Local-to-Global Algorithms in Biology: *Drosophila* and Beyond (maybe ...)

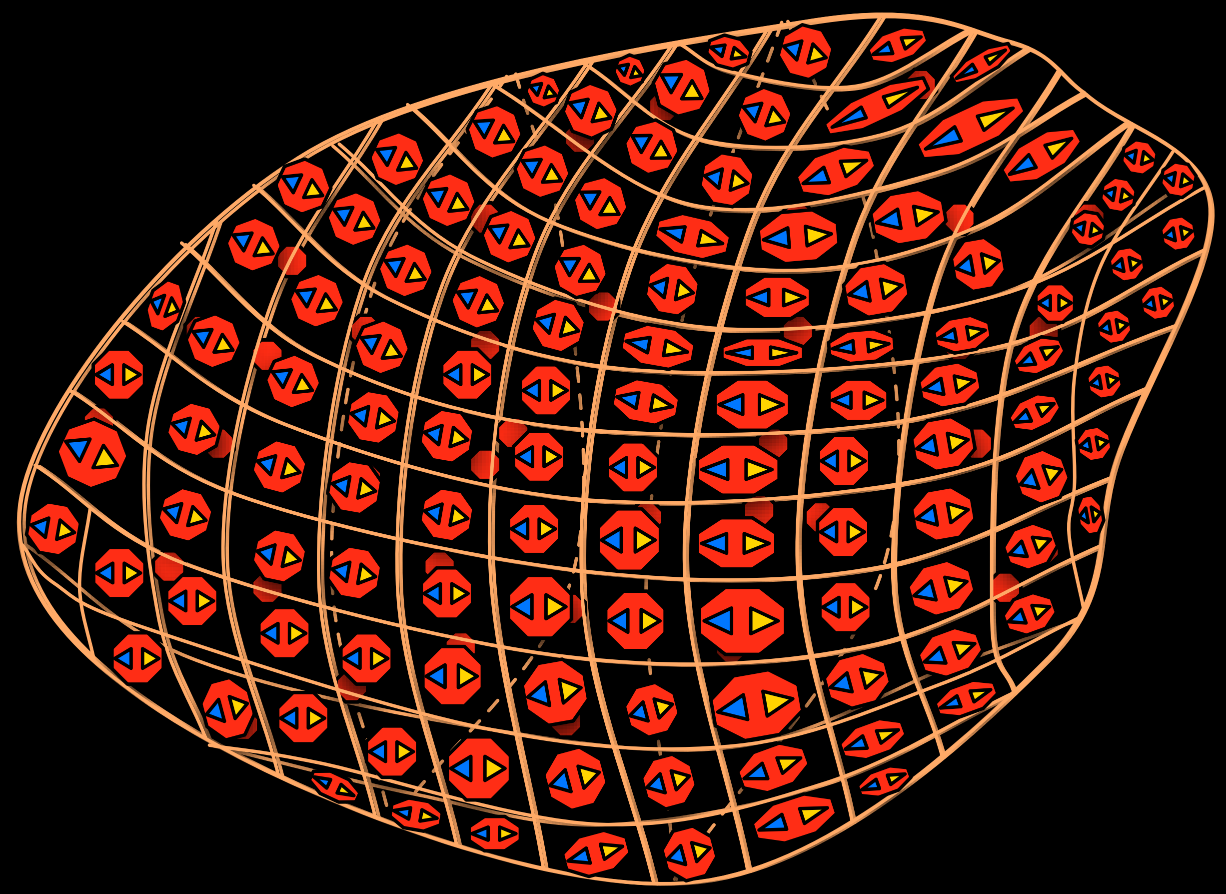
MIT Synthetic Biology Working Group

Dan Yamins

2007.12.05

spatial multi-agent systems

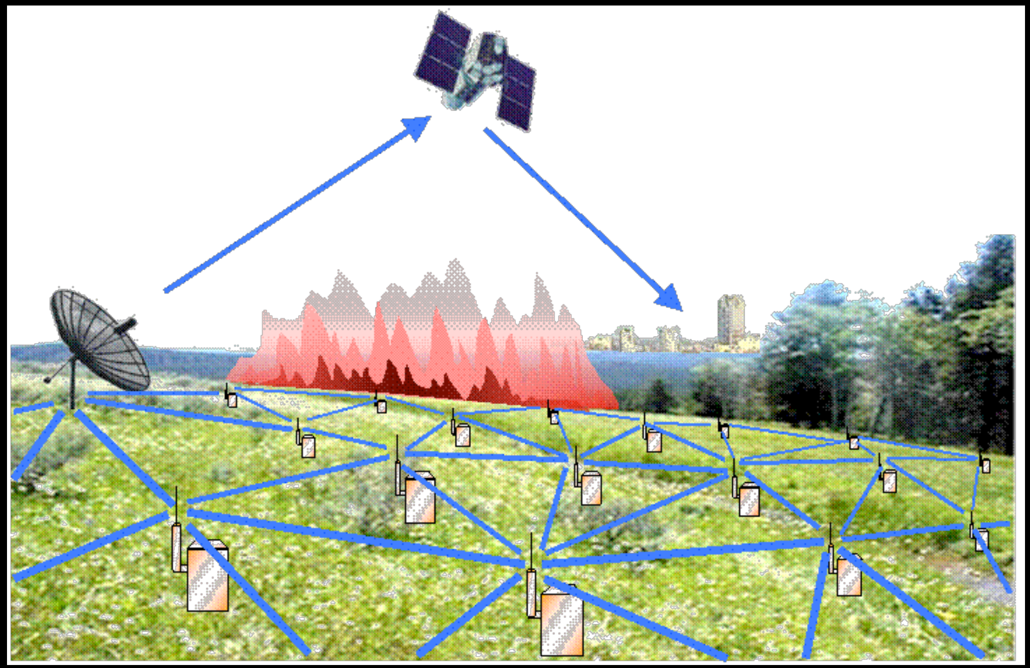
a space with agents embedded in the space



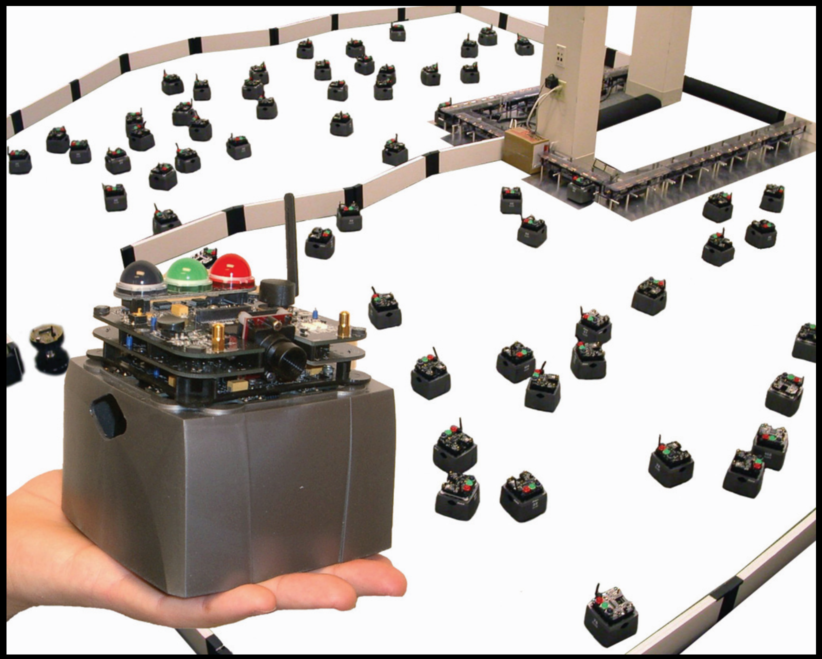
local information and processing

globally defined tasks

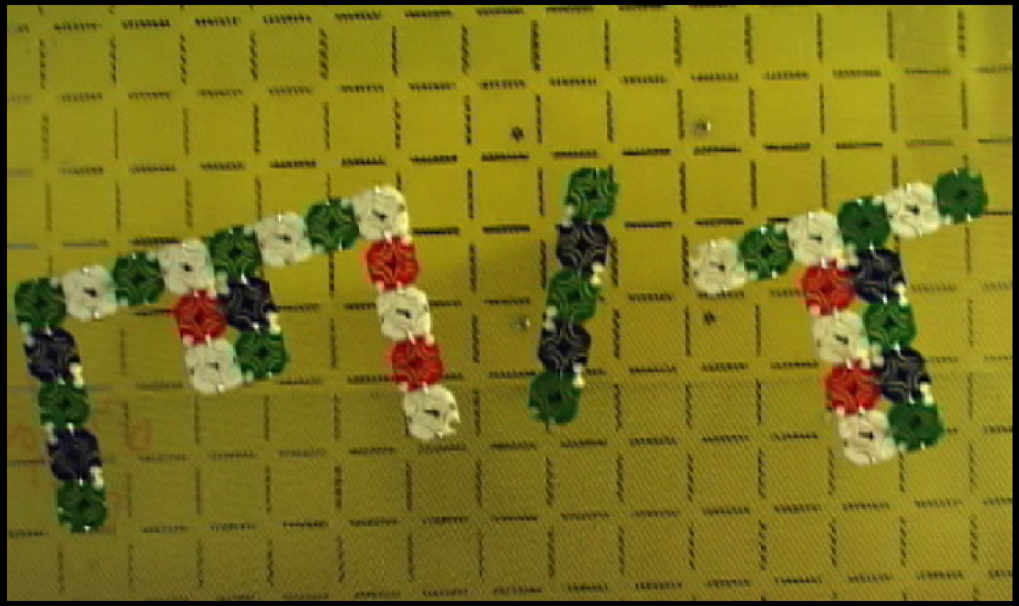
engineering:



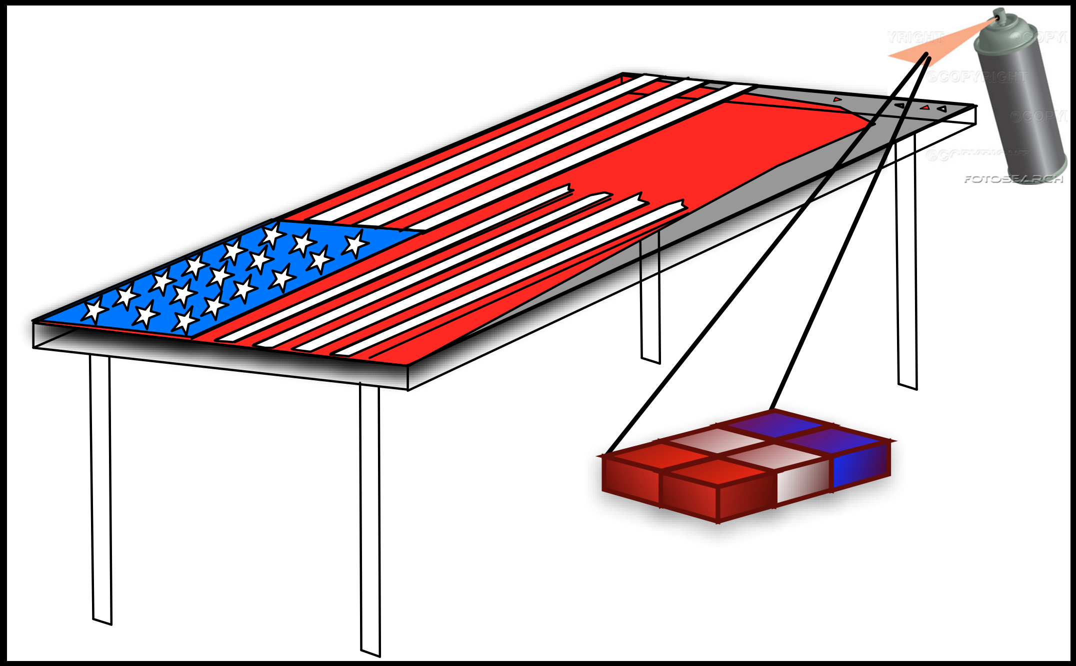
Sensor Networks



McLurkin iRobot Swarm

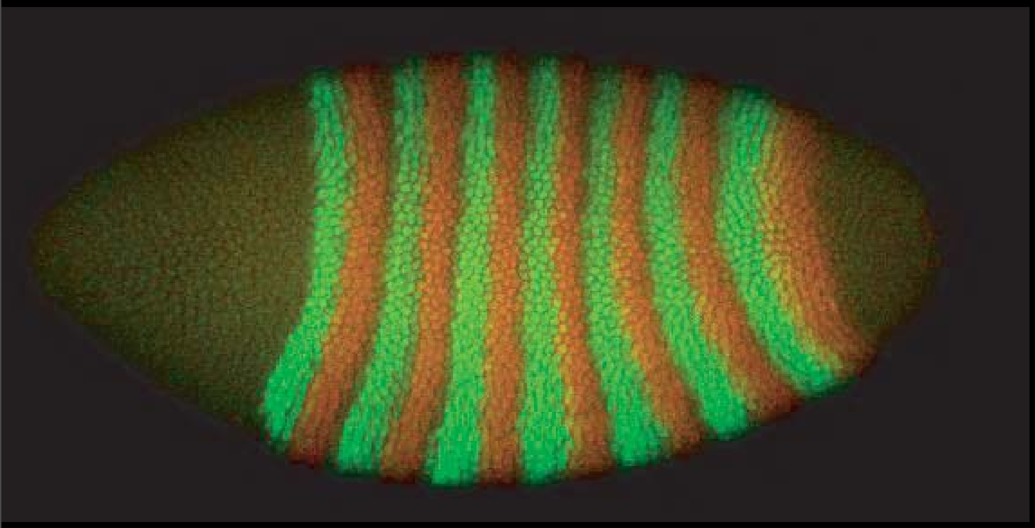


Saul Griffith's self-folding structures

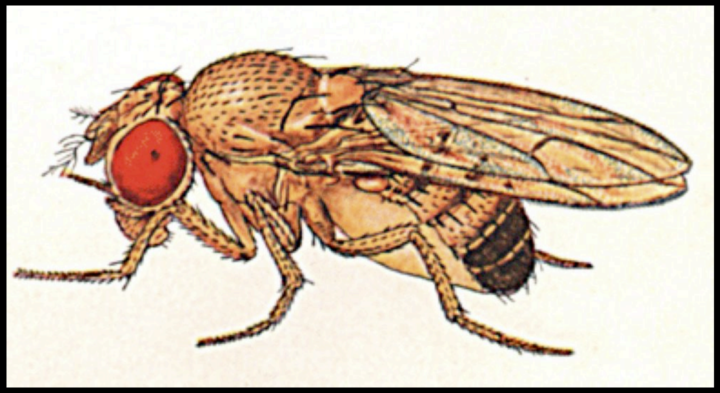


Butera's "paintable computer" concept

biology:



drosophila embryo



- What does thinking of natural spatial multi-agent systems *qua* computers tell us scientifically?
- What agent resource capacities are required to solve a given task?
- Can we then turn around this knowledge to build a better bug?

outline

I. Drosophila background

II. Multi-agent systems Model of *Drosophila*

III. An information bound

IV. The Radius/State Tradeoff

The first four sections study a known system, showing how multi-agent systems reasoning can help us understand features of local-to-global systems design.

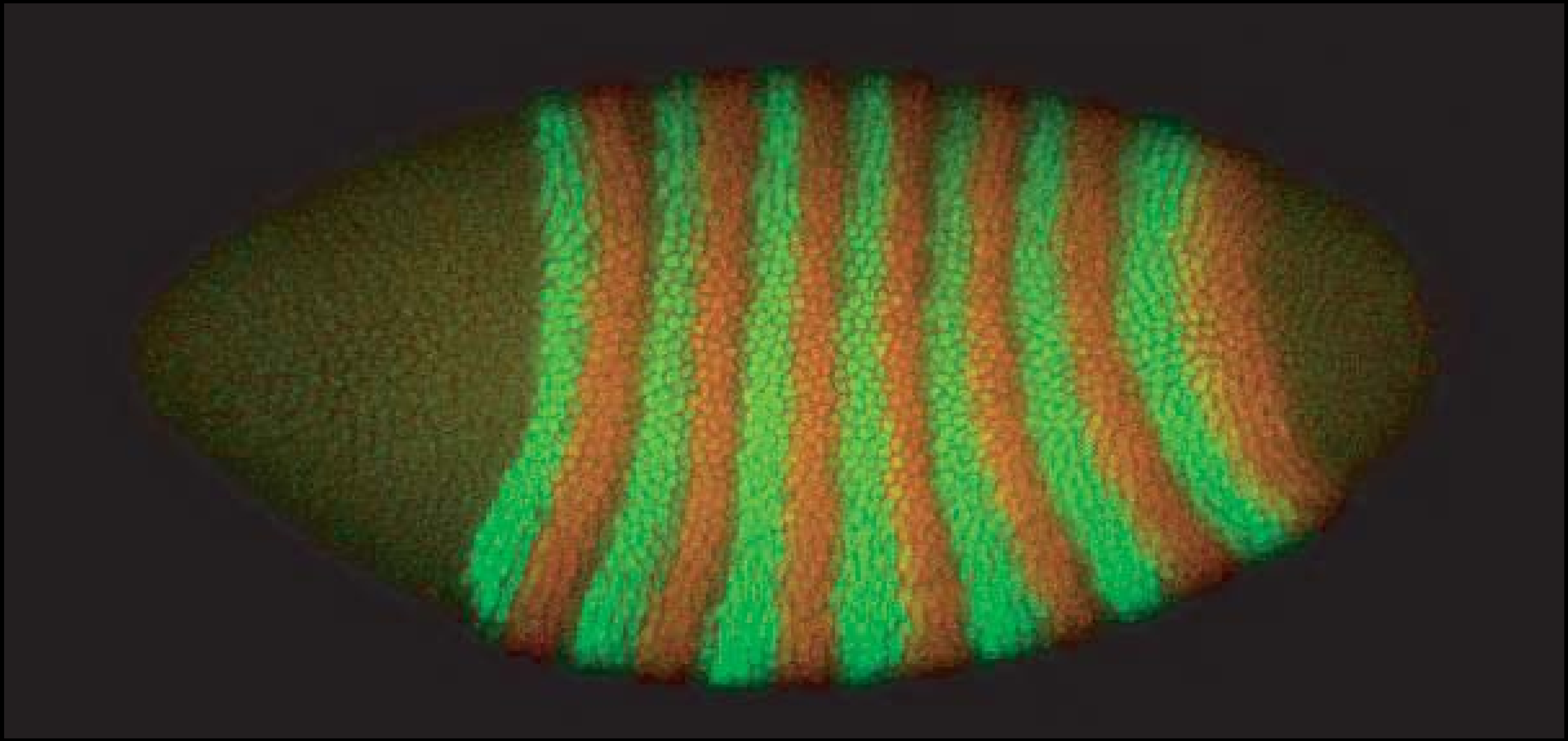
V. Designing Local Rules

The fifth section takes this knowledge, and applies it to develop the beginnings of a protocol for designing novel systems.

Drosophila
Background

(Actually ... one stage
in early *Drosophila*,
utterly simplified)

Drosophila



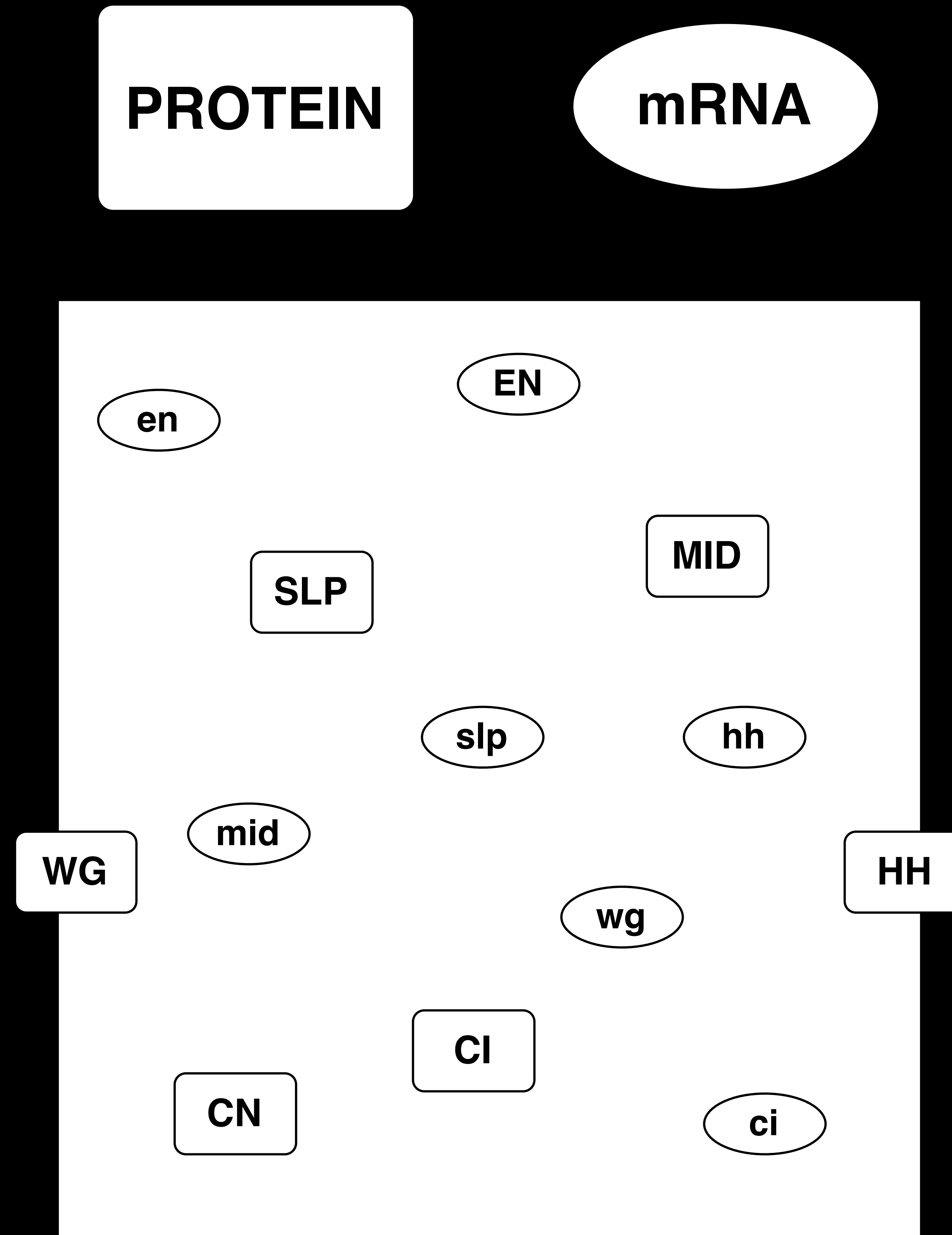
Drosophila melanogaster embryo
about 100 minutes post-fertilization

Drosophila

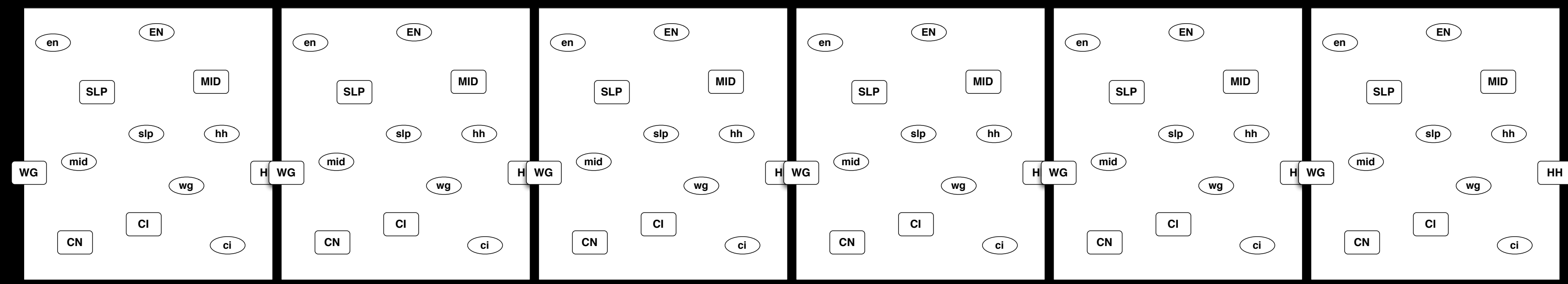
Let's think of the cell as a ....

.... bag of biomolecules

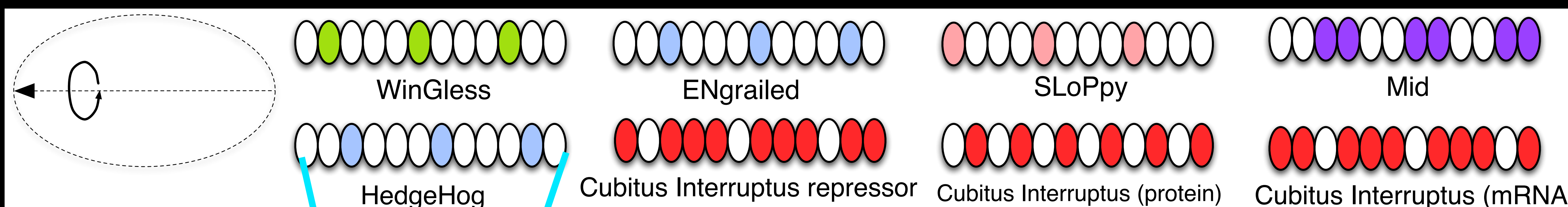
HIghlighted here are 13 substances that play are involved in the stripes in the picture on the previous slide

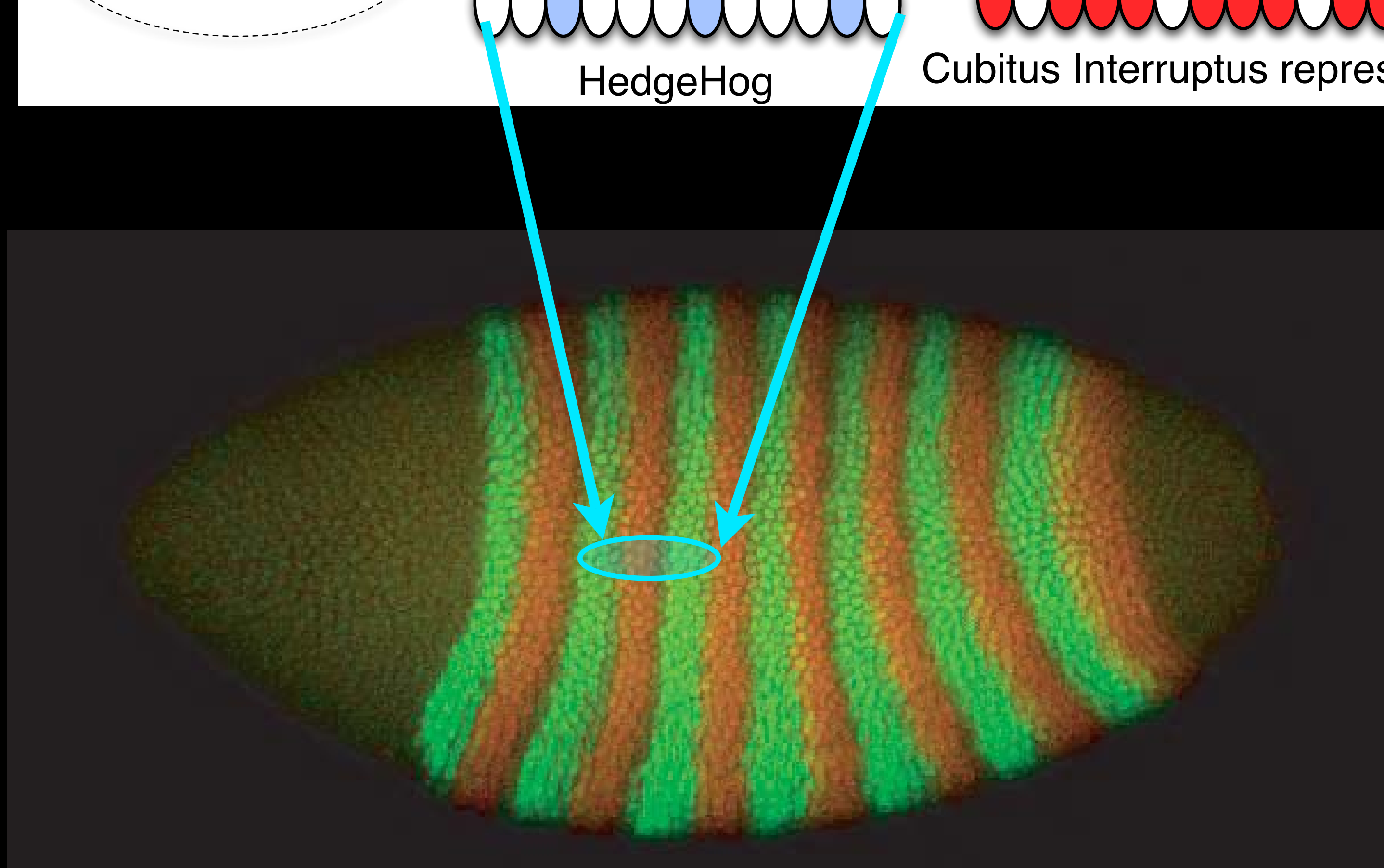


Considering cells along the A-P axis:



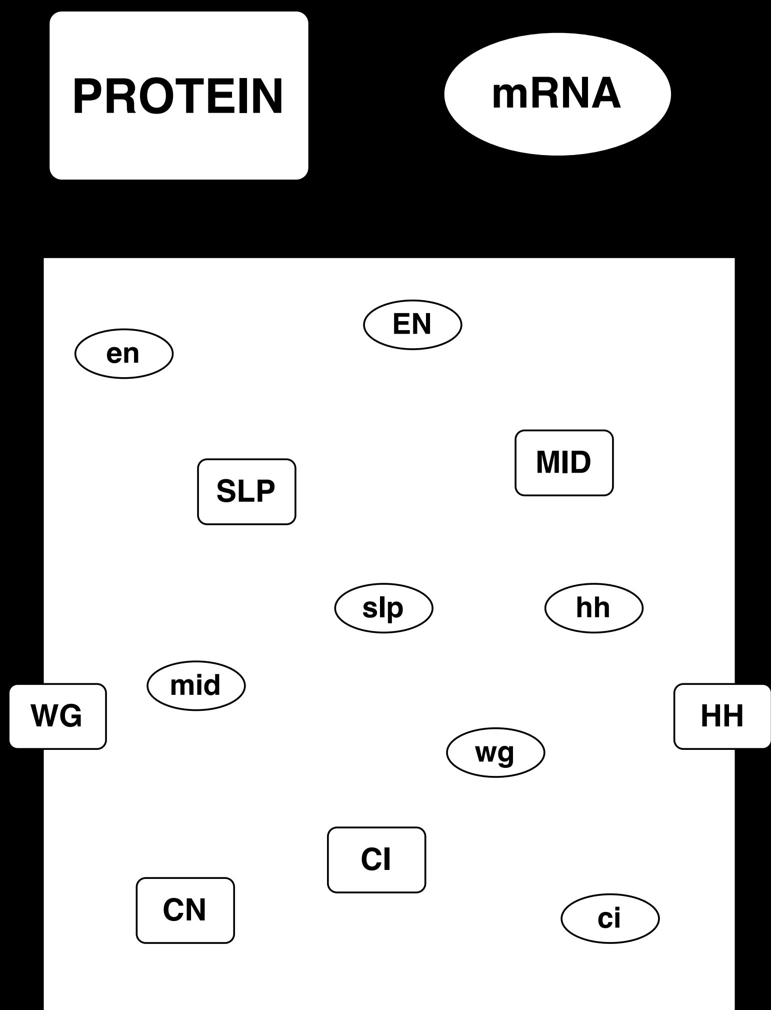
... there is a periodic pattern in the concentration of each of various substances depicted on the previous slide:



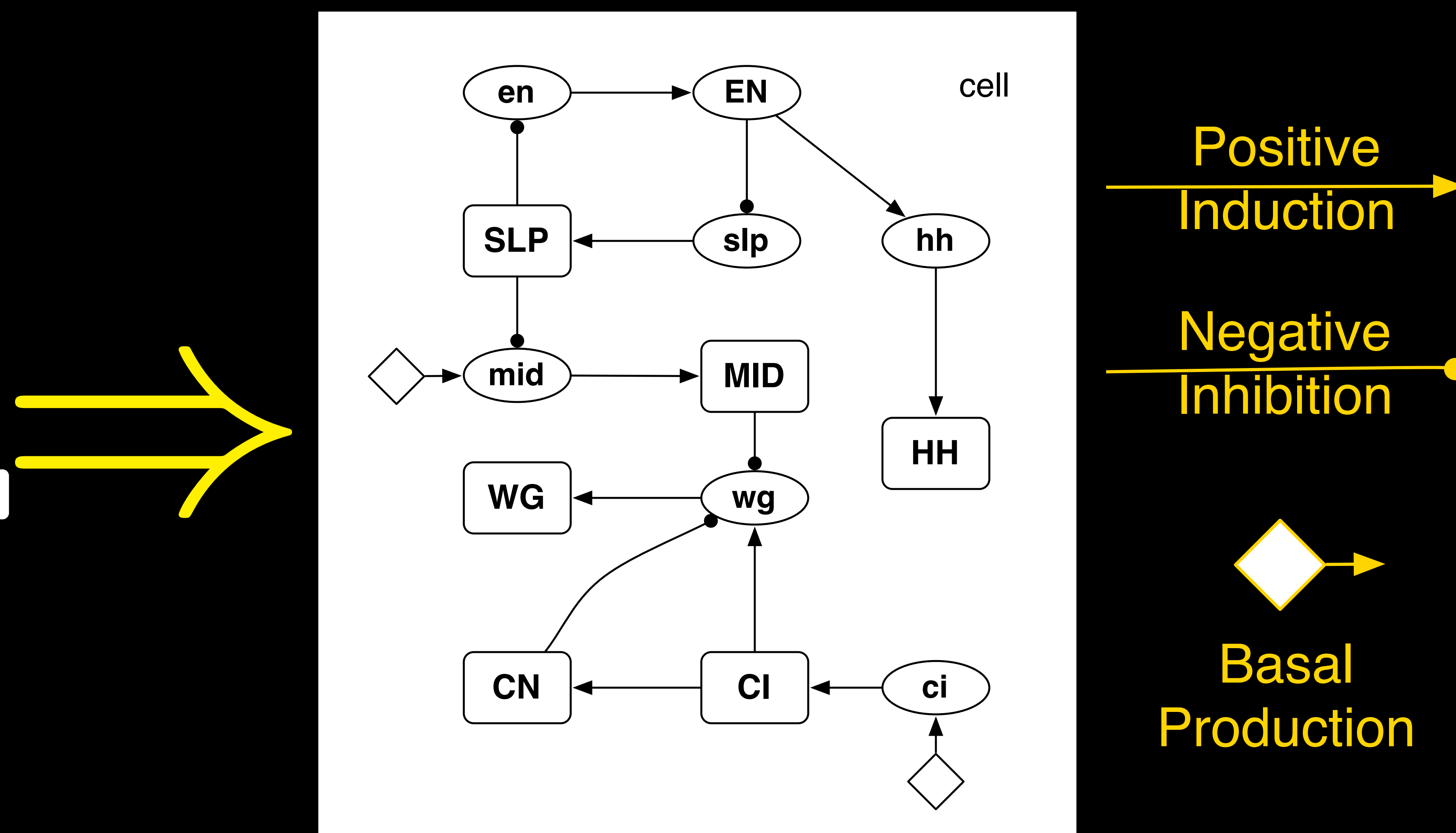


The concentration profile pattern at the molecular level corresponds the observed stripe pattern.

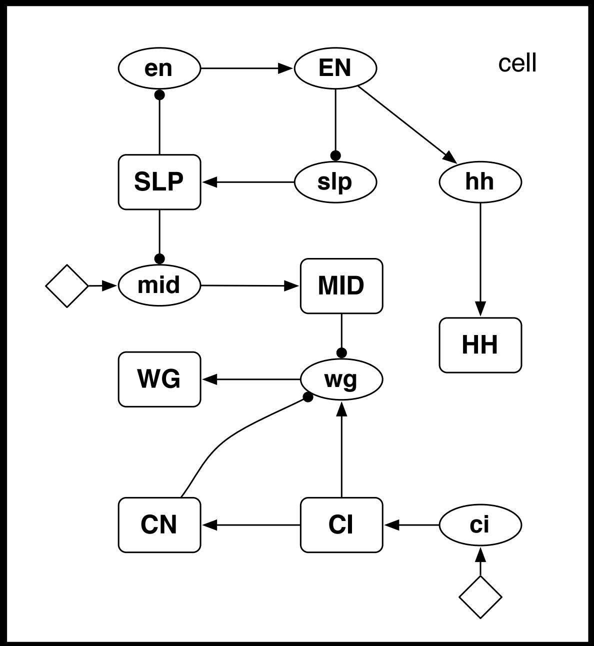
Drosophila



The bag of biomolecules ...

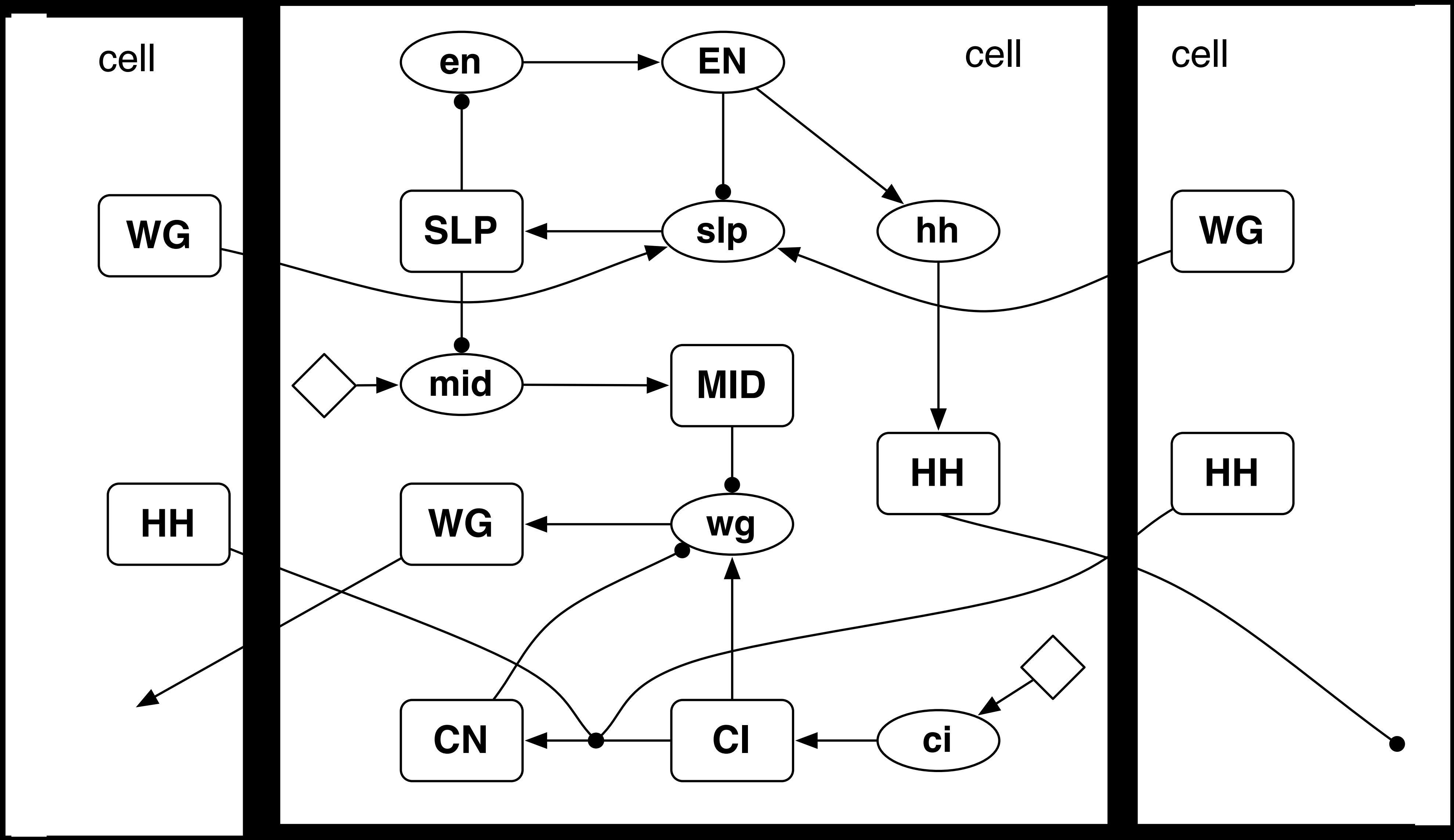


... is actually a Gene Regulatory Network

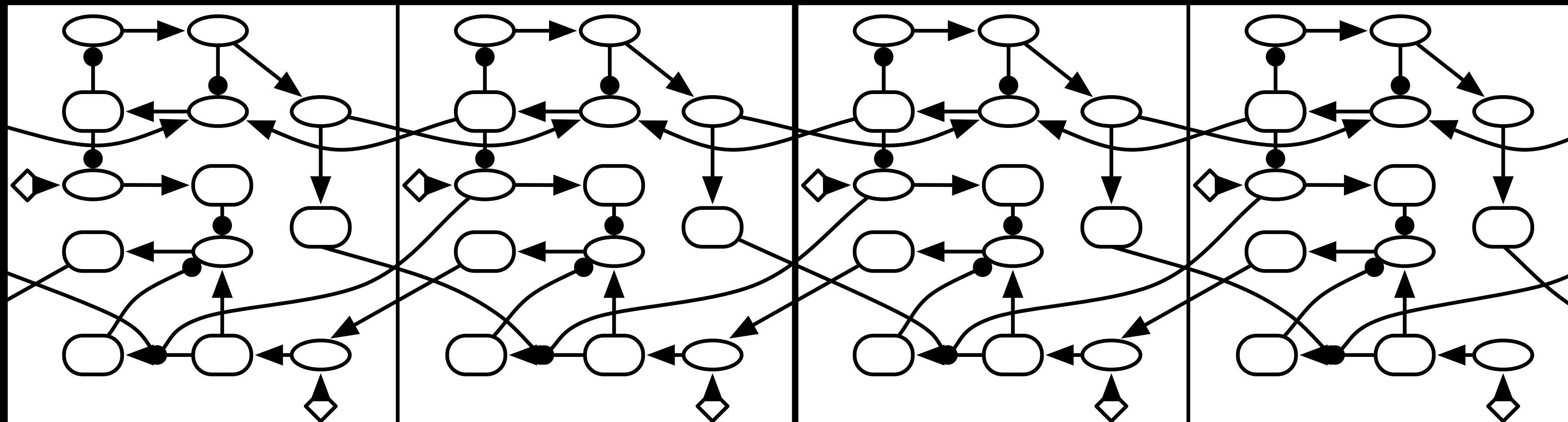


Some components of the Gene Regulatory Network in one cell ...

... influence those in neighboring cells.



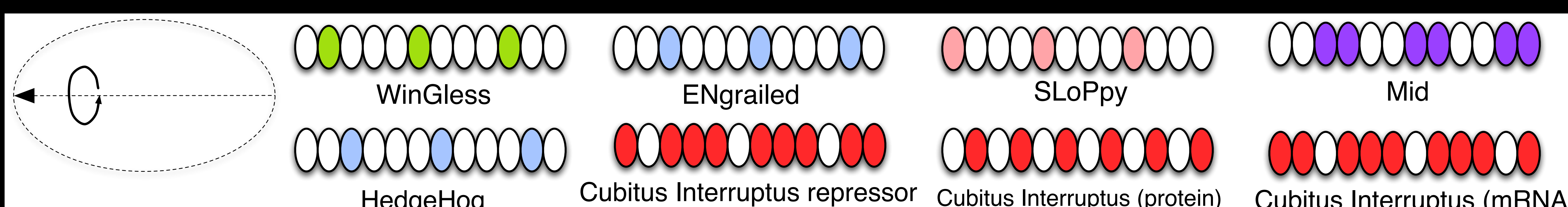
It's a spatially coupled Gene Regulatory Network.



Such a network can be modeled as a coupled ODE:

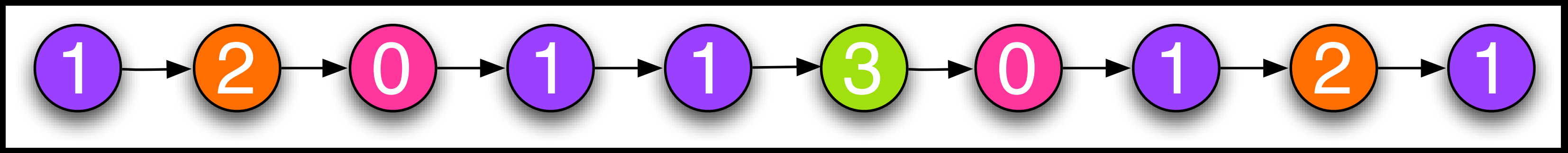
$$\frac{dS_j^i}{dt} = f_j\left(S_1^i, S_2^i, \ldots, S_N^i, S_1^{i-1}, S_2^{i-1}, \ldots, S_N^{i-1}, S_1^{i+1}, S_2^{i+1}, \ldots S_N^{i+1}\right)$$

And the ODE will (a la Garrett Odell) have the pattern as a *stable nondegenerate steady state:*



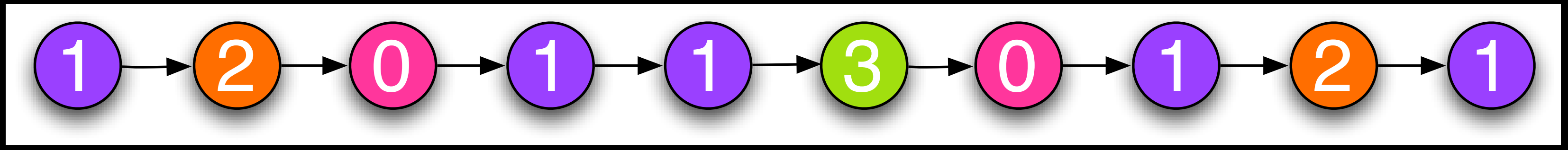
a multi-agent model of drosophila

(a *sketchy* multi-agent model of drosophila)

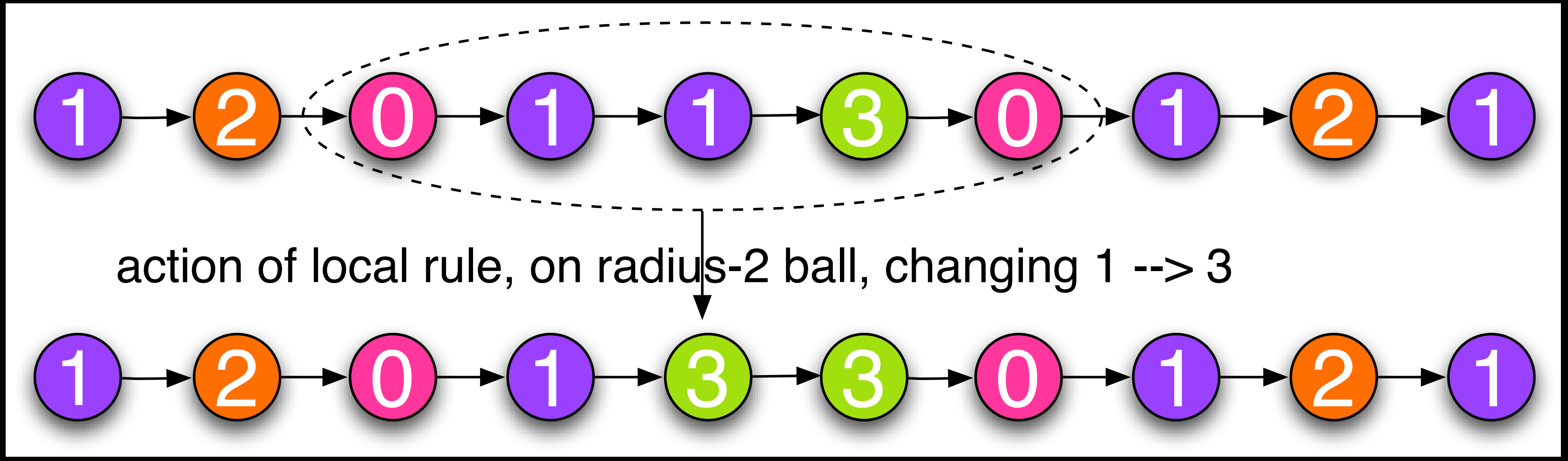


... a simple "1-D" organism, each cell of which has some internal state (0,1,2,3, etc...)

multi-agent model

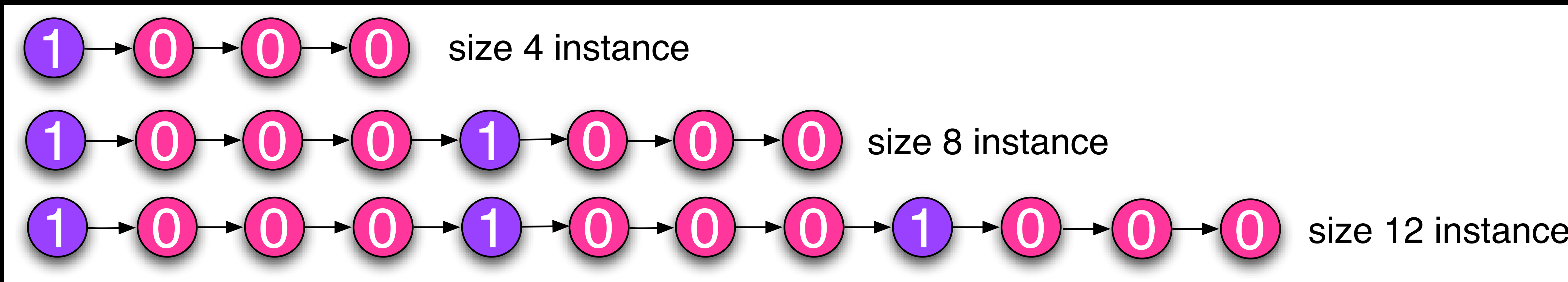


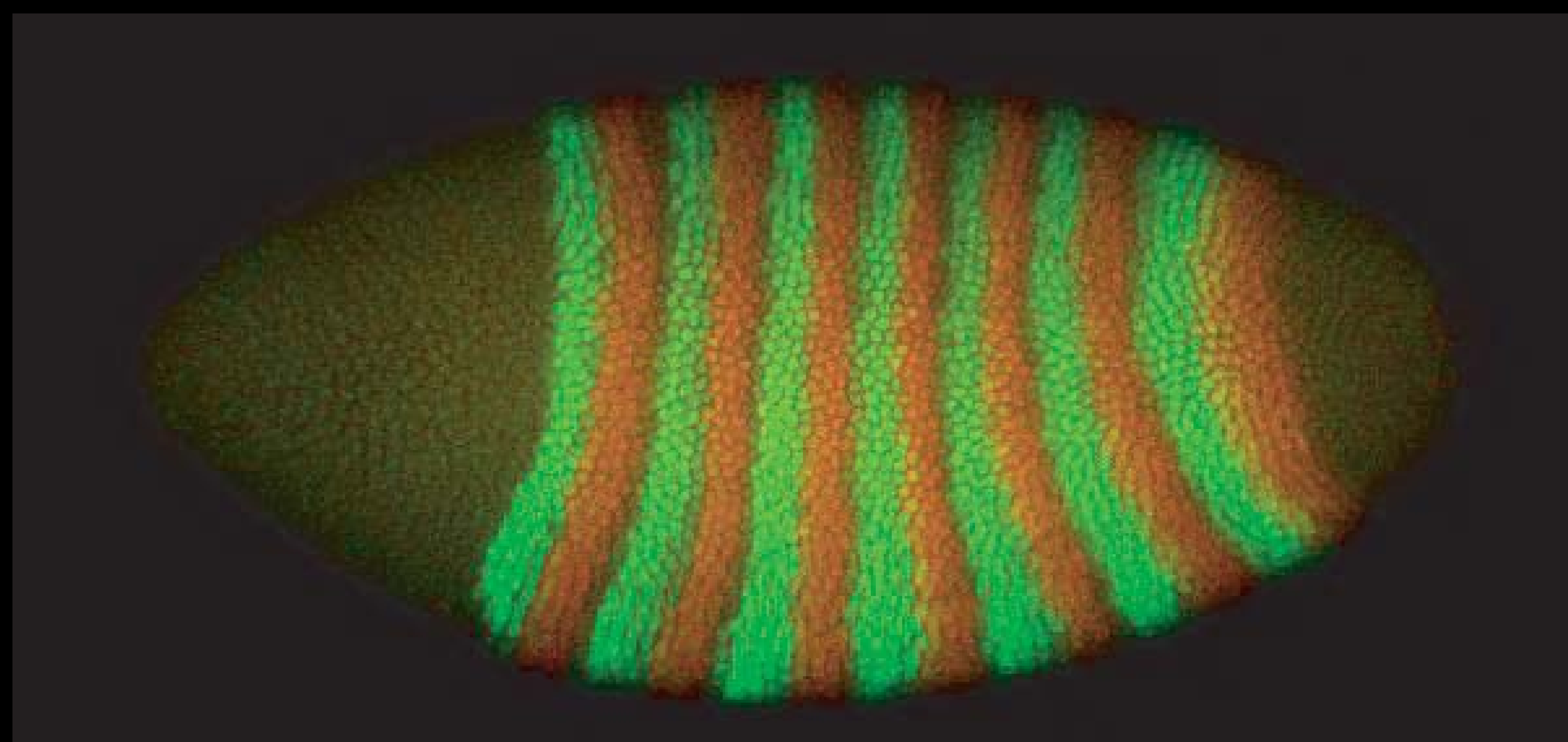
local rules by which the agent integrates information about states of nearby agents and takes a differentiation action

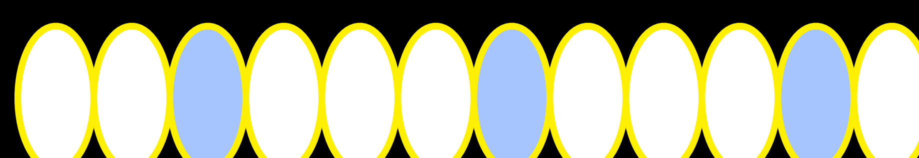


any radius-1 (nearest neighbor) local rule is allowed.

multi-agent model

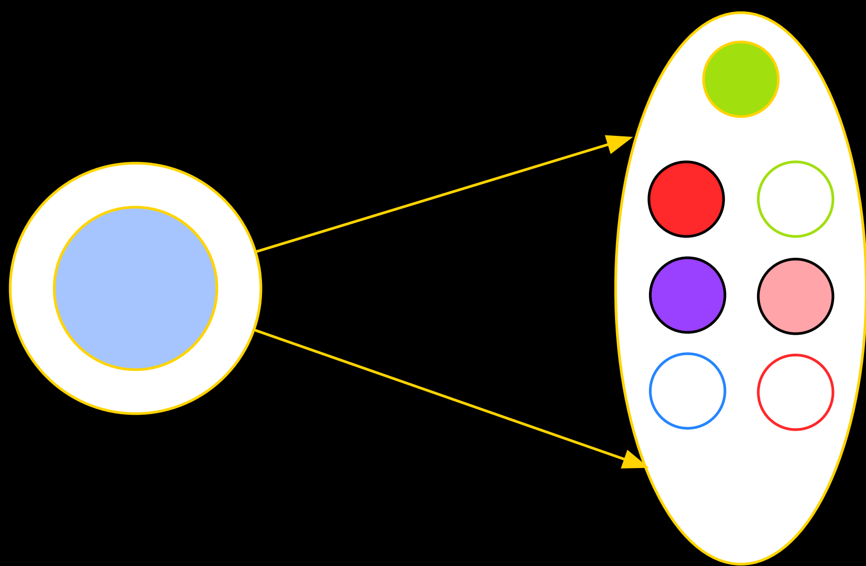






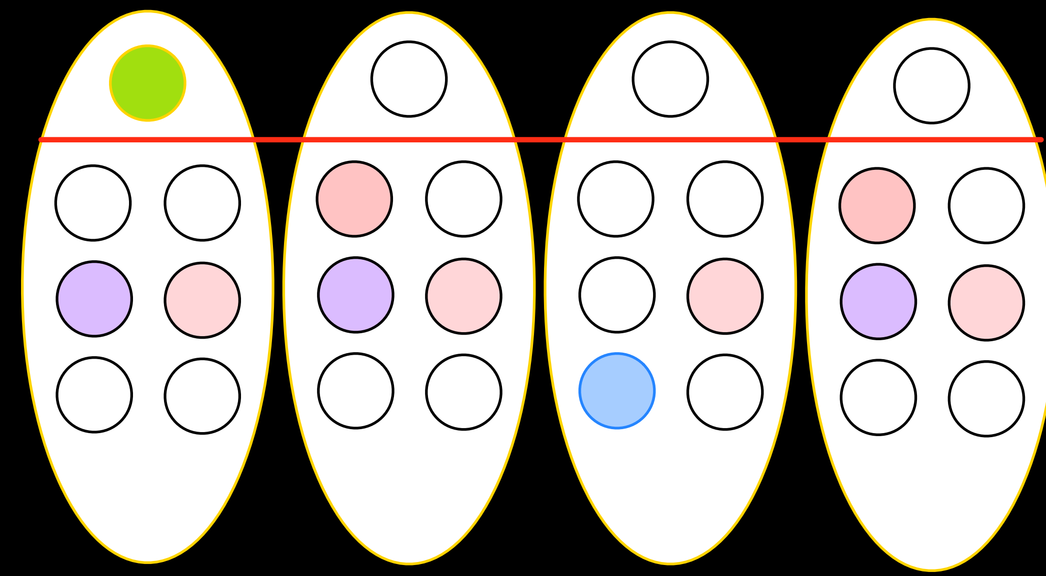
ENgrailed

multi-agent model

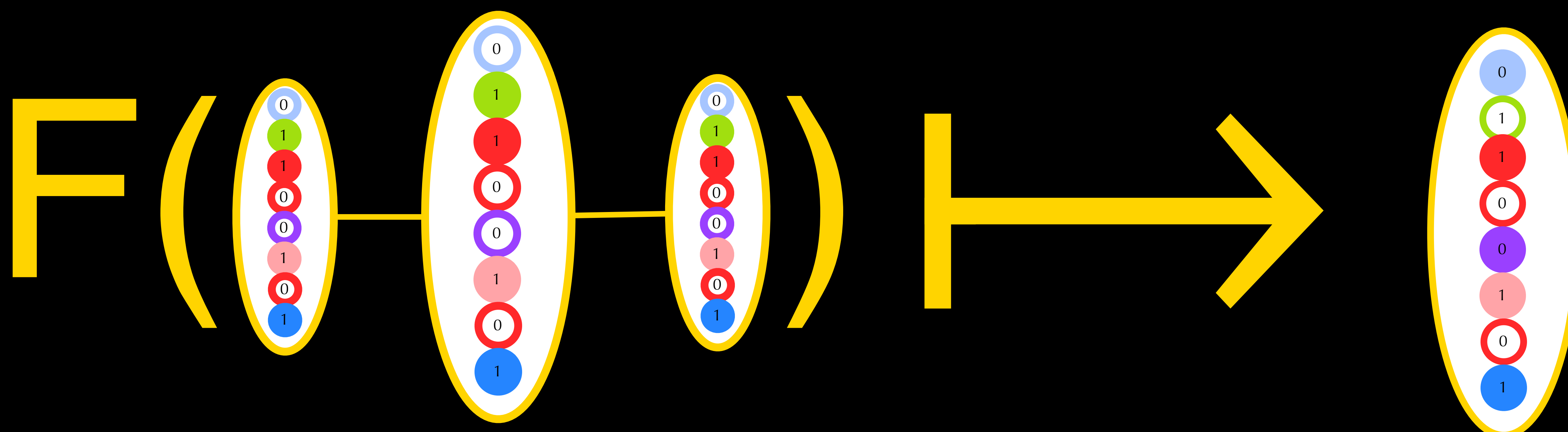


Agent with unstructured internal state.

Add “slots.”

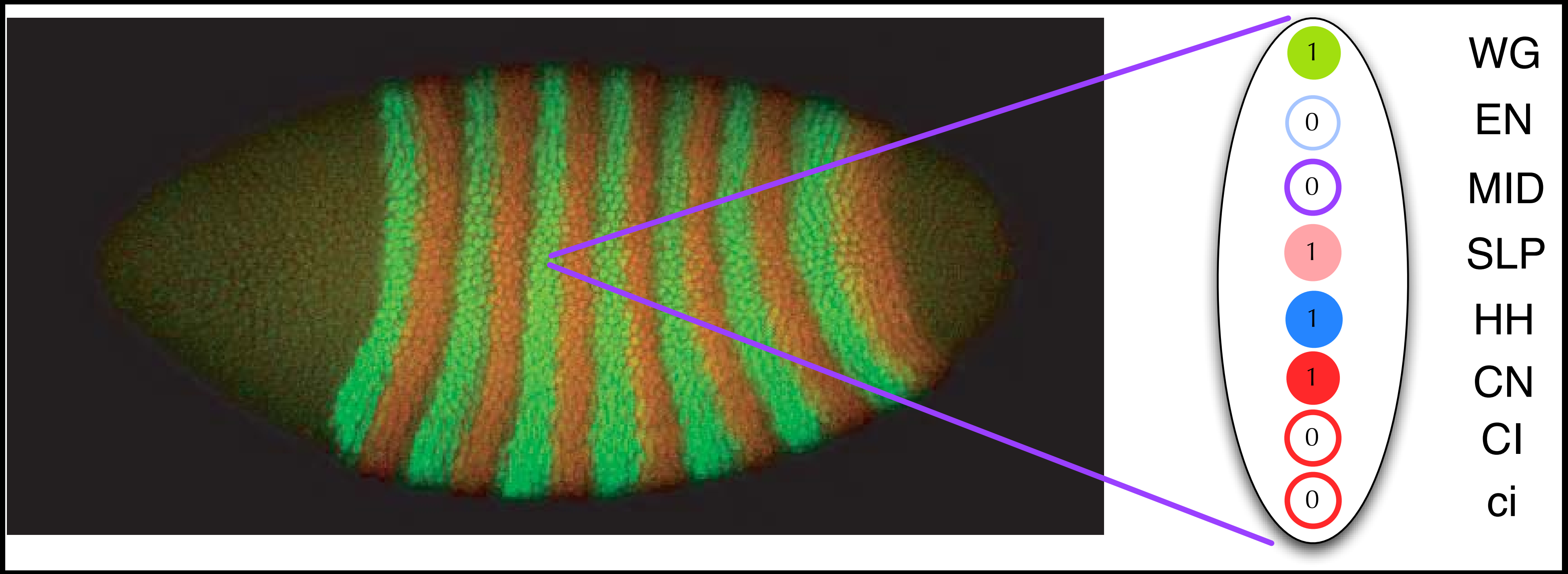


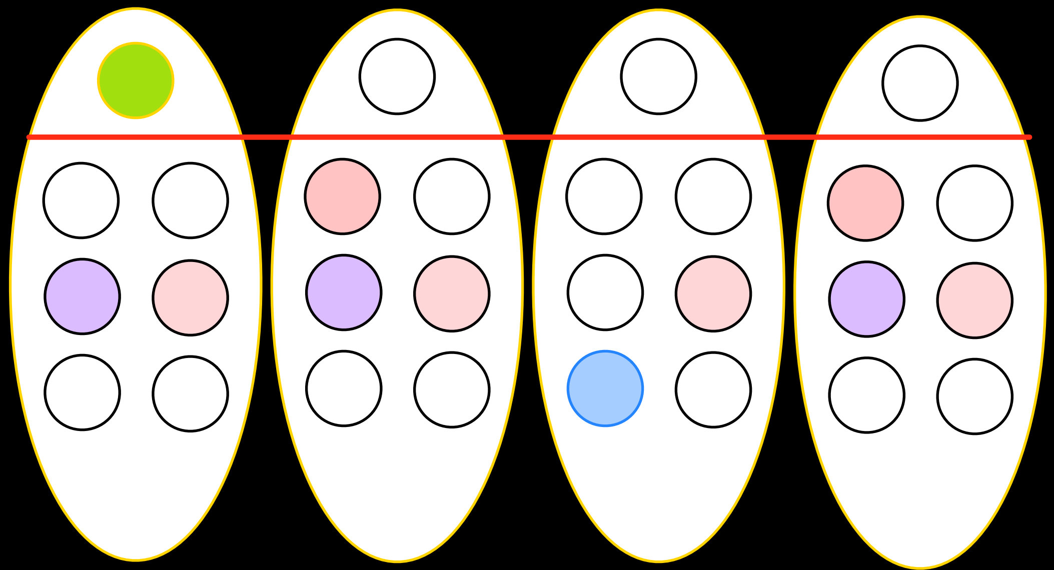
Structured “multi-register” internal states.

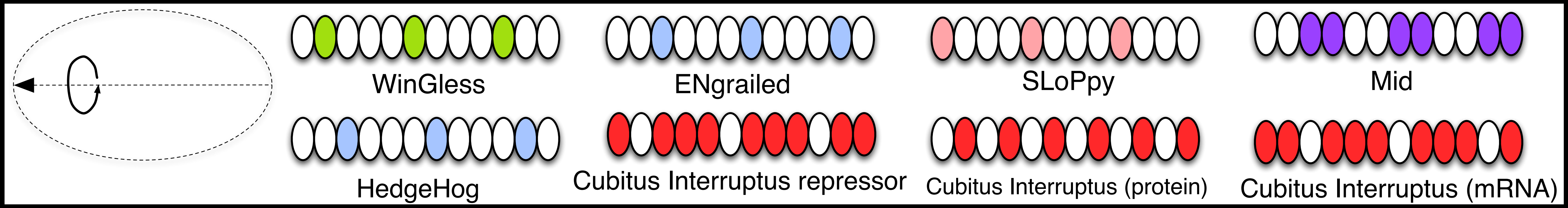


Multislot dynamics

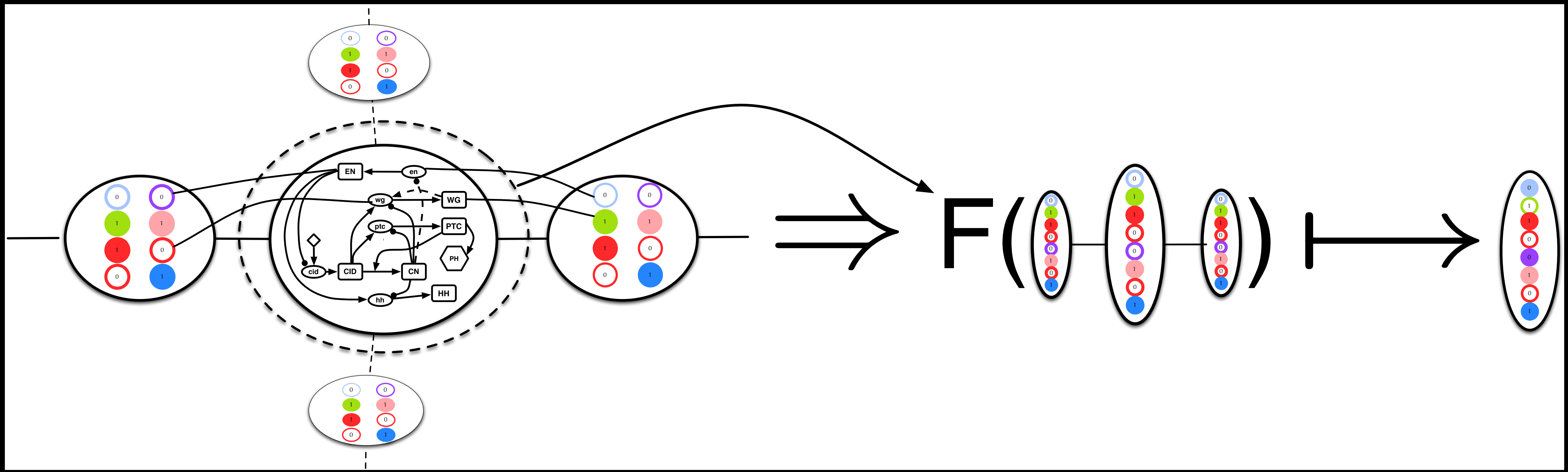
multi-agent model







multi-agent model



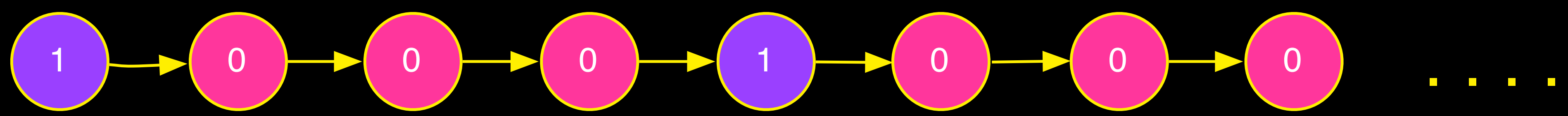
the gene regulatory network
as a nearest-neighbor local rule

the "brains" of the cellular-agent

An informational bound

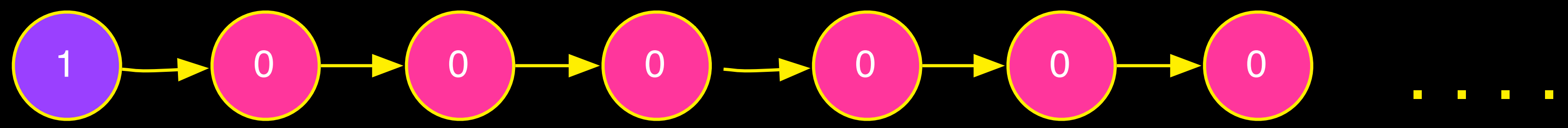
Local Checkability

Consider the repeat pattern:



Can this pattern be solved robustly with a nearest-neighbor rule?

Answer: No.   Because the with a radius 1 rule, 000 would have to be a fixed state.

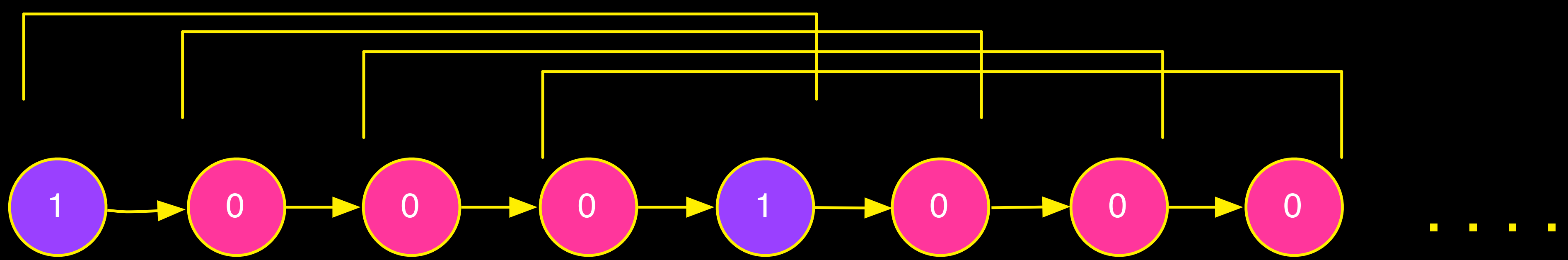


local checkability; multi-agent reasoning about local equilibria

A function $\Theta : B_r \rightarrow \{0, 1\}$ is a *local check scheme* for a pattern T if

$$\bigwedge_{i \leq |X|} (\Theta(B_r(i, X)) = 1) \quad \Rightarrow \quad X \in T$$

This means: if all cells think "criterion Theta" is true ... then actually the organismal pattern is correct



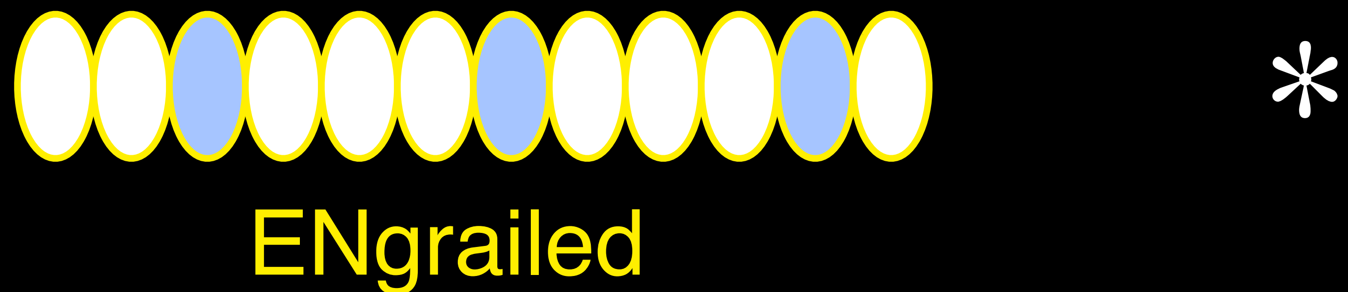
1000 has a radius-2 check scheme.

All repeated patterns are locally checkable.

Any pattern thus has a *minimal* check radius, a lower bound on how far local rules have to see to form the pattern robustly.

local checkability; multi-agent reasoning about local equilibria

Hey. Wait.



Local check radius of this pattern is = 2, and in fact to actually make a rule radius 5/2 is required.

But the coupled gene network rules are nearest neighbor (i.e. r = 1).

Put another way, no equation of the form:

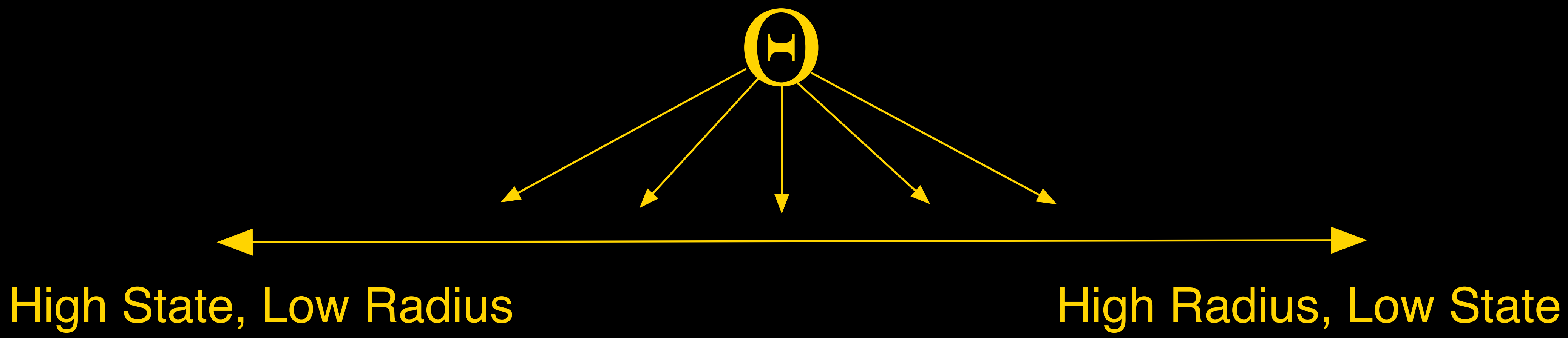
$$\frac{dS_j^i}{dt} = f_j\left(S_1^i, S_2^i, \ldots, S_N^i, S_1^{i-1}, S_2^{i-1}, \ldots, S_N^{i-1}, S_1^{i+1}, S_2^{i+1}, \ldots S_N^{i+1}\right)$$

can have a nondegenerate stable steady state of the form * if engrailed is decoupled.

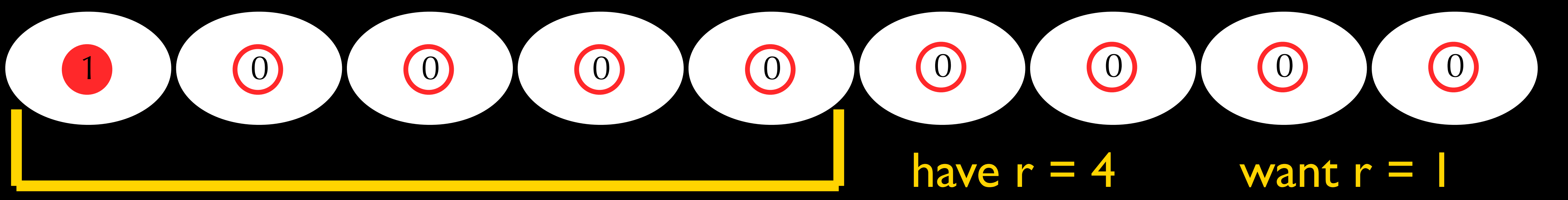
The system is "trying" to make a 4-coordinate....

The Radius/State Tradeoff

The Radius/State Tradeoff



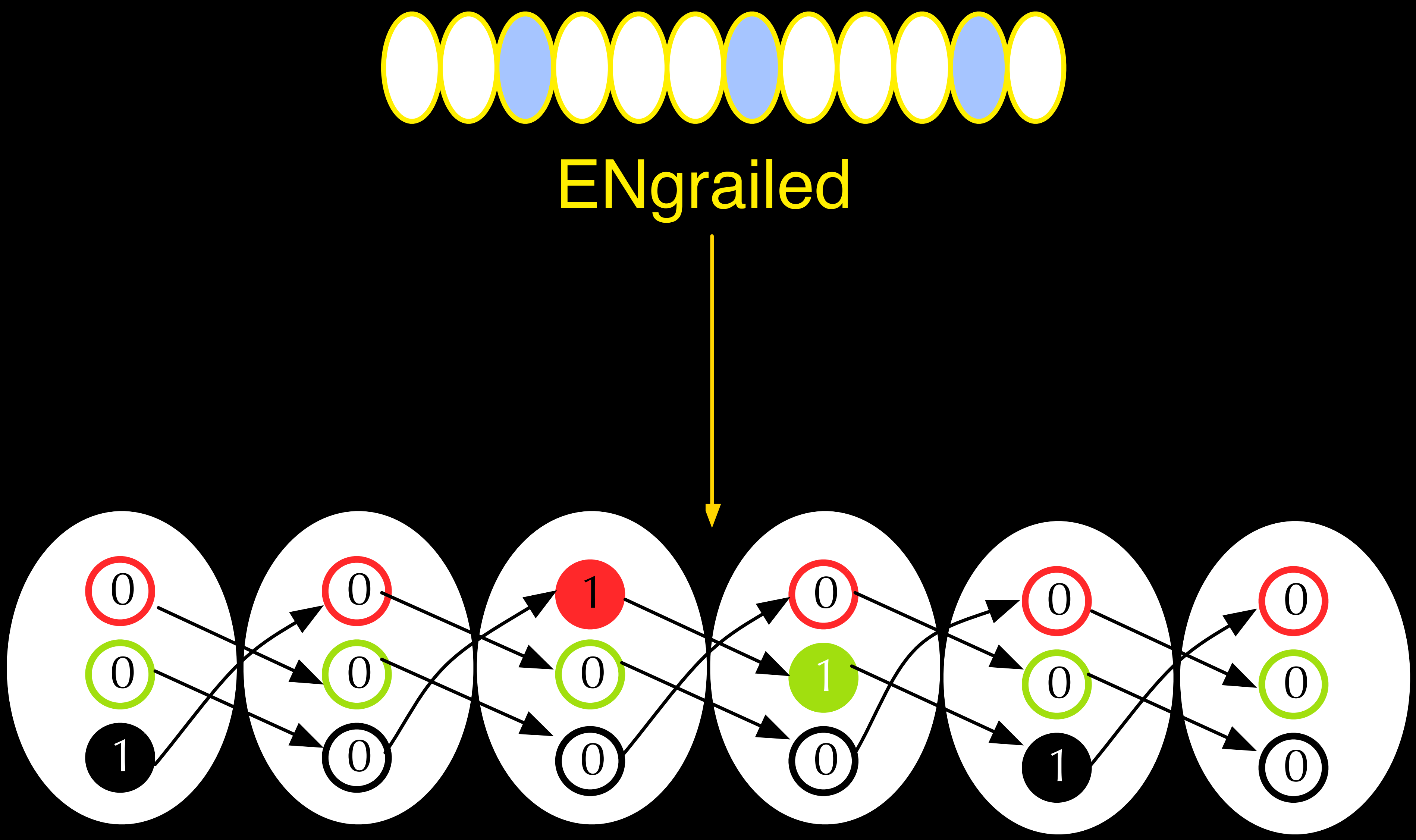
A generic "cut-and-shift" procedure for implementing the Radius -> tradeoff:



#slots = $\lceil (2r+1)/(2k+1) \rceil$ to go from radius r to radius k.

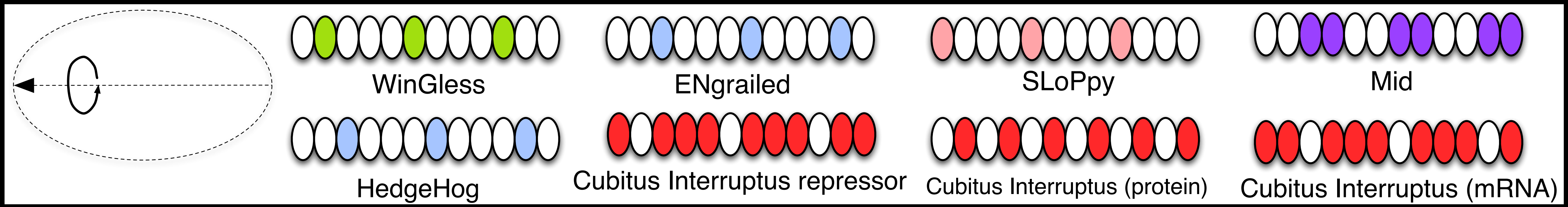
The Radius/State Tradeoff

Let's apply radius/state tradeoff "cut-and-shift" procedure.

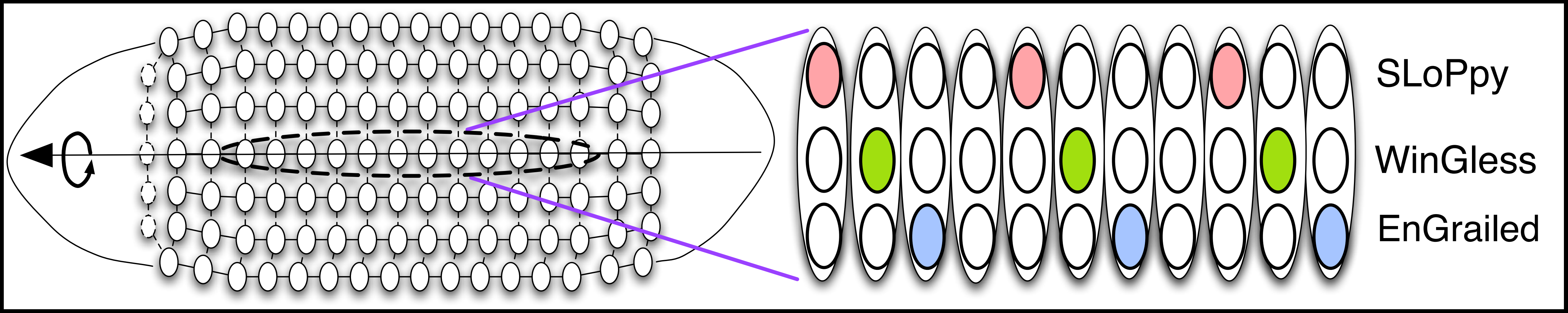


To get a make the trade, the algorithm adds two extra "slots."

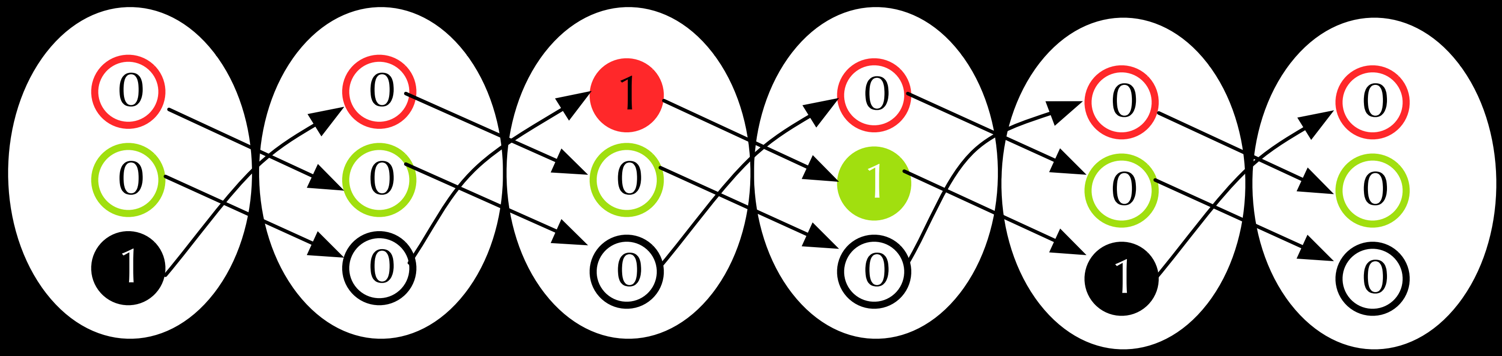
The Radius/State Tradeoff



Focus on the three proteins: Sloppy, Wingless, Engrailed.

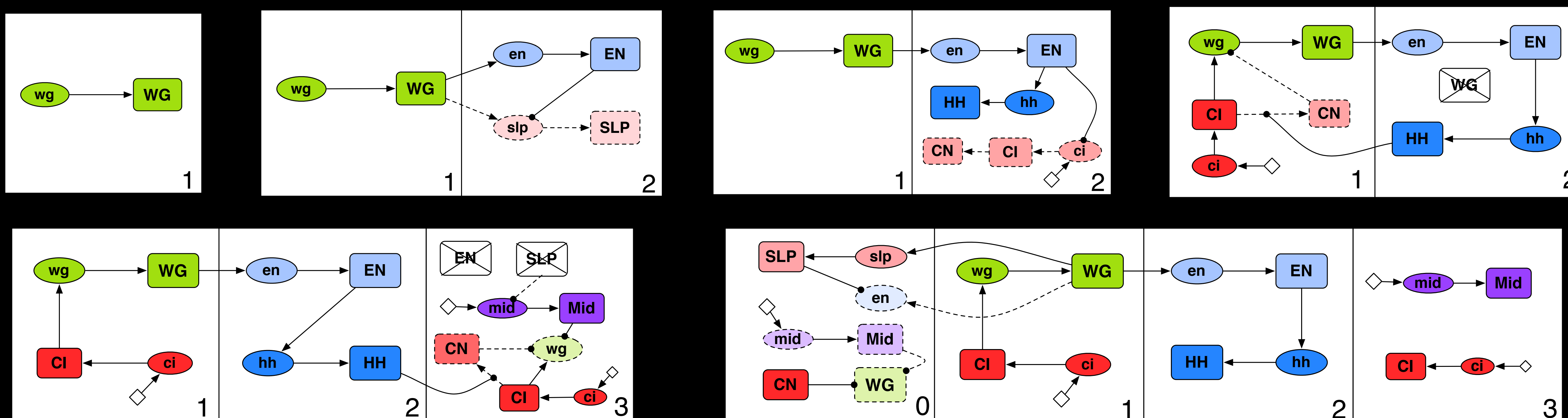


Interesting superficial resemblance:

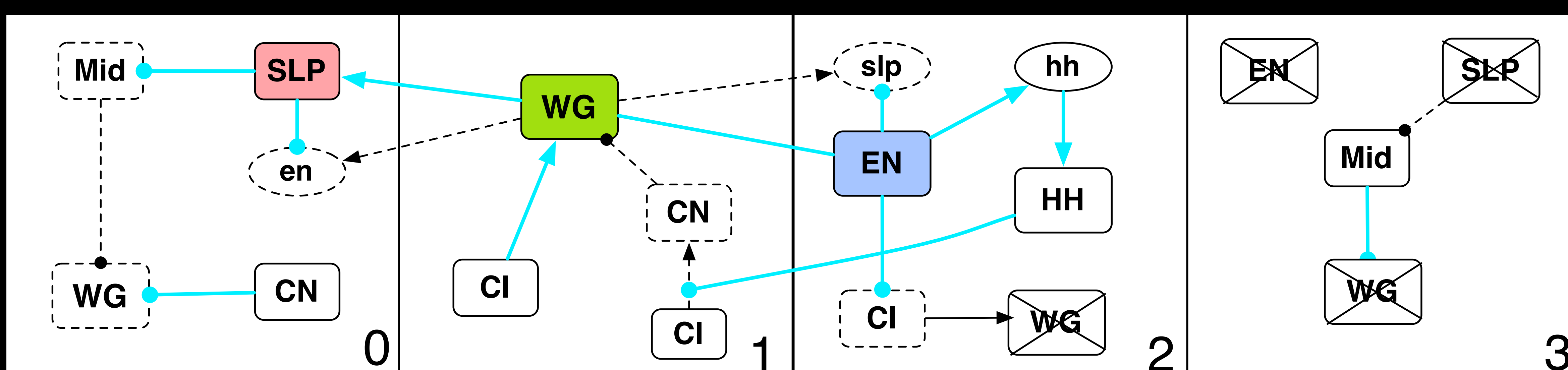


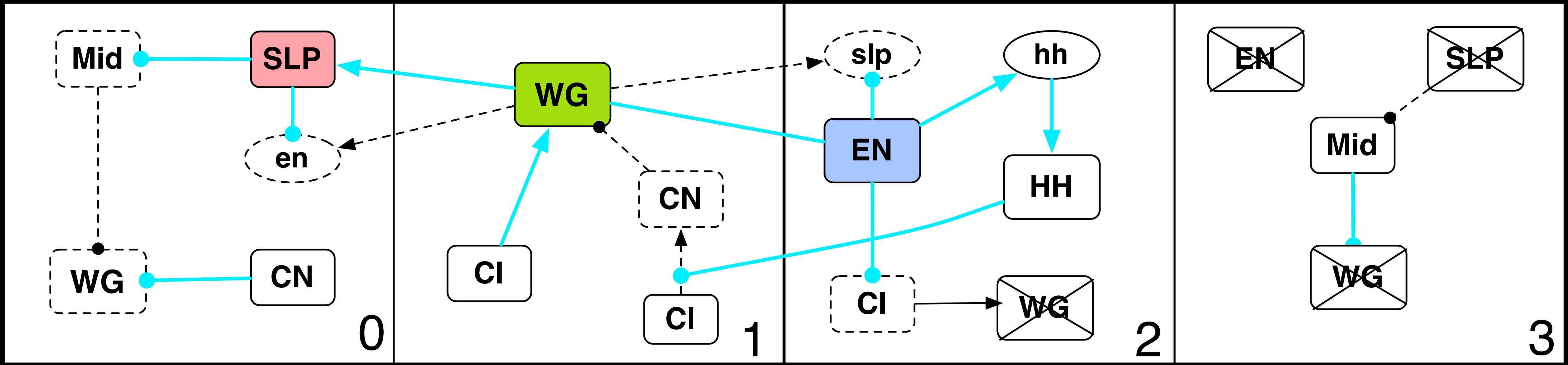
The Radius/State Tradeoff

Much stronger form of validation comes by isolating a "path" within the gene regulatory network that stabilizes the three proteins in a spatial feedback sequence:

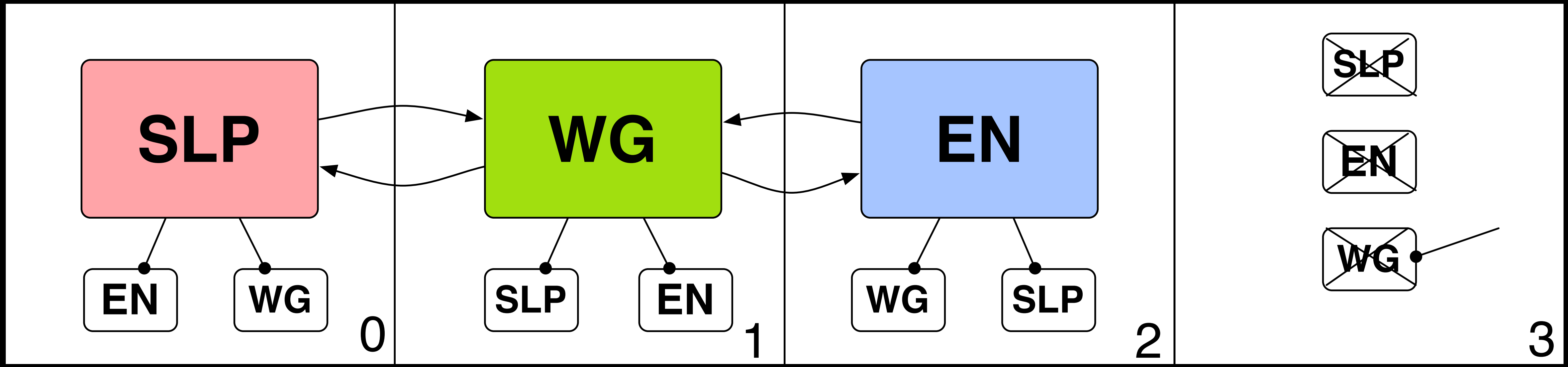


The dynamically activated subnetworks:





Consolidating the intermediates:



Conclusion: the coordination of multiple input-output state relations that unfold over space and time allow cellular agents with nearest-neighbor views to generate long-range coordinate patterns.

Experimental: look in related drosophilids for modifications (DePace lab)

Beyond
Drosophila

(kinda, sorta, *maybe*)

- What does thinking of natural spatial multi-agent systems *qua* computers tell us scientically?

It tells about local information requres, helping understand:

- What agent resource capacities are required to solve a given task.

But the things we learned (i.e. cut and shift) were general.

- Can we then turn around this knowledge to build a better bug?

so that we could, for example, make general patterns?

In theory.

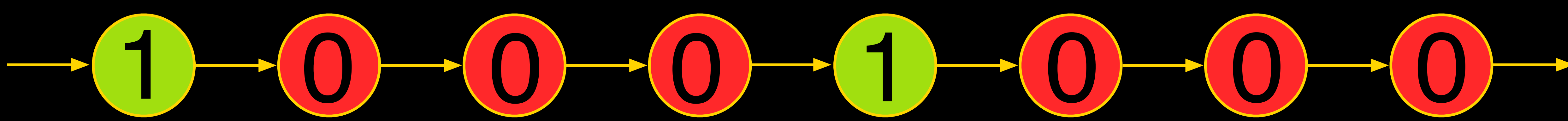
Designing Local Rules

Goal: Design local rules to make given patterns.

Repeat patterns have a <u>well-defined gradient</u>.

$$\nabla_{\Theta}(B)^{+} = \begin{cases} i, & \text{if } B \circ i \text{ is consistent with } \Theta \\ B(2R), & \text{otherwise} \end{cases}$$

For example: 1000-repeat pattern (which has an r = 2 check scheme),



$\nabla_{\Theta}(100)^{+} = 0$

$\nabla_{\Theta}(000)^{+} = 1$

$\nabla_{\Theta}(001)^{+} = 0$

$\nabla_{\Theta}(010)^{+} = 0$

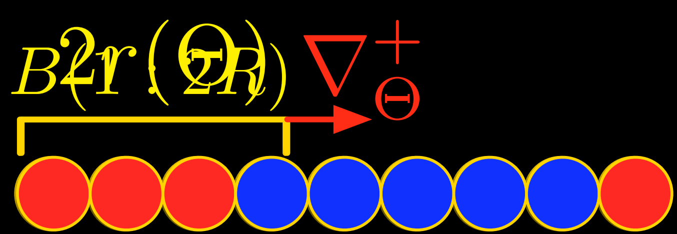
Otherwise,

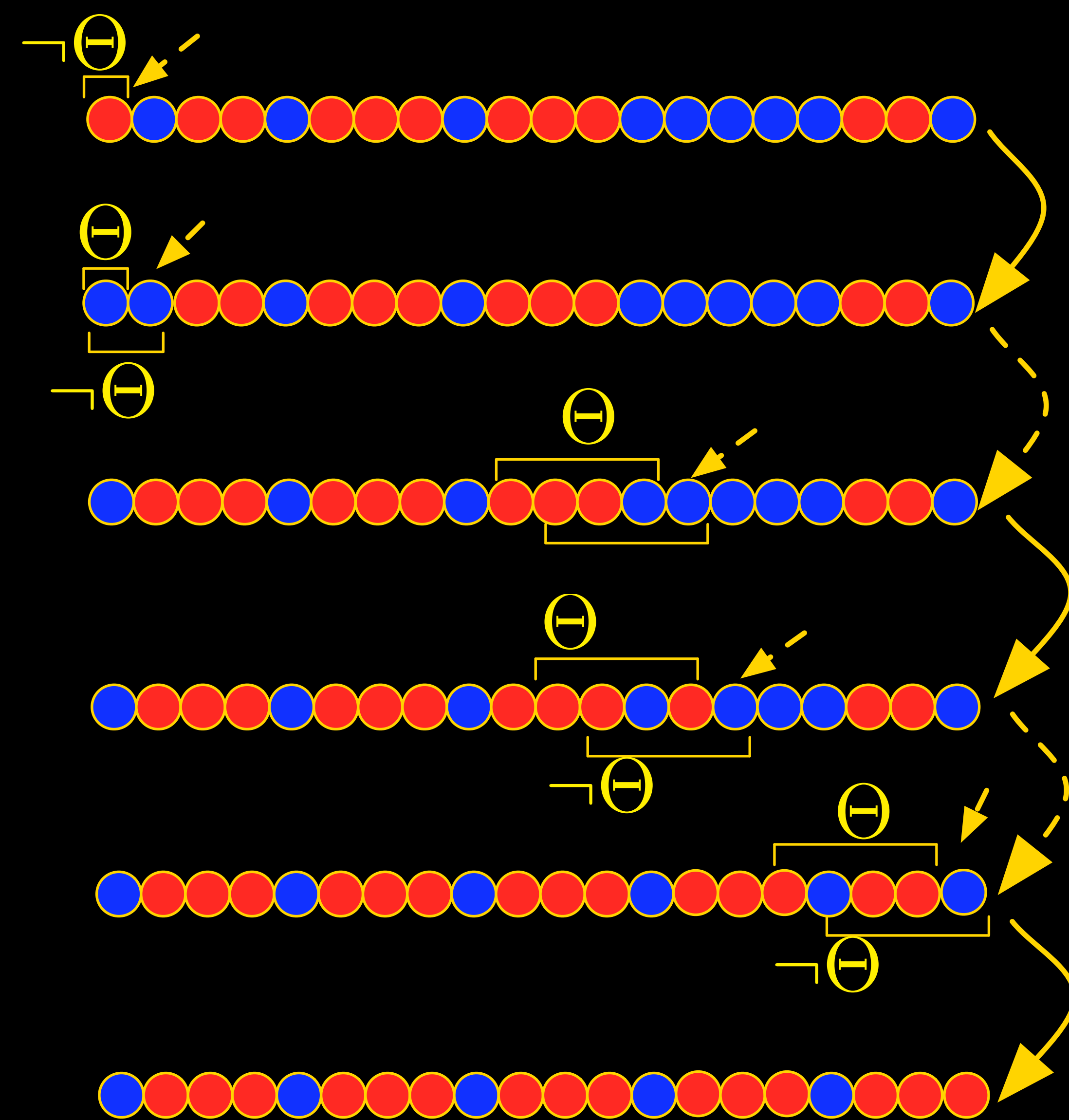
$\nabla_{\Theta}(b) = b(3)$

Designing Local Rules

Now simply define:

$$F(B) = \nabla^{+}_{\Theta}(B(1:2r))$$



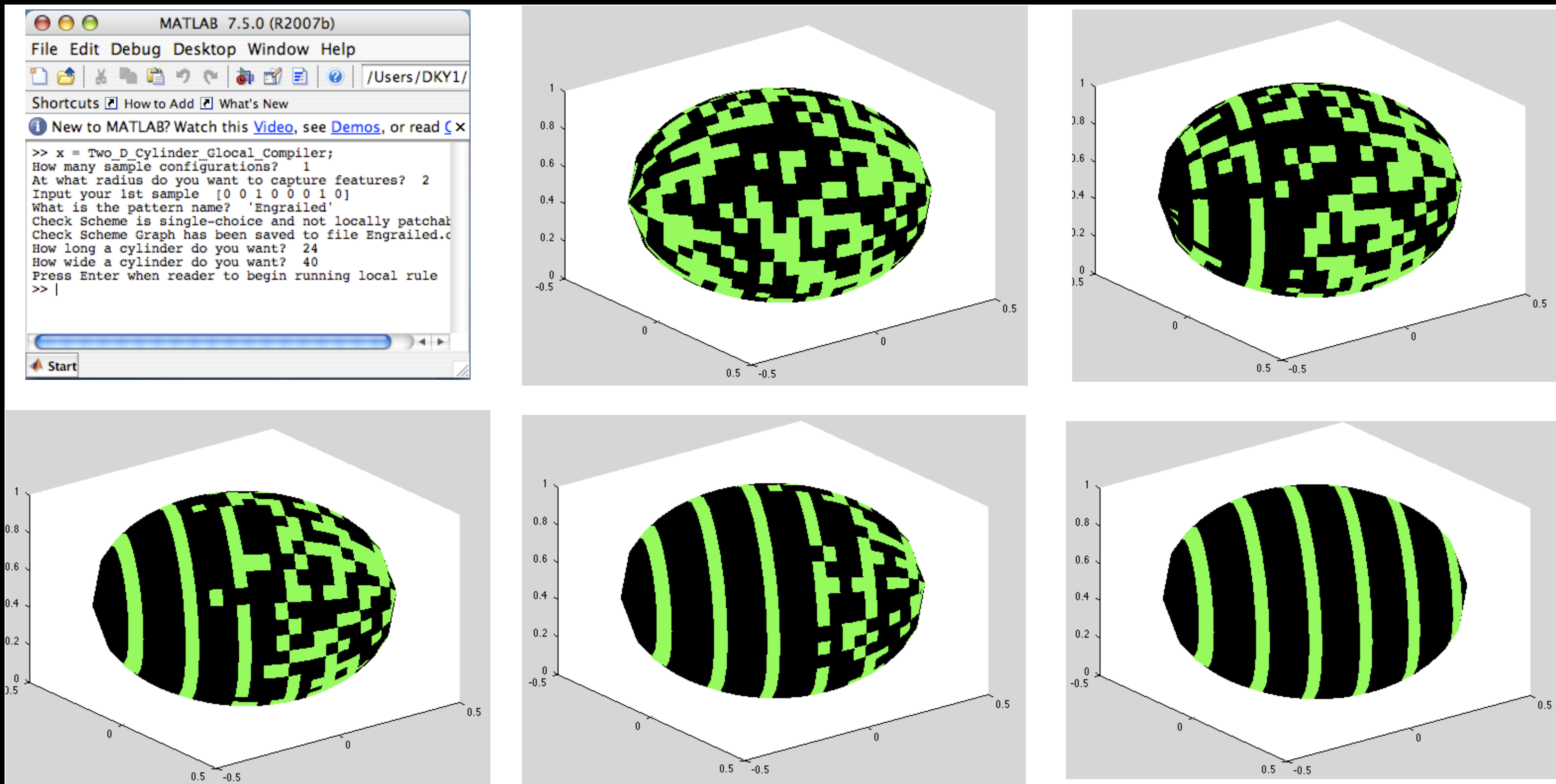


Gradient "spreads correctness."

Gradient waves ... Multiple "waves" all at once in actuality ....

Not hard to implement in higher dimensions.

Designing Local Rules



Resulting rule is 100% robust to:

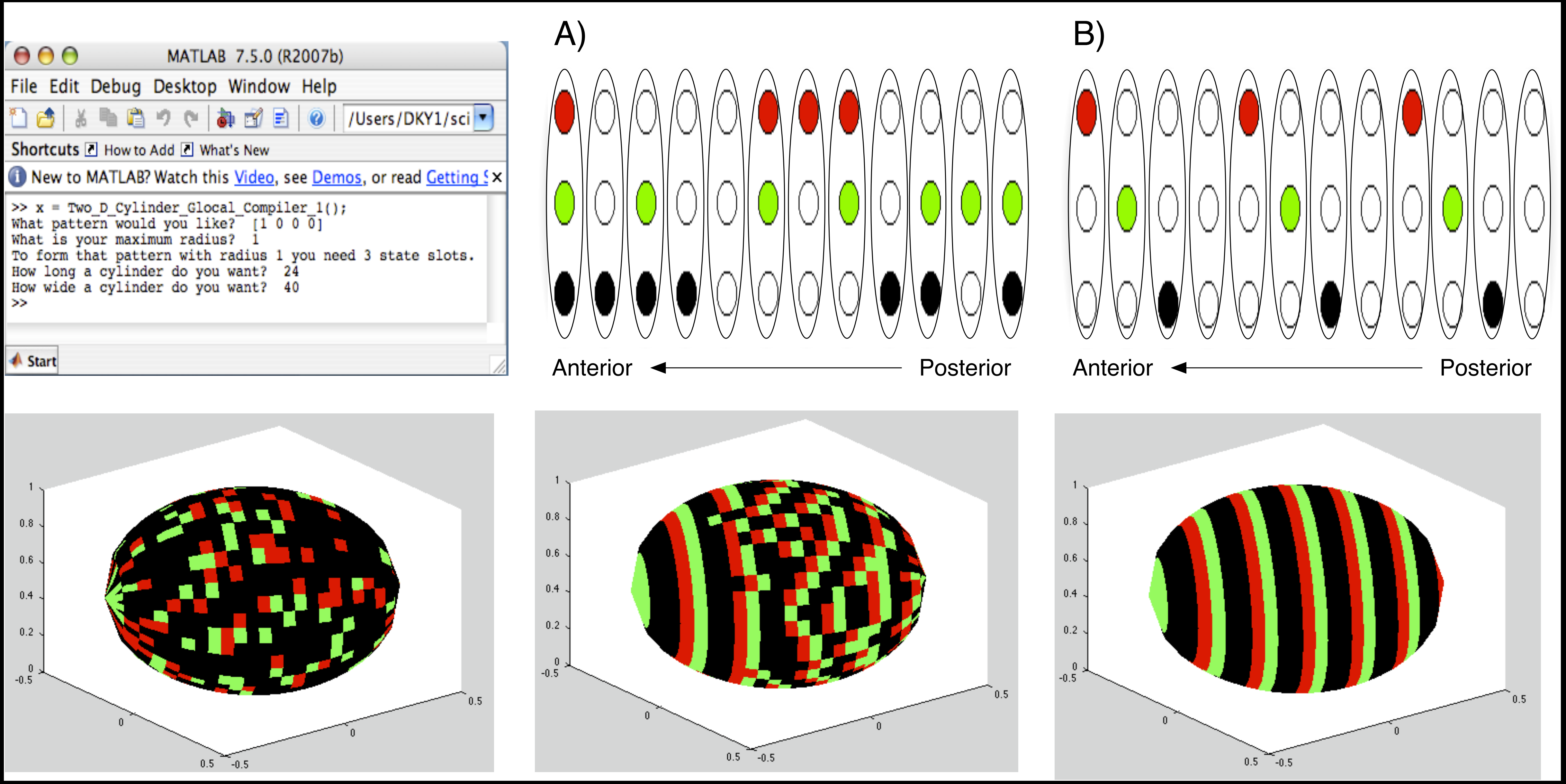
- Initial condition perturbations.
- Timing issues.

Designing Local Rules

But: the rule to solve a repeat pattern of size |q| takes a radius of |q|.

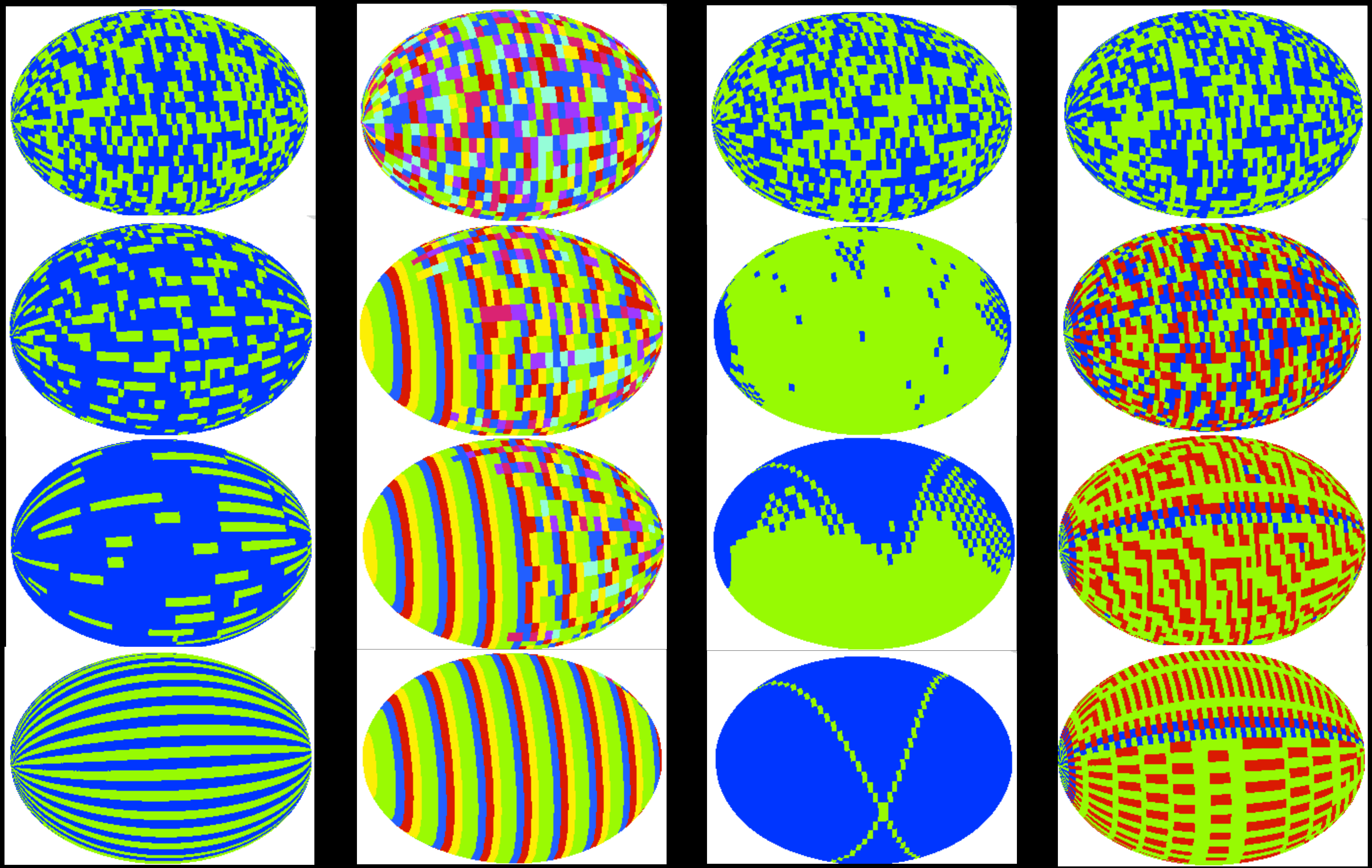
Can run a radius-minimization algorithm.

And then, couple in the radius/state tradeoff.



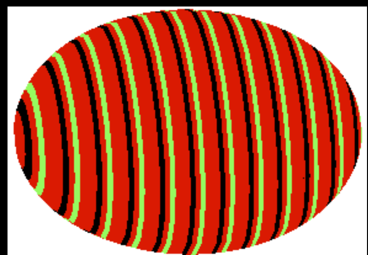
Designing Local Rules

Similar principles can be applied to develop a language for building a diverse array of patterns.



Designing Local Rules

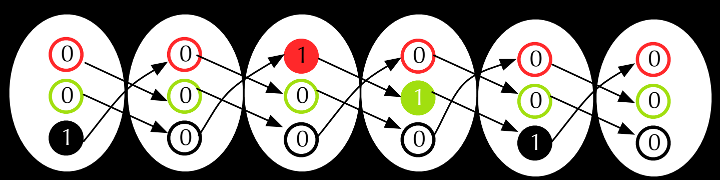
Pattern



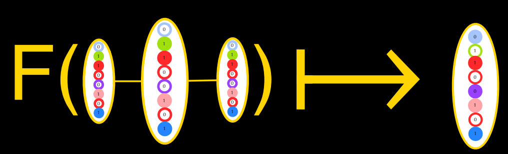
→ Local representation Θ

→ Minimize radius Θ^{min}

→ Tradeoff



→ Dynamic Rule



→ Network

???

Designing Local Rules

- Some VERY simple ideas in multi-cellular synthetic biology *might* be ``understandable'' (or at any rate, imagineable) and connectable to cellular programs

- Is anything here doable? even remotely?

- What is the simplest system that could possibly exploit some of these ideas?

- What I'd really like to explore is .... evolution.