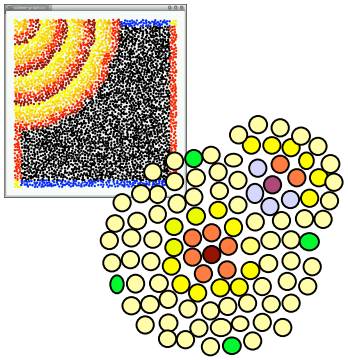
Kondacs, Chang

Self-Reconfiguration using directed growth

Goal Shape:
described using
block-construction representation



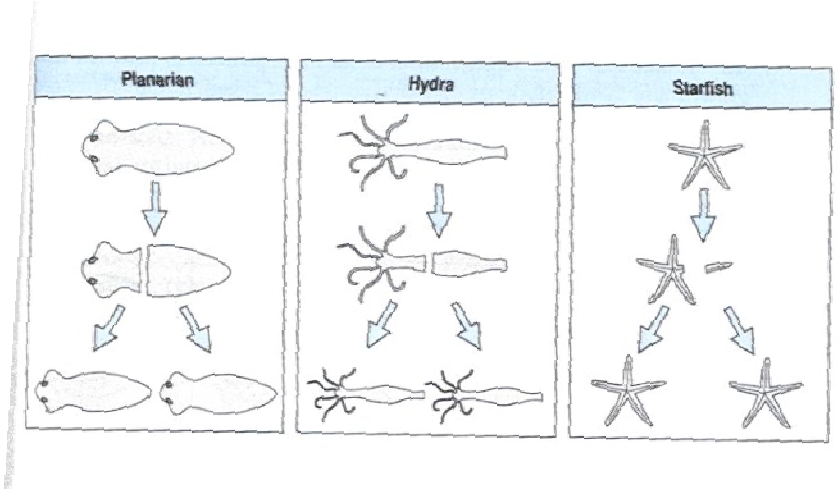
COMPILED

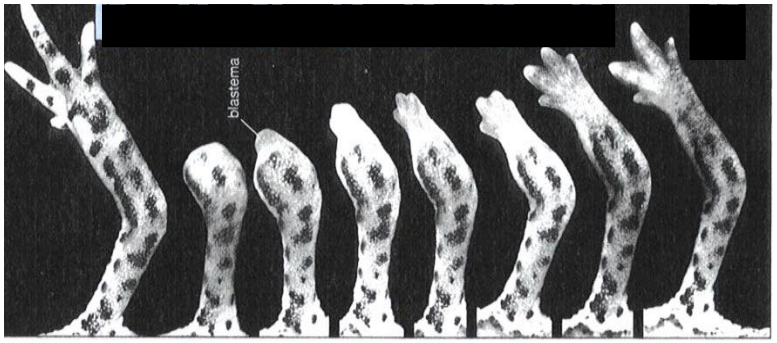


Program run by
identically-programmed mobile modules,

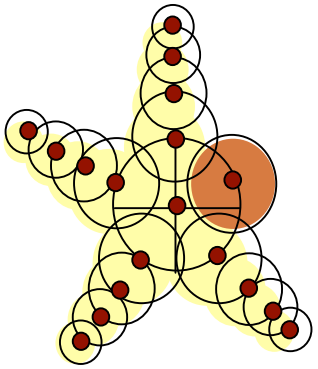
Kasper Stoy,
Univ of Southern Denmark

Self-Repair and Regeneration



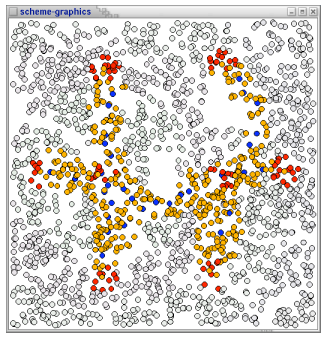


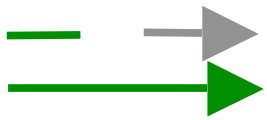
Regenerating structures



Absence of neighbor causes circle to recreate its neighbor, which in turn recreates its neighbor - thus regenerating the broken structure

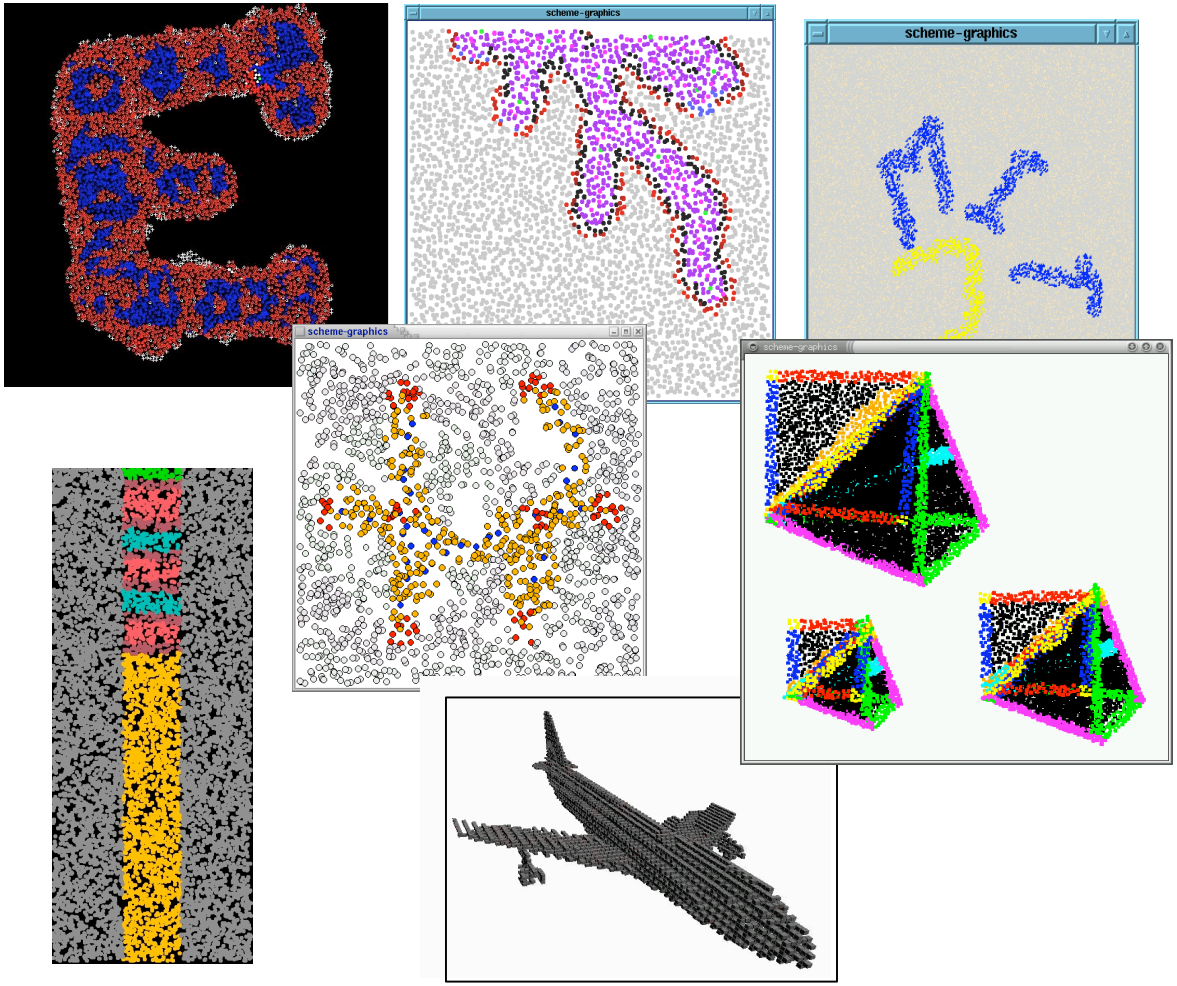
Self-repairing patterns





If a line is broken, one part dies off and the other regrows

Languages for Pattern Formation



Languages for "constructing" pattern

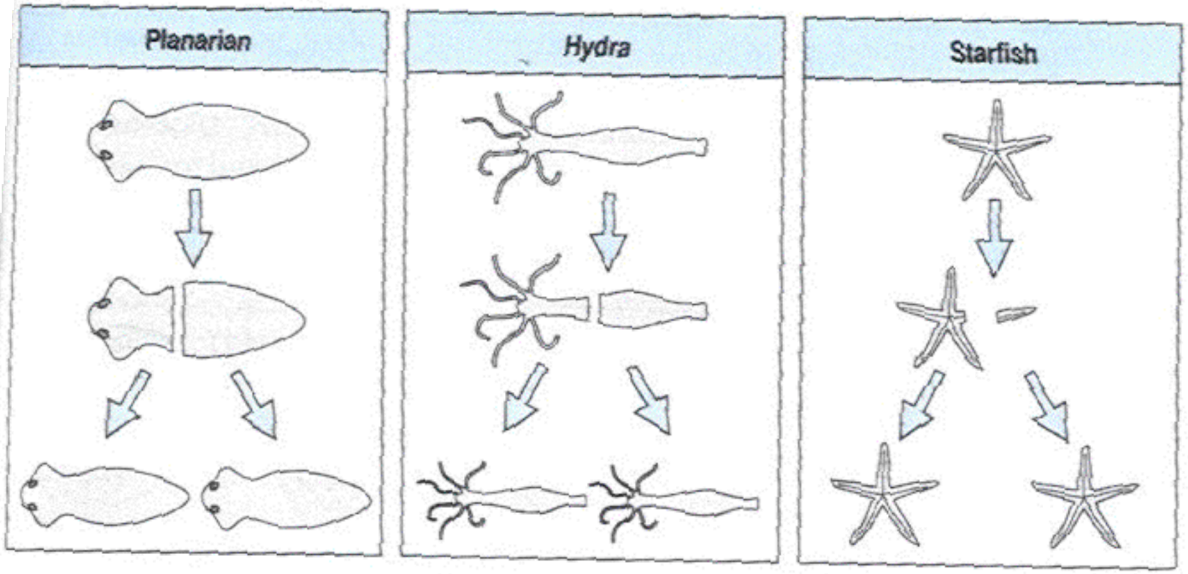
COMPILER

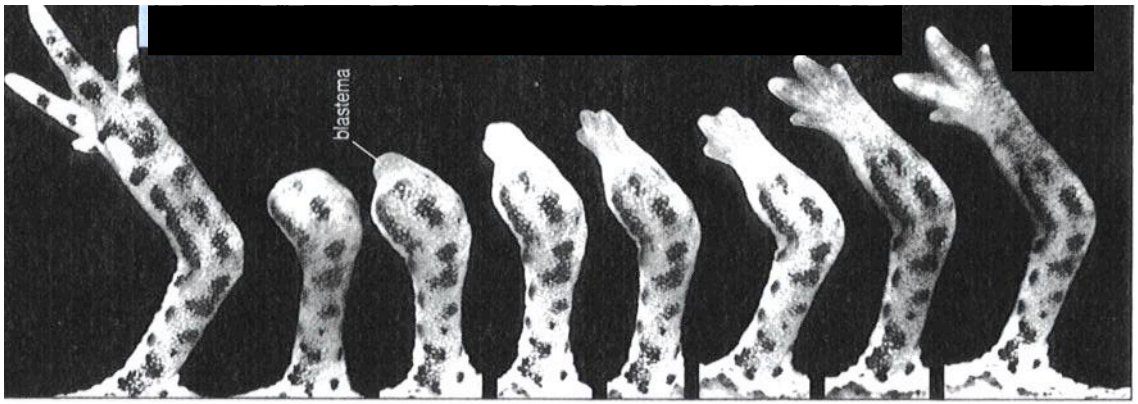
Behavior of an individual element

Amorphous Computing Project (MIT AI Lab)

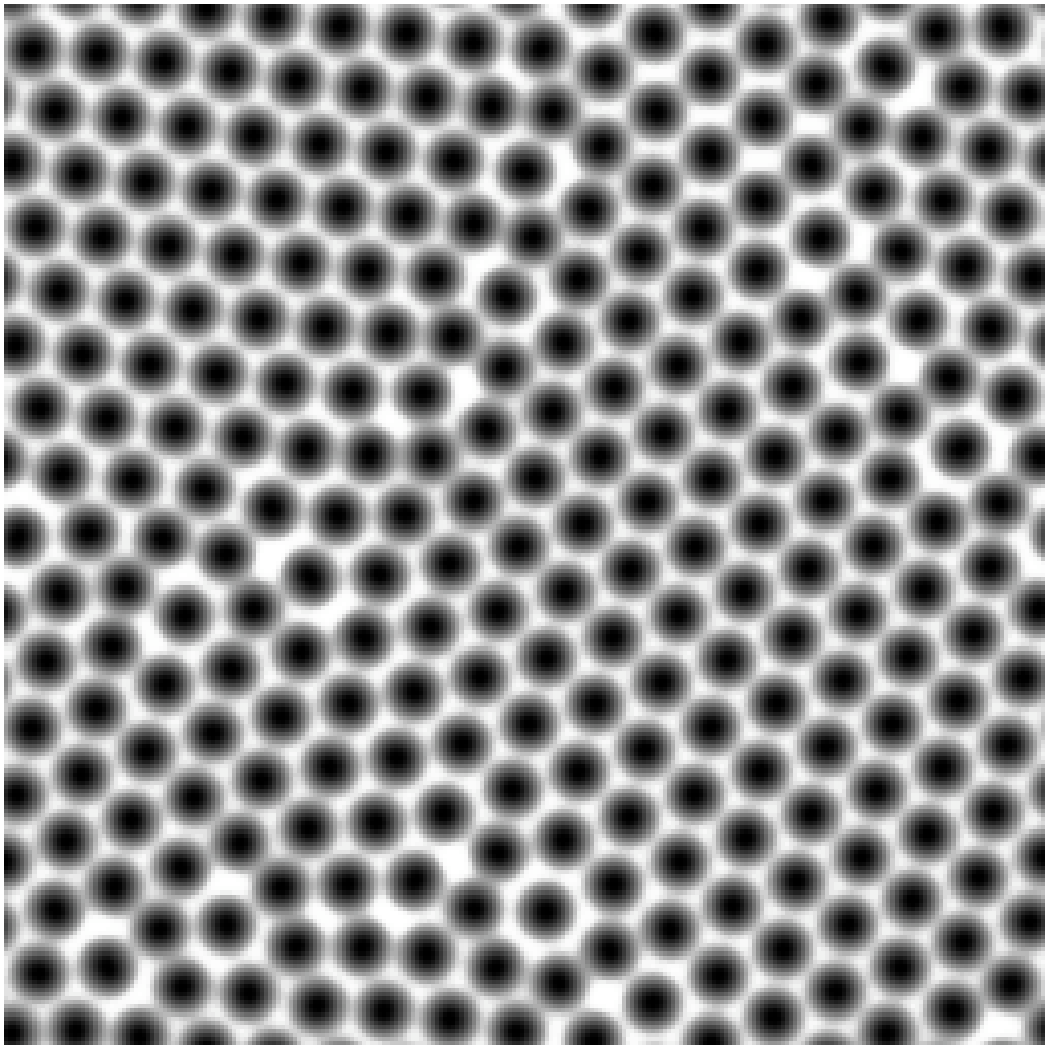
Gerry Sussman, Hal Abelson, Tom Knight, Daniel Coore, Ron Weiss, Jake Beal, Attila Kondacs, Catherine Chang, Lauren Clement

Suppose we want to make things that regenerate?





Back to Polka Dots

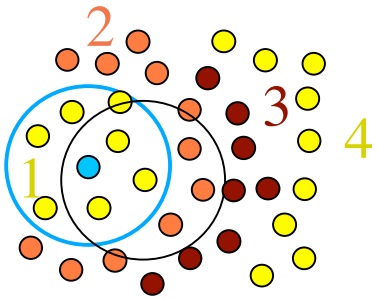


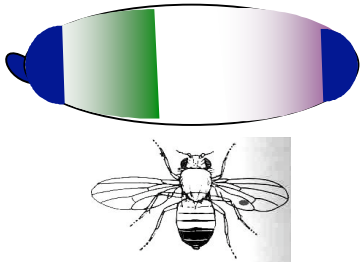
Summary

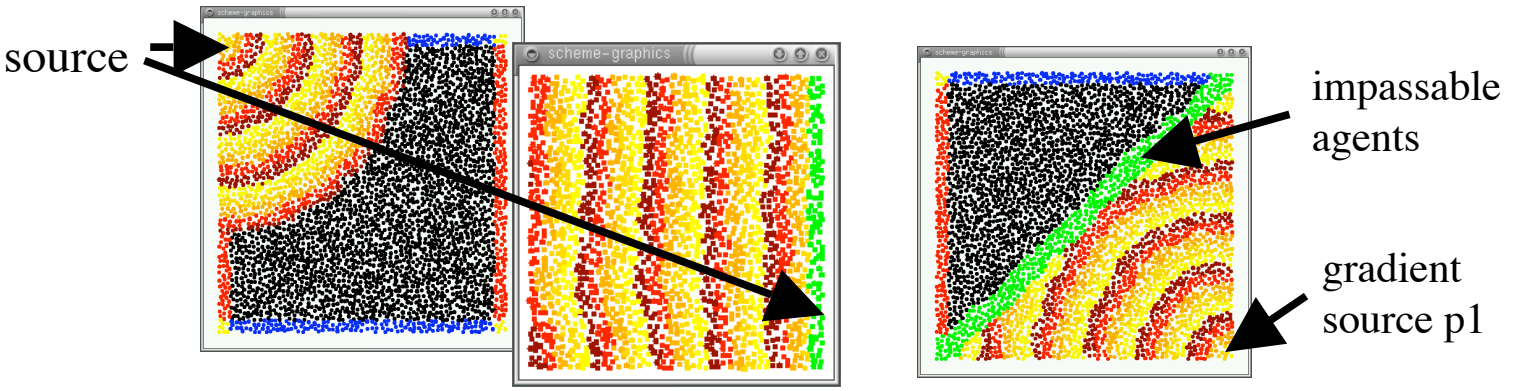
- Cellular Automata Model
- Two ways of making Polka Dots
  - Reaction Diffusion
  - Lateral Inhibition
  - What about more complex dots?
- More Complex Patterns
  - French Flag
  - Inverter
  - How much more complex is a chain of Inverters?
- Patterns from Growth
  - Starfish Program

Local Interactions: Biologically-Inspired Primitives

1. Gradients



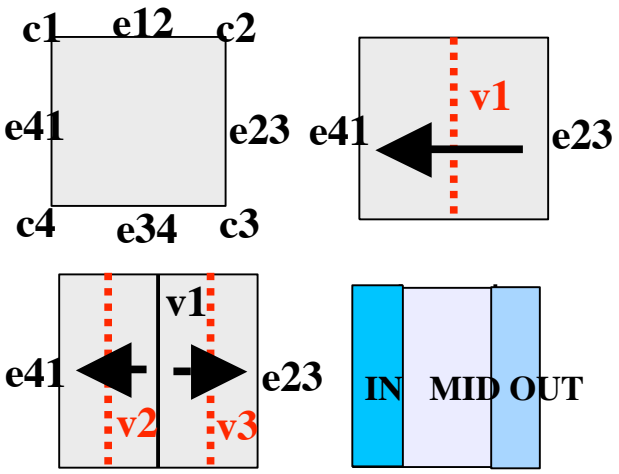




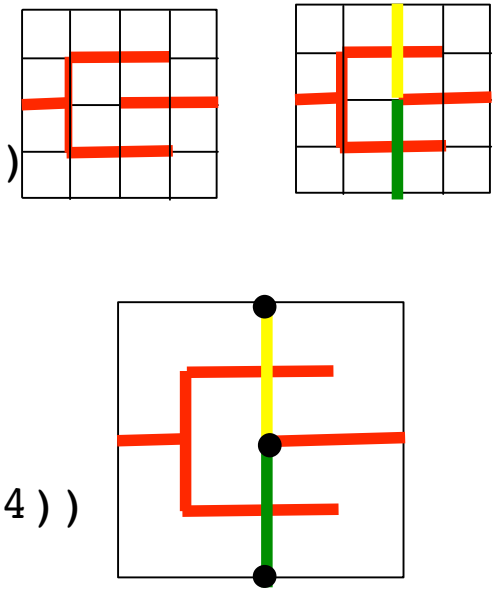
A "Generative" Global Program

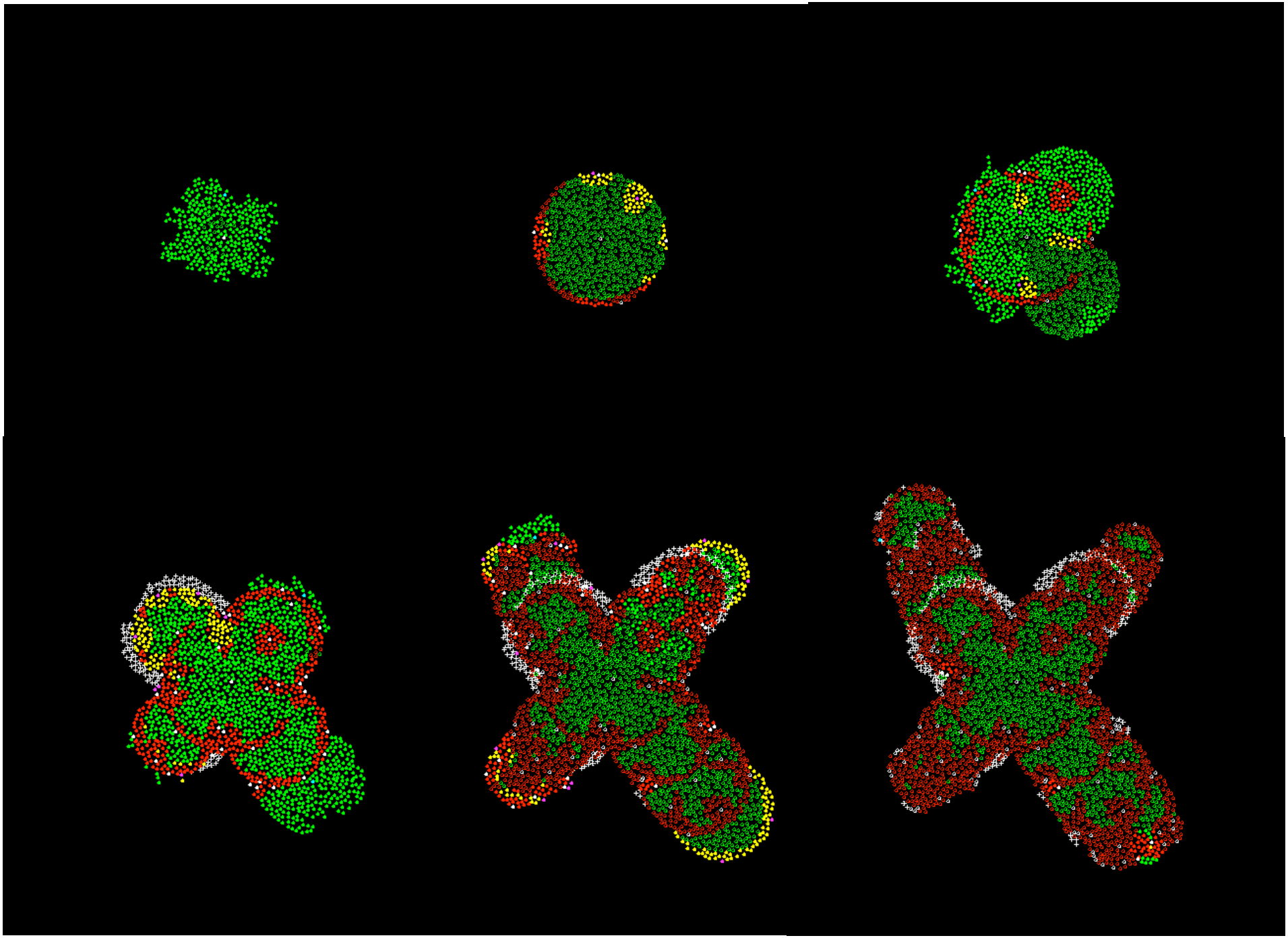
OSL Program for Drawing a CMOS Inverter

```
(define v1 (crease-l2l e23 e41))
(define v2 (crease-l2l v1 e23))
(define v3 (crease-l2l v1 e41))
(define IN  (create-region e41 v3))
(define OUT (create-region e23 v1))
(define MID (create-region v1 (or v2 v3))
;; Similarly create horizontal regions...
```



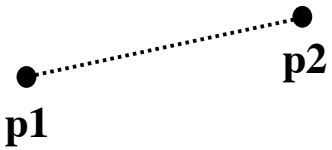

```
;; Lay down Material (differentiate)
(within-region IN (color h1 "poly"))
(within-region MID (color (or h2 h3) "poly"))
(within-region OUT (color h1 "poly"))
(within-region CNTR (color v3 "poly"))

(within-region UP (color v1 "n-diff"))
(within-region DOWN (color v1 "p-diff"))
(define contacts (intersect v1 (or e12 h1 e34))
(color contacts "contacts")
```

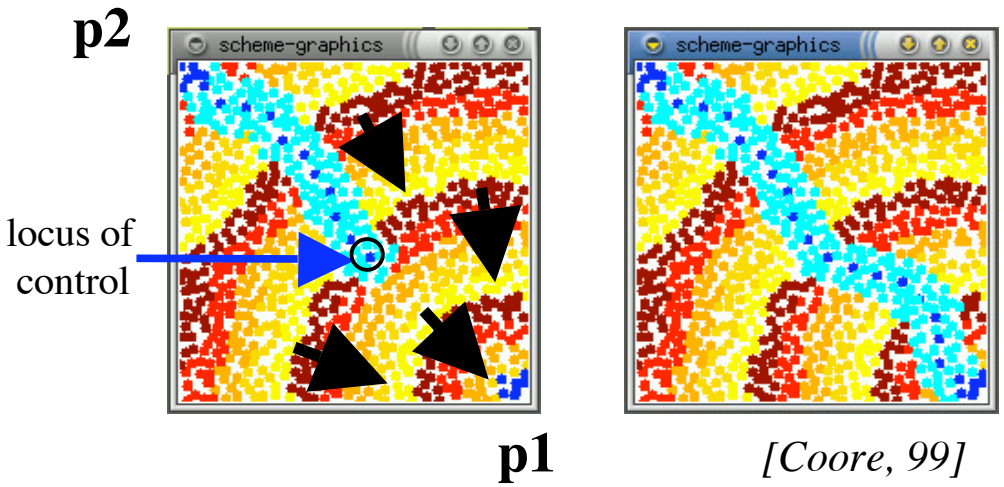




Implementing the Rules

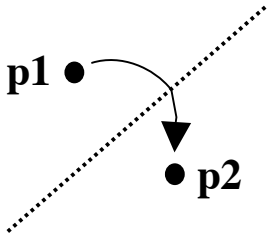
A1: (crease-lbp p1 p2)

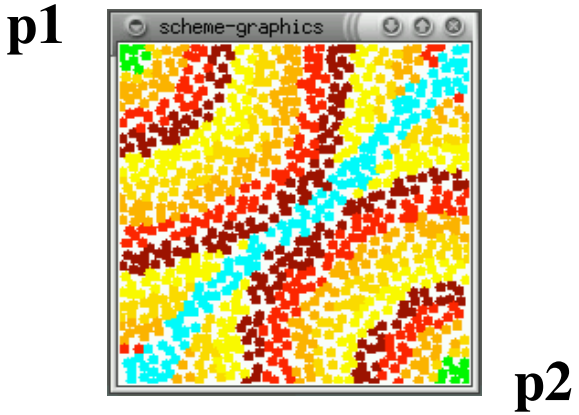




[Coore, 99]

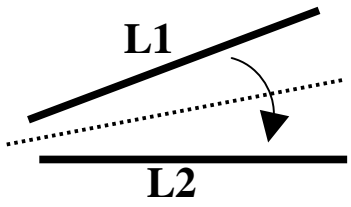
A2:(crease-p2p p1 p2)

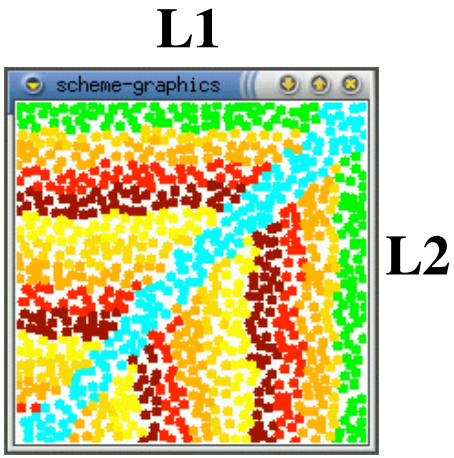




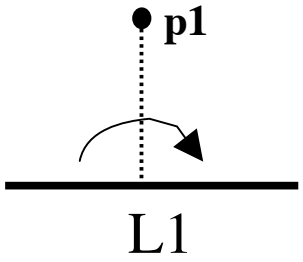
Implementing the Axioms

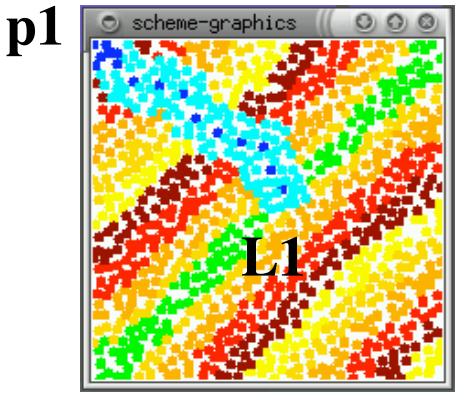
A3: (crease-l2l l1 l2)





A4: (crease-l2s p1 l1)



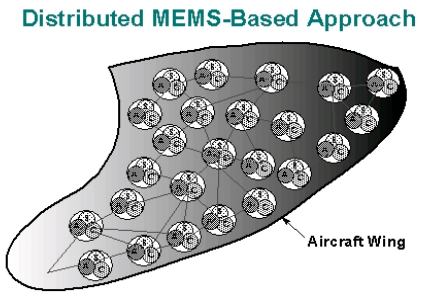


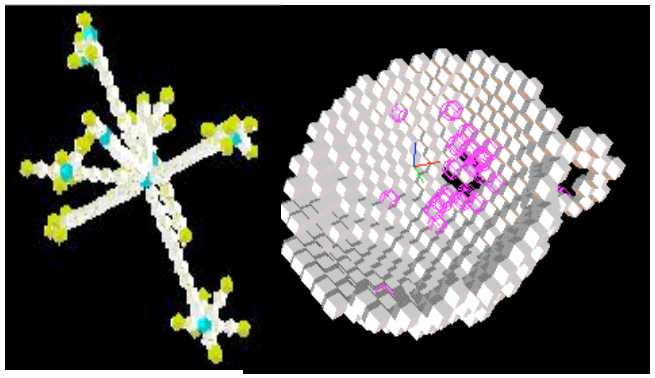
More Info:

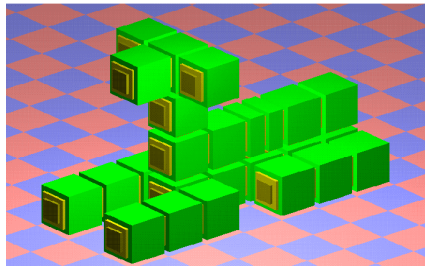
- **Reaction Diffusion:**
  - Amorphous Computing Gray Scott page (also applet) (http://www.swiss.ai.mit.edu/projects/amorphous/GrayScott/).
  - "**Biological Pattern Formation**" by Hans Meinhardt (http://www.biologie.uni-hamburg.de/b-online/e28_1/pattern.htm),
  - **Visual Models of Morphogenesis: A Guided Tour** by Prusinkiewicz, Hammel, and Mech (http://www.cpsc.ucalgary.ca/Research/bmv/vmm-deluxe/)
  - (lots of links to papers and books)

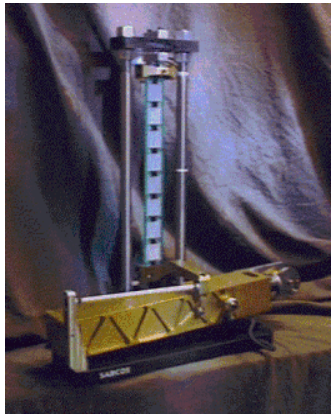
- **Lateral Inhibition:**
  - "Making of a Fly", by Peter Lawrence (Chapter on Spacing)
  - "From Egg to Embryo" by Jonathan Slack (Chapter 2 on theoretical models of gradients, and patterning using gradients)

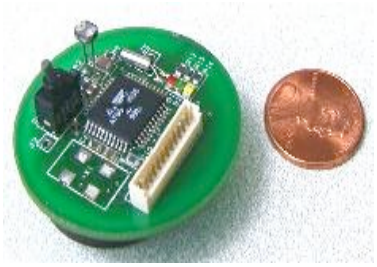
Systems Composed of Many Parts

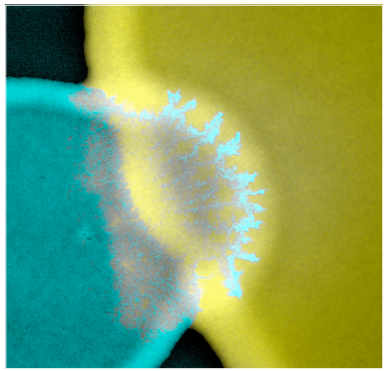






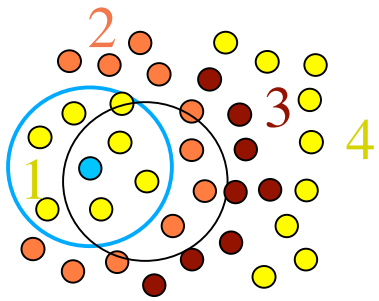


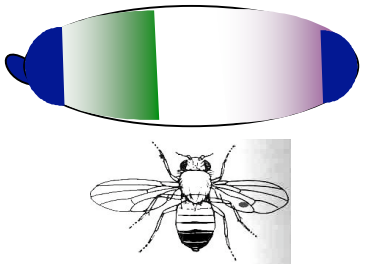


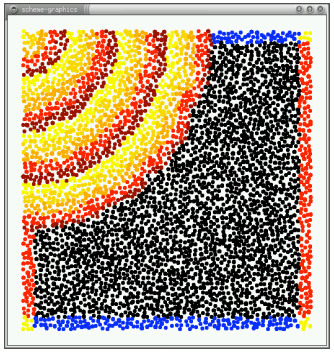


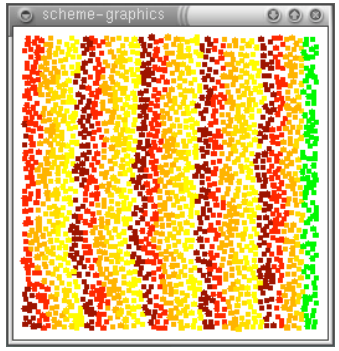
Biologically-Inspired Primitives for local interactions

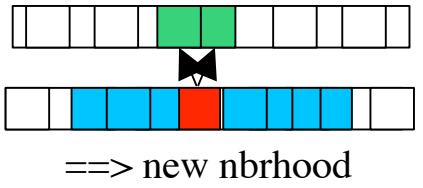
1. Morphogen Gradients
2. Cell-to-cell Contact
3. Neighborhood query and selection
4. Local Inhibition





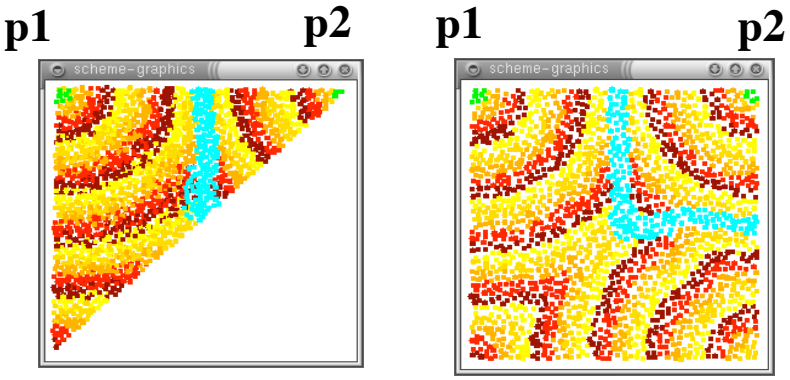




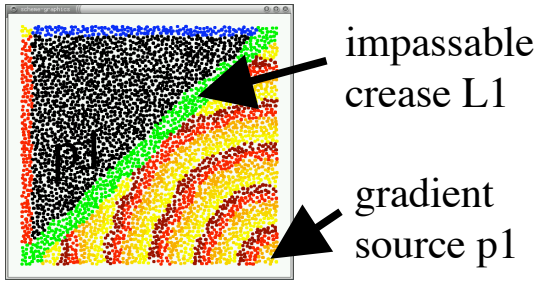


Implementing Other Operations

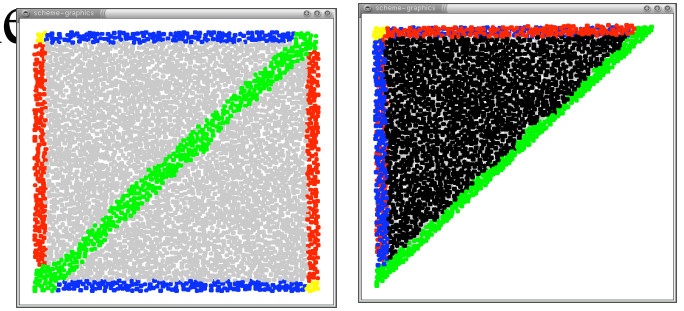
- Cell-to-cell contact
  - Sheet acts a single layer



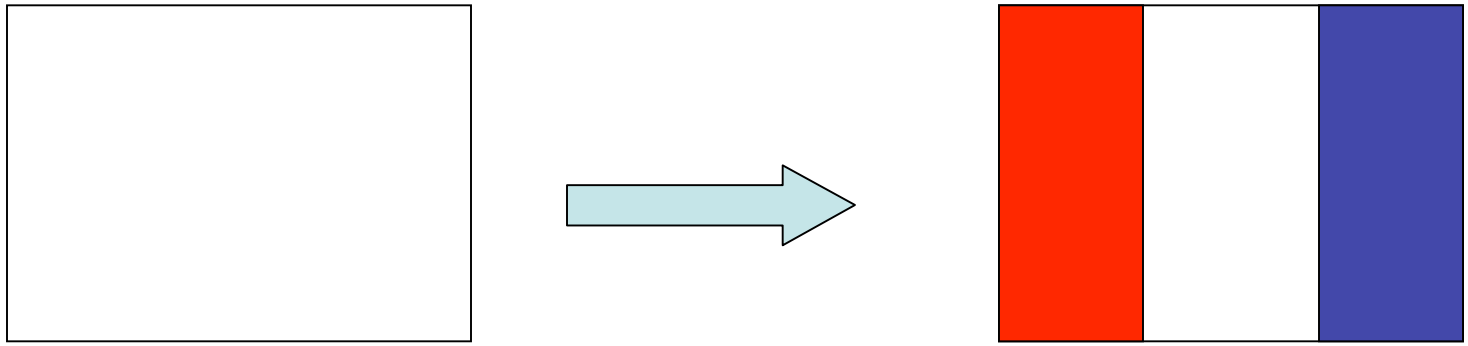
- Region
  - Bounded gradients



- Execute-Fold
  - fold is approximated by the
  - simulator



Positional Information using Gradients of Morphogens



(French Flag Model, Wolpert)

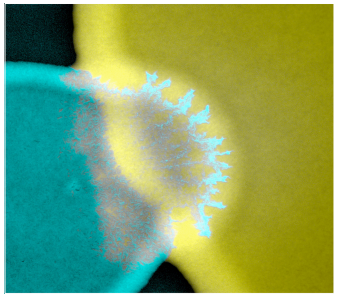
Biologically-Inspired Distributed Computing

Design systems capable of self-organization, self-repair

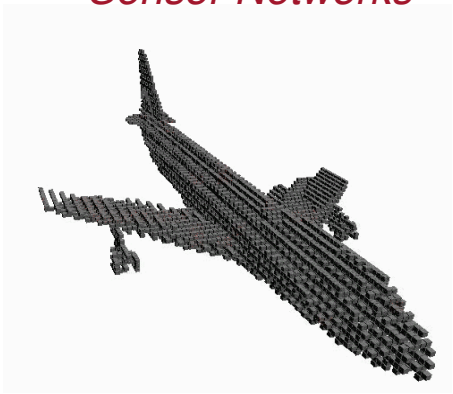
Distributed Systems

Swarm robotics

Sensor Networks

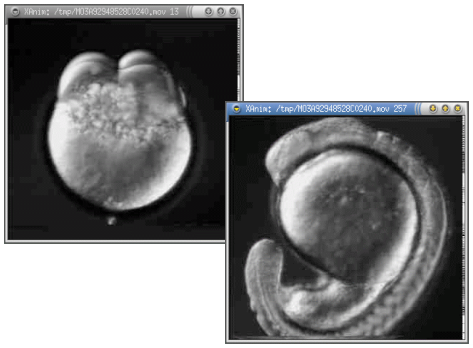


Synthetic Biology



Reconfigurable Robots

Inspiration: Biological Systems

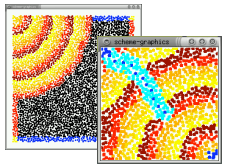


Programming Myriads

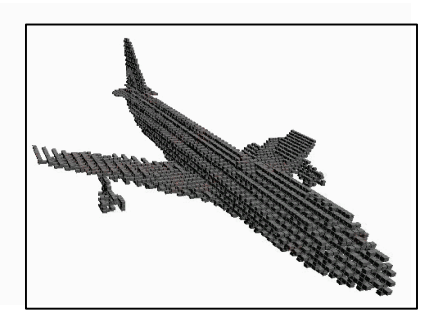
High-level Programming Languages
in terms of Goals / Tasks

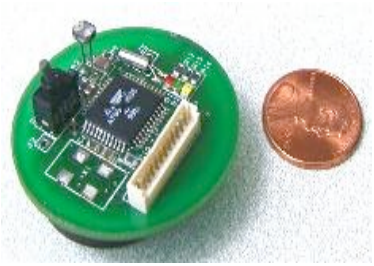
Global-to-Local COMPILER

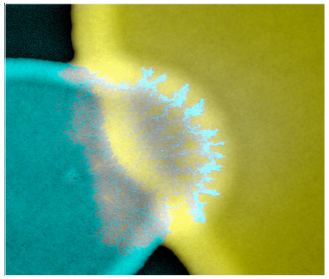
Biologically-inspired primitives for Robust Collective Behavior



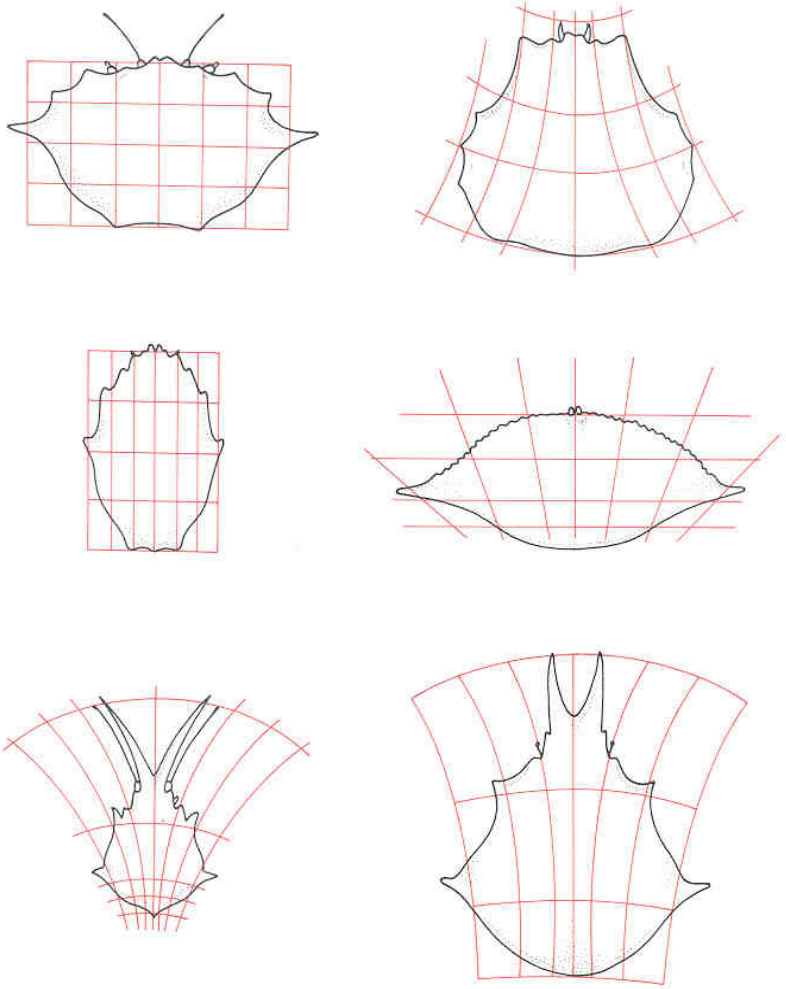
Program for Myriads of Agents

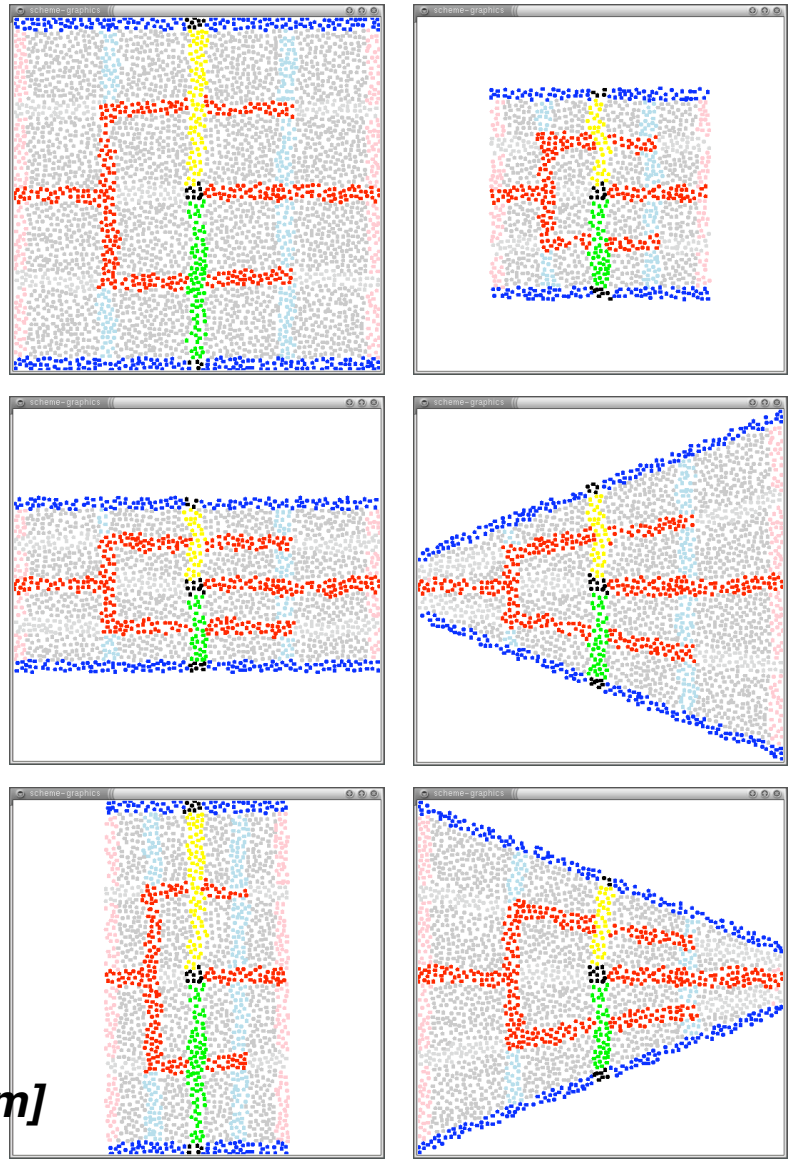






Asymmetric Scaling



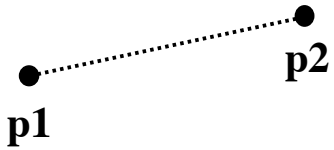


[D'Arcy W.Thompson, On Growth and Form]

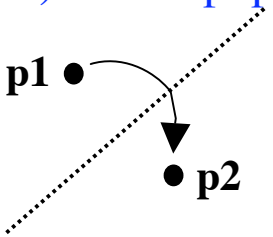
The Origami Shape Language (OSL)

- Huzita's Axioms of Origami + Other Operations

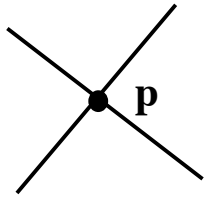
(A1) crease-lbp



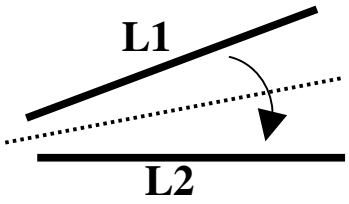
(A2) crease-p2p



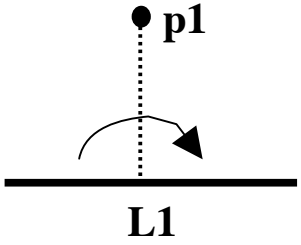
intersect



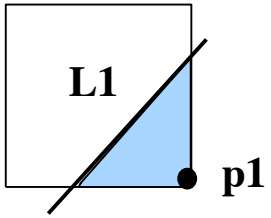
(A3) crease-l2l



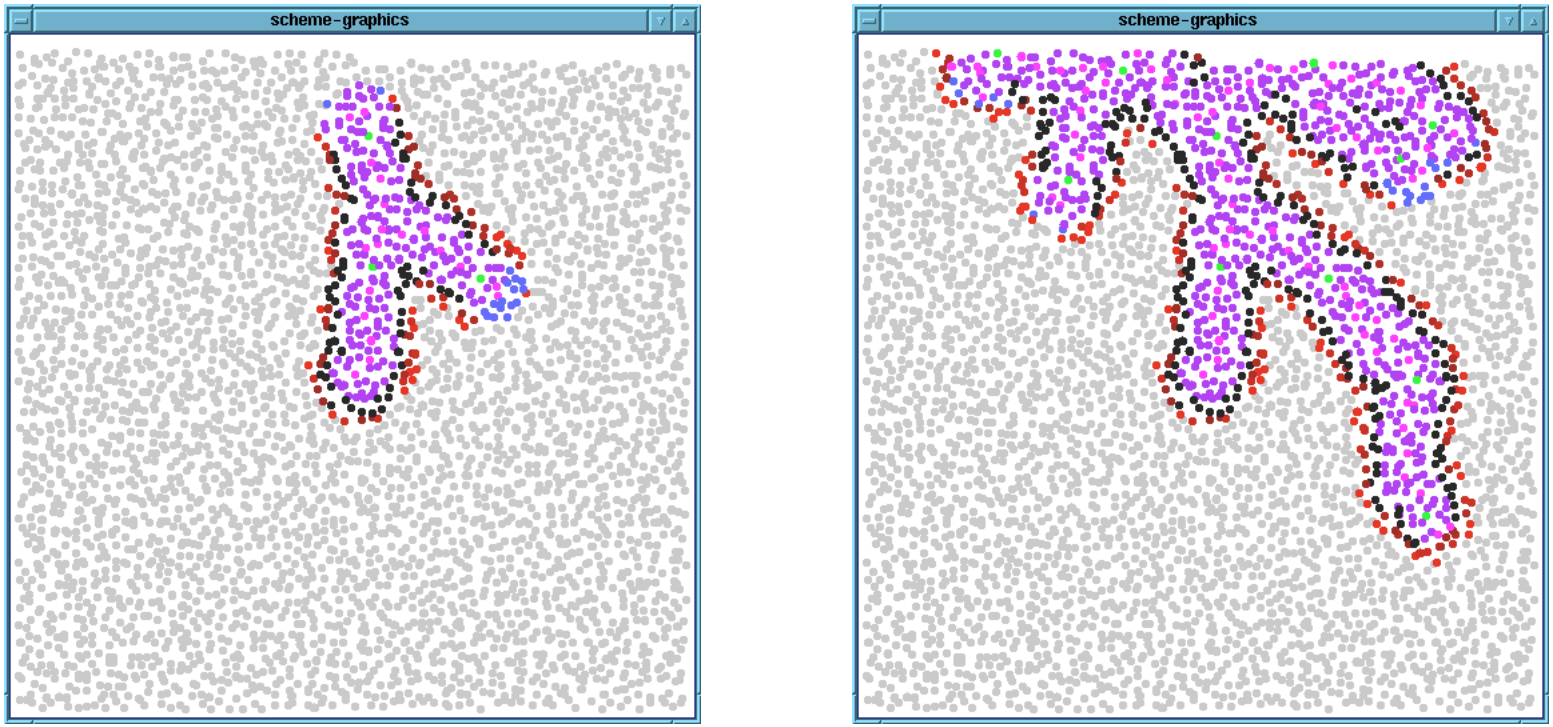
(A4) crease-l2self



region

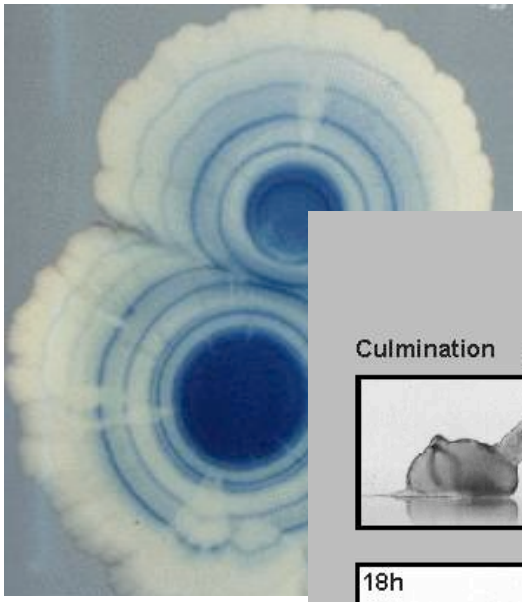


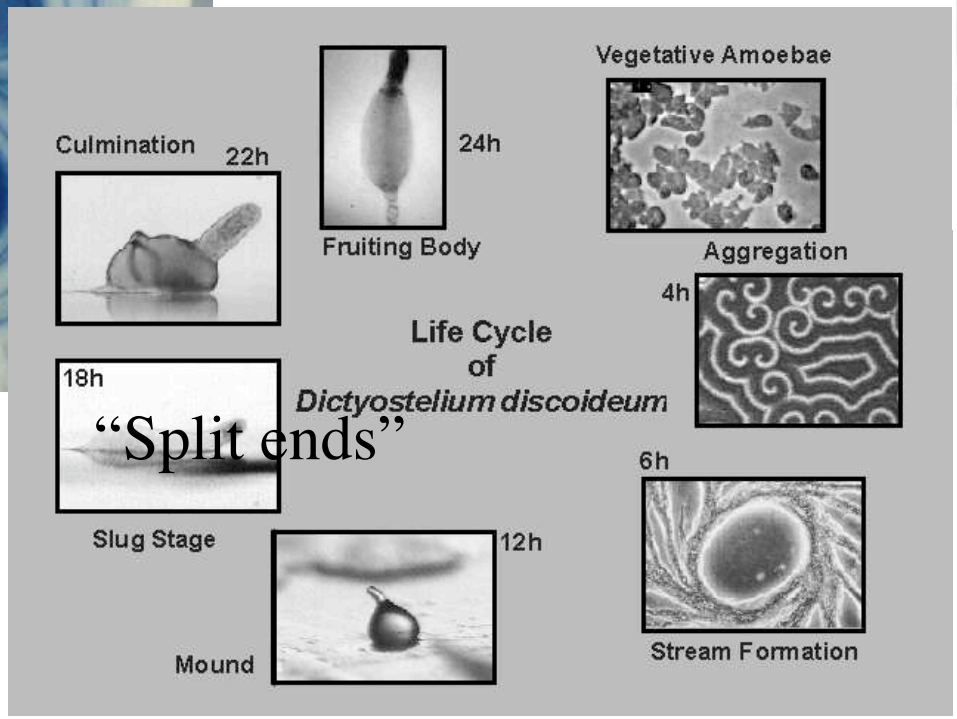
Bifurcating Tubes: an example of emergent behavior



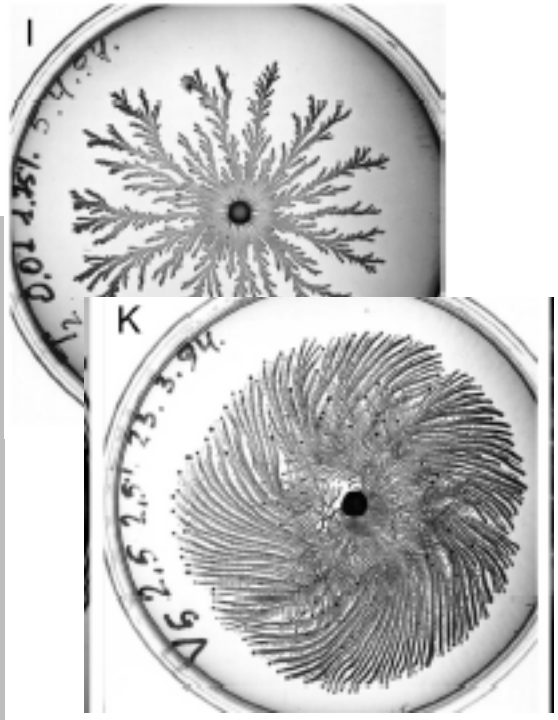
This is an amorphous physical simulation of a weak membrane bounding a pressure vessel. When a bulge appears, the membrane thins and the bulge expands. (by Radhika Nagpal)

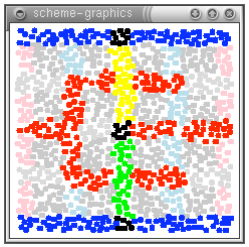
Patterns in Nature



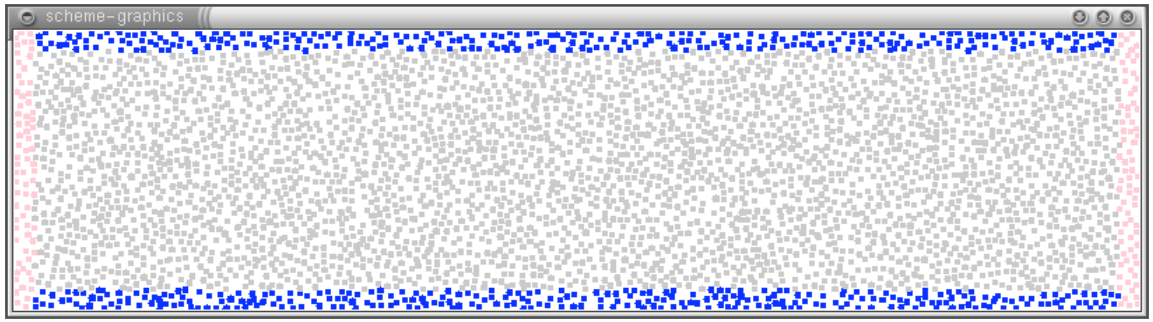


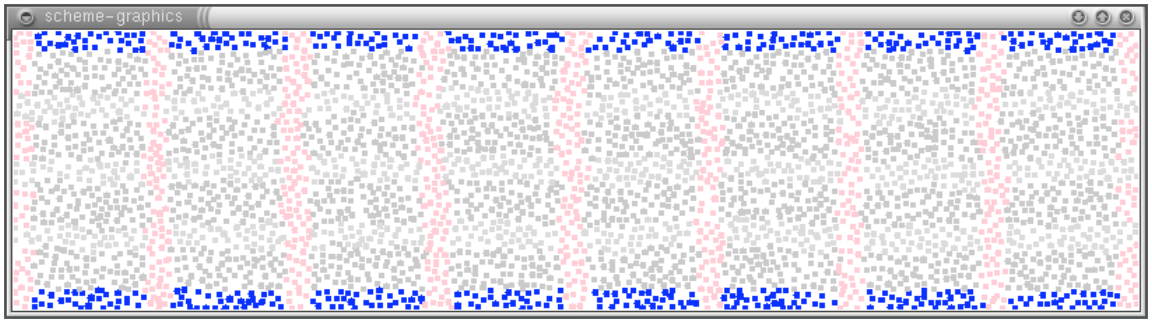
"Split ends"

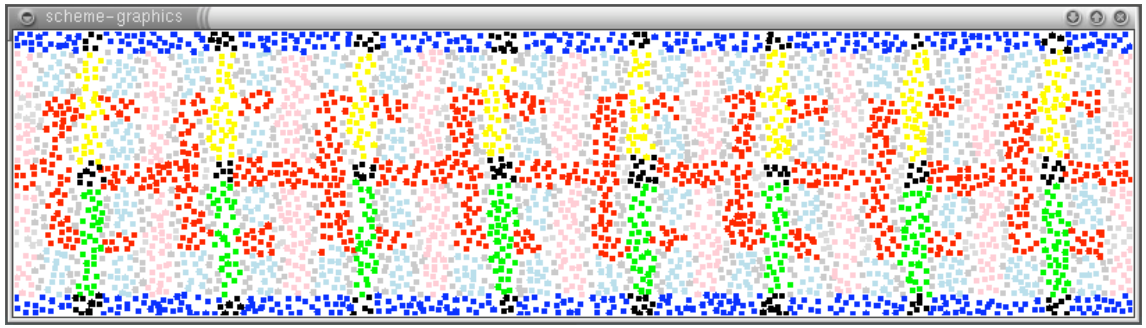




Chain of Inverters







Results

- Resource Usage
  - Small amount of local state, short-lived gradients
  - Code: mostly conserved

- Robustness
  - Good geometric accuracy, with sufficient density
  - Without relying on regular grids, global clock or beacons, unique global ids; reliable in the presence of random agent death
    - *Theoretically analyzable behavior*
  - *Limitations: regional death, malfunctioning agents*

Two Morphogens