

Gary Overgard  
NNG

'Making Molehills out of Mountains'

*"The fact is that computation is known to be a thing of surpassing complexity generated from a simple set of rules. That's why we do it."*

-Dr Alvy Ray Smith

# An experience report

--original goal:

eliminate all manager type code  
decouple as much as possible

--viewpoint of experimenter/practioner

--methods of dealing with large network problems

--the domain

# What to keep in mind...

Agents

Ruleforms

Animation engines

Hox Software genes

and...

# Continuations

basic idea: allow code execution deferment

Some examples:

**Seaside**-- save state of current transaction, but code in typical manner

**Swarm**-- pass code to execute around to data objects scattered over web

**Hox SW genes**-- code in a manner to allow code deferment  
across many objects

# What to keep in mind...

## Agents

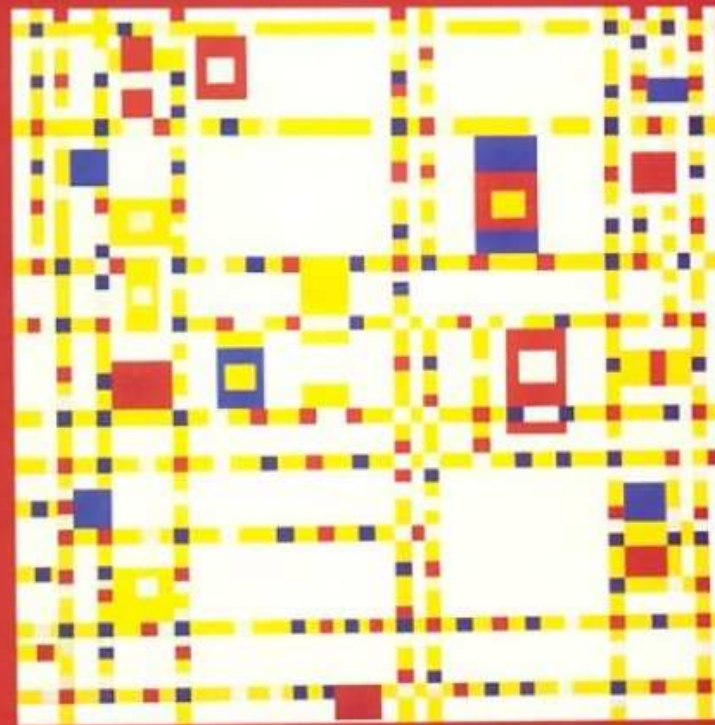
Ruleforms

Animation engines

Hox Software genes

GROWING  
ARTIFICIAL  
SOCIETIES

SOCIAL SCIENCE FROM THE BOTTOM UP



JOSHUA M. EPSTEIN  
ROBERT AXTELL

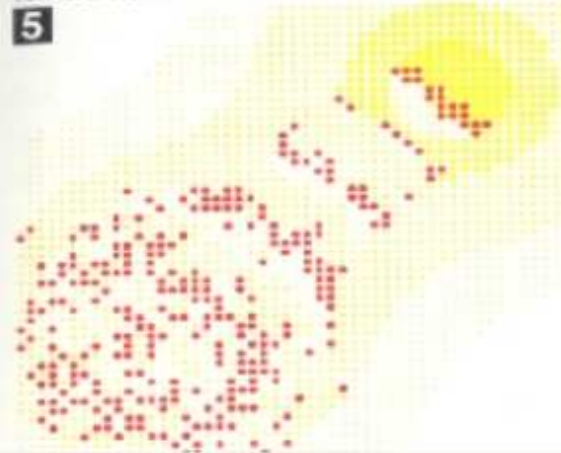
1



3



5



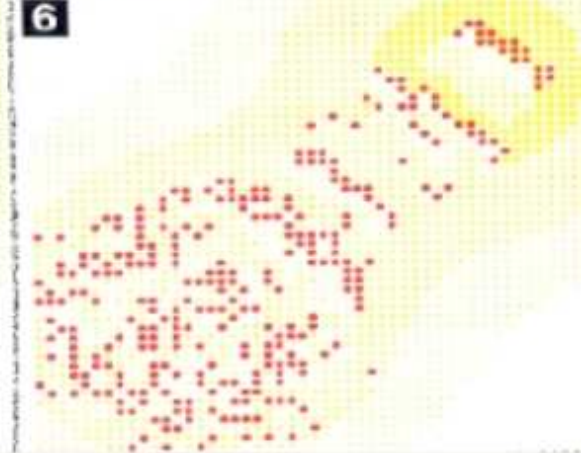
2



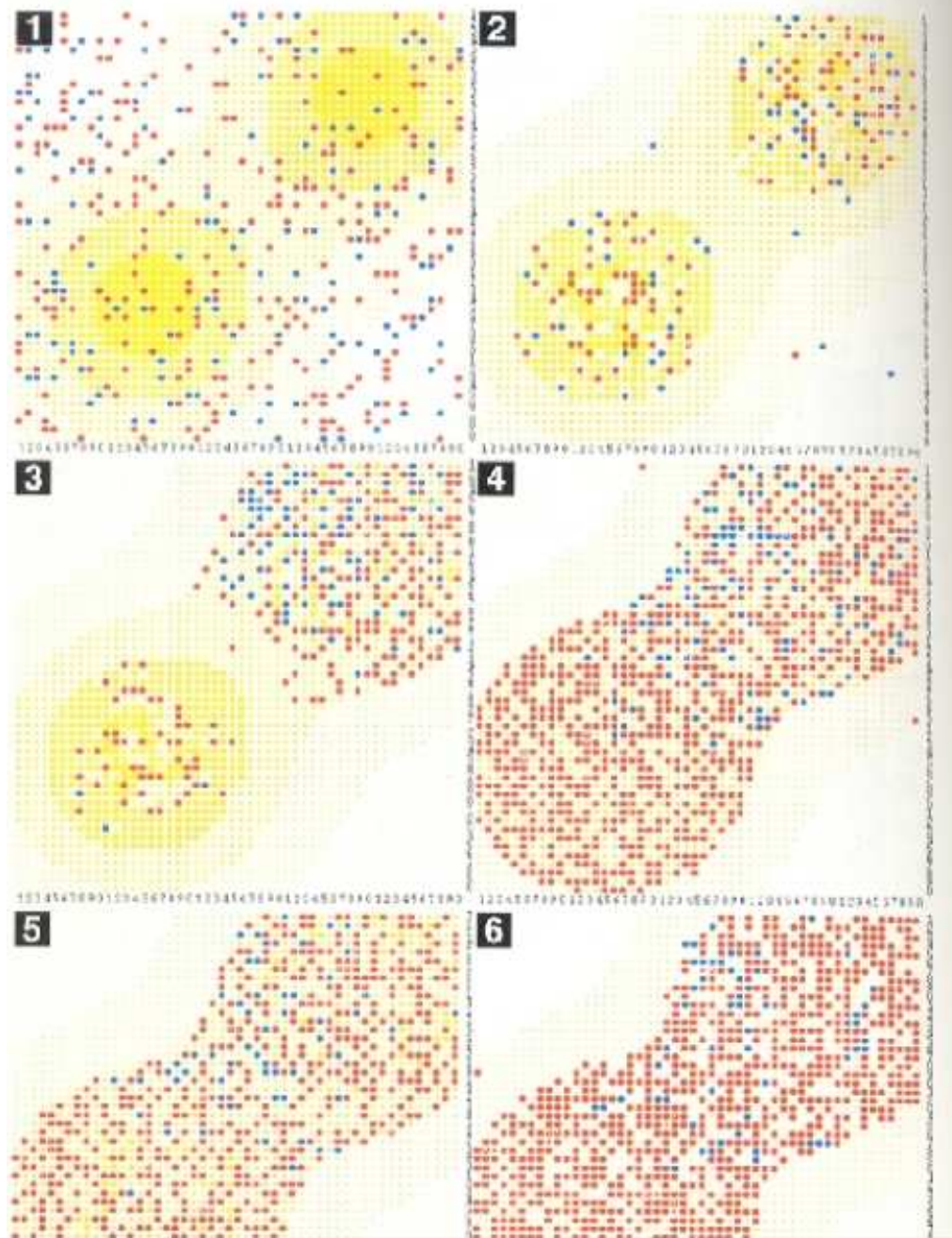
4



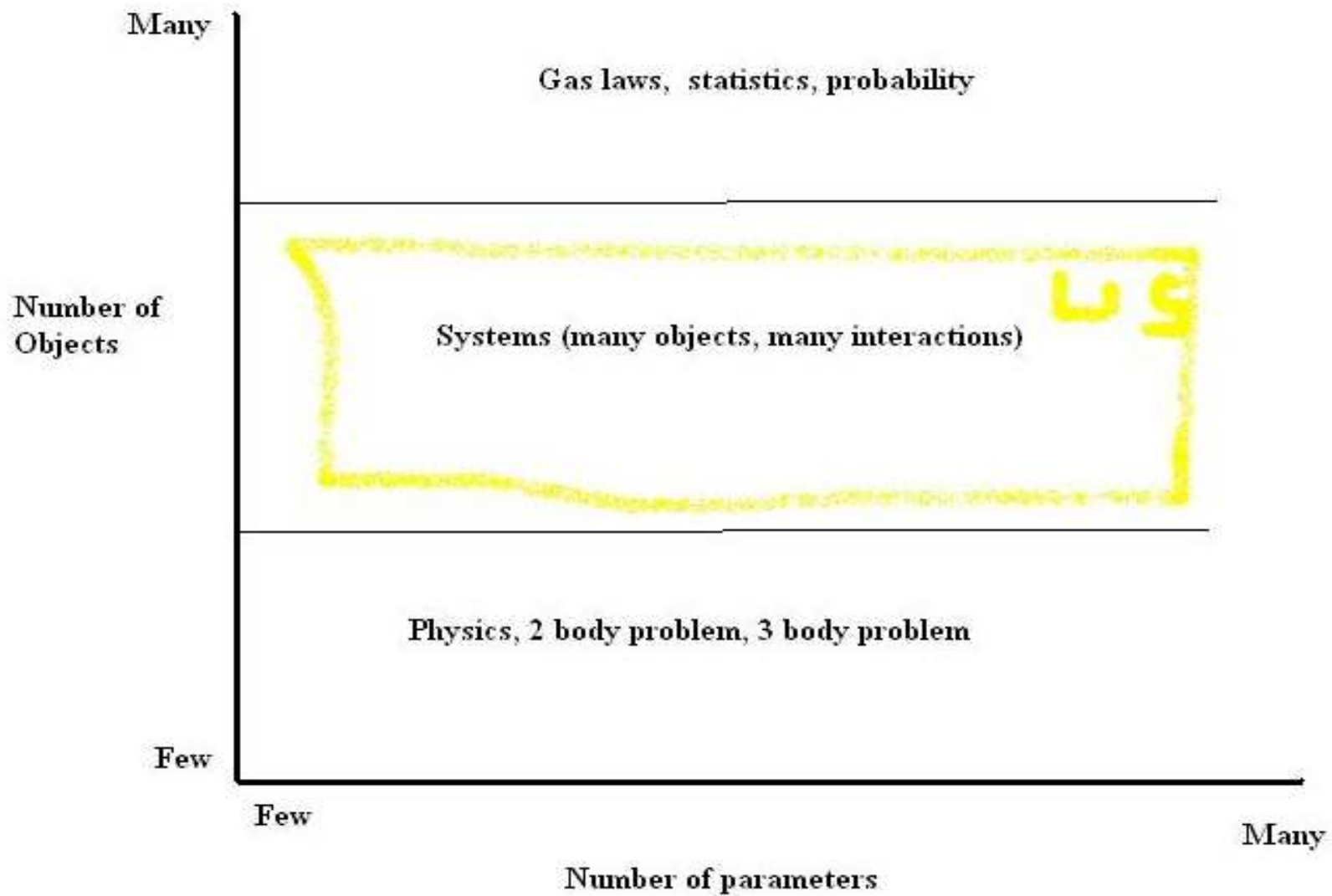
6



Animation III-2. Societal Evolution under Rules  $\{(G_1), \{M, S\}\}$ ,  
Coloring by Agent Vision



# Why this is cool...



# *But, Issues in the business world*

--need deterministic results

--not obvious how to use

--we expect our community of objects to behave in very constrained ways

# What to keep in mind

Agents

## Ruleforms

Animation engines

Hox Software genes

# Ruleforms

Precursor to theories of emergence

idea of deep structure

generate surface layer (process) by execution of simple rules

within a domain/context there exists deep forms that are invariant

## Ruleforms

If A and B and C , then consider X, Y

|•-----if portion -----•||•-----then portion-----•|

<b>Camouflage</b>	<b>Spotted</b>	<b>Cover</b>	<b>Action</b>	<b>UseCover</b>
Good	#(hp lp false)	Good	Look	True
#(ok bad)	#(hp lp false)	----	Duck	True
----	True	Good	Flight	True
----	True	#(ok bad)	MaxFlight	False
----	----	----	Look	True

# Brian Foote, Shearing layers



different artifacts change at different rates

# Ruleform sources

*The Ruleform Hypothesis: Complex system structures and behaviors are generated by not necessarily complex processes; these processes are generated by the animation of operating rules. Operating rules can be grouped into a small number of classes, whose form is prescribed by Ruleforms. - UltraStructure ACM 1/ 95, Jeffrey Long*

*other:*

*<http://www.cincomsmalltalk.com/userblogs/cincom/digest?content=2001-files-ruleforms>*

# What to keep in mind

Agents

Ruleforms

**Animation engines**

Hox Software genes

# Entanglement, Chemical soups & more

Wiki: quantum computing--constituting objects are linked together so that one object can no longer be adequately described without full mention of its counterpart — even though the individual objects may be [spatially separated](#). This interconnection leads to non-classical [correlations](#) between observable [physical properties](#) of remote [systems](#)

Chemical Soups: many enzymes acting as cpu's for agents  
i.e. problem with a single processor & many objects that have requests at same time

# Brownian motion

Issue: how to give all the agents the necessary jiggling they need

Simple solution: simple loops with termination condition that checks to see when Agents are done

# Things to keep in mind...

Agents

Ruleforms

Animation engines

Hox Software genes

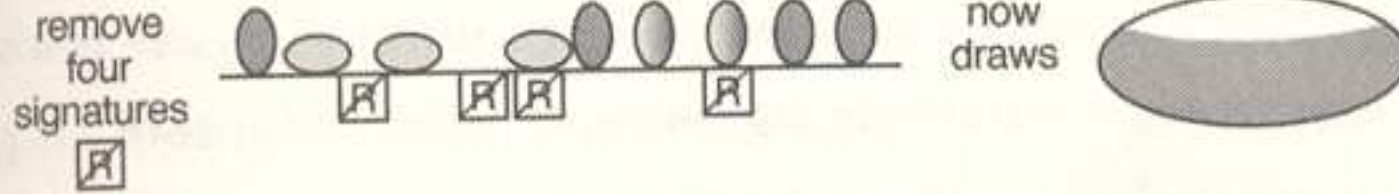
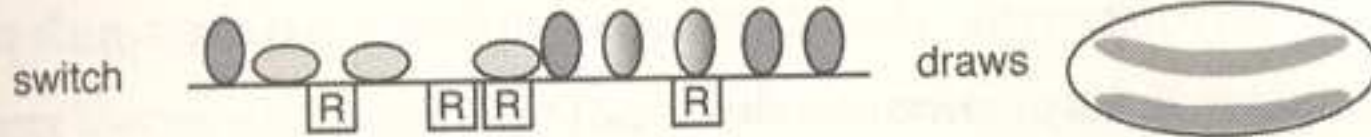
Copyrighted Material

THE NEW SCIENCE  
OF EVO DEVO

ENDLESS FORMS  
MOST BEAUTIFUL

SEAN B. CARROLL

Removing signatures for a repressor  
(symbols appear above and below when overlapping)



**Hox genes** are a group of related [genes](#) that specify the anterior-posterior axis and [segment identity](#)

The regulators of a given operon, including [repressors](#), [corepressors](#), and [activators](#), are not necessarily coded for by that operon

*But, there is a simpler way to think of it*

Hox SW genes based on their current environment either block or allow signals to be processed

--Back to idea of community, sometimes actions are appropriate, other times they are not

--Pecking order

# How does this all come together?

Hox genes are used in conjunction with Agents to control when & what they can do (repress or permit behaviors)

Ruleforms aggregate all the desired behaviors (control logic) into a single form

Animation Engine makes it behave as if it were a chemical soup with many chances at action

## Local Hox

### --Path Binding

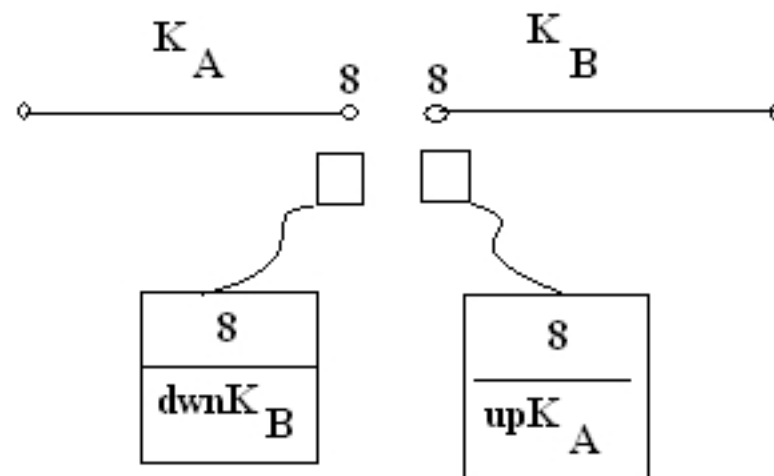
attributes: point number

contract number

upstream contract number

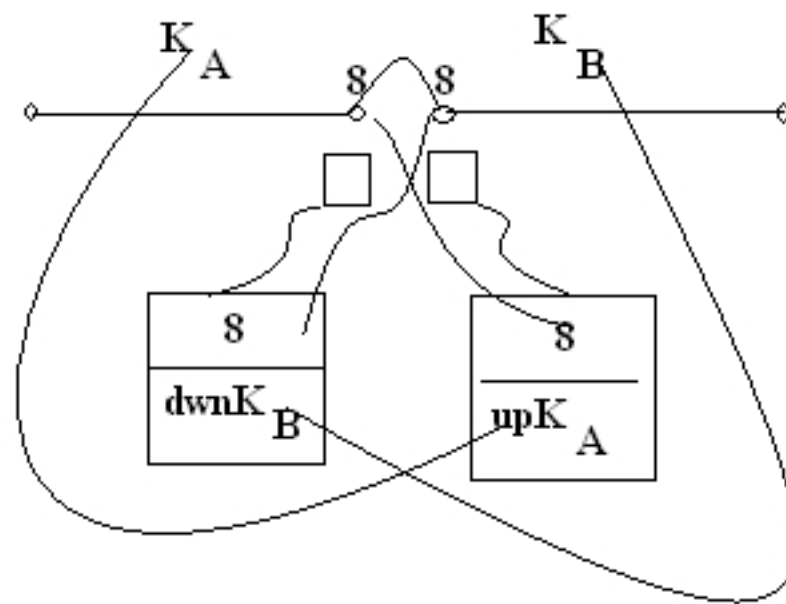
downstream contract number

example: bind the following 2 paths



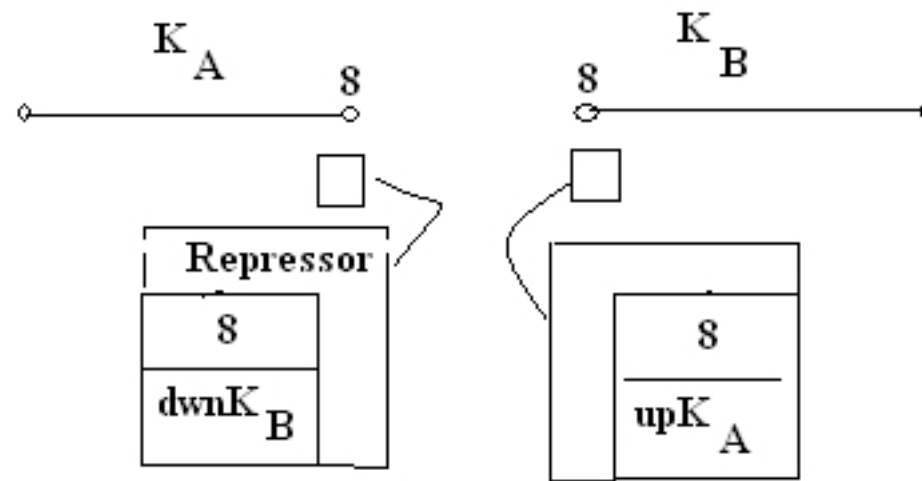
**Local Hox**

**--2 paths bound**



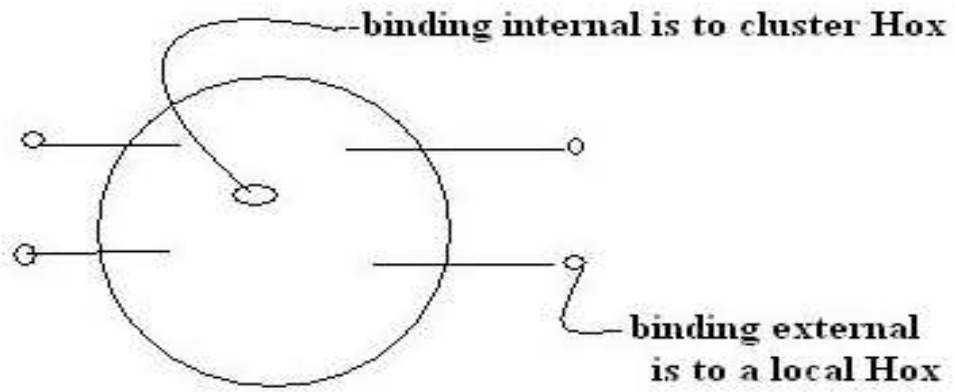
### Local Hox with Repressor

--Assume contract A has expired

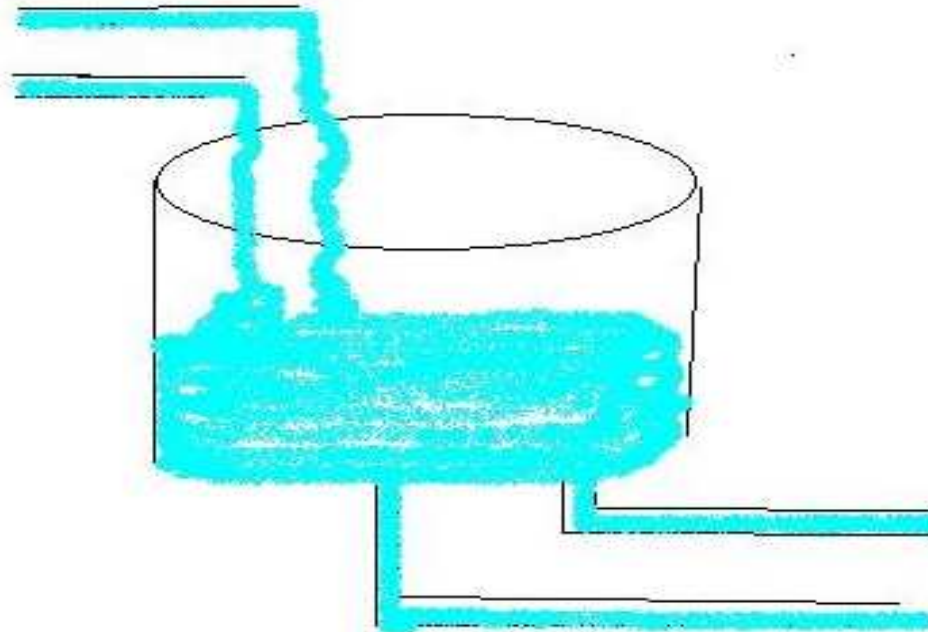


### Clustered Hox

--hox displaying a binding from a local collection

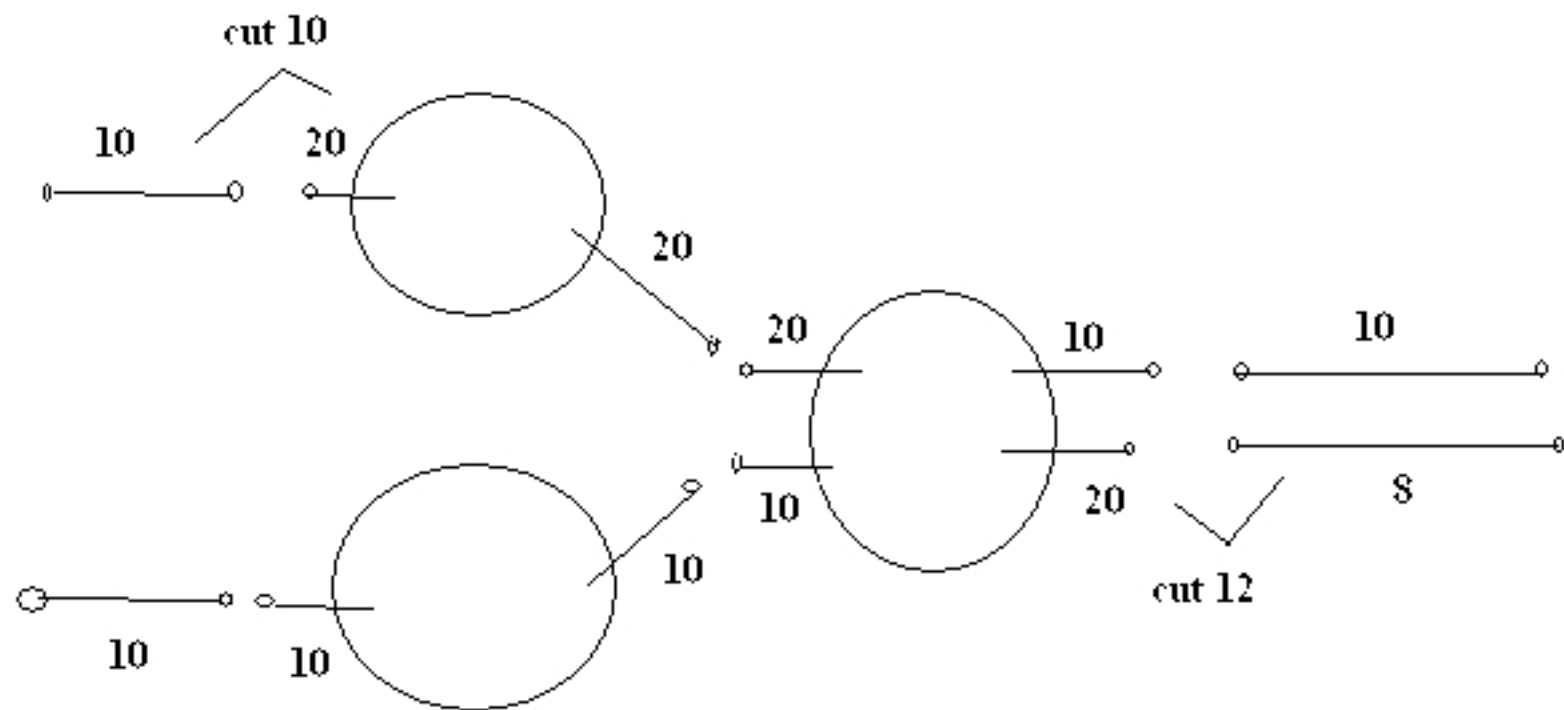


Pool -- aggregates gas. --think swimming pool



## Clustered Hox

-- example how this would be used with a repressor

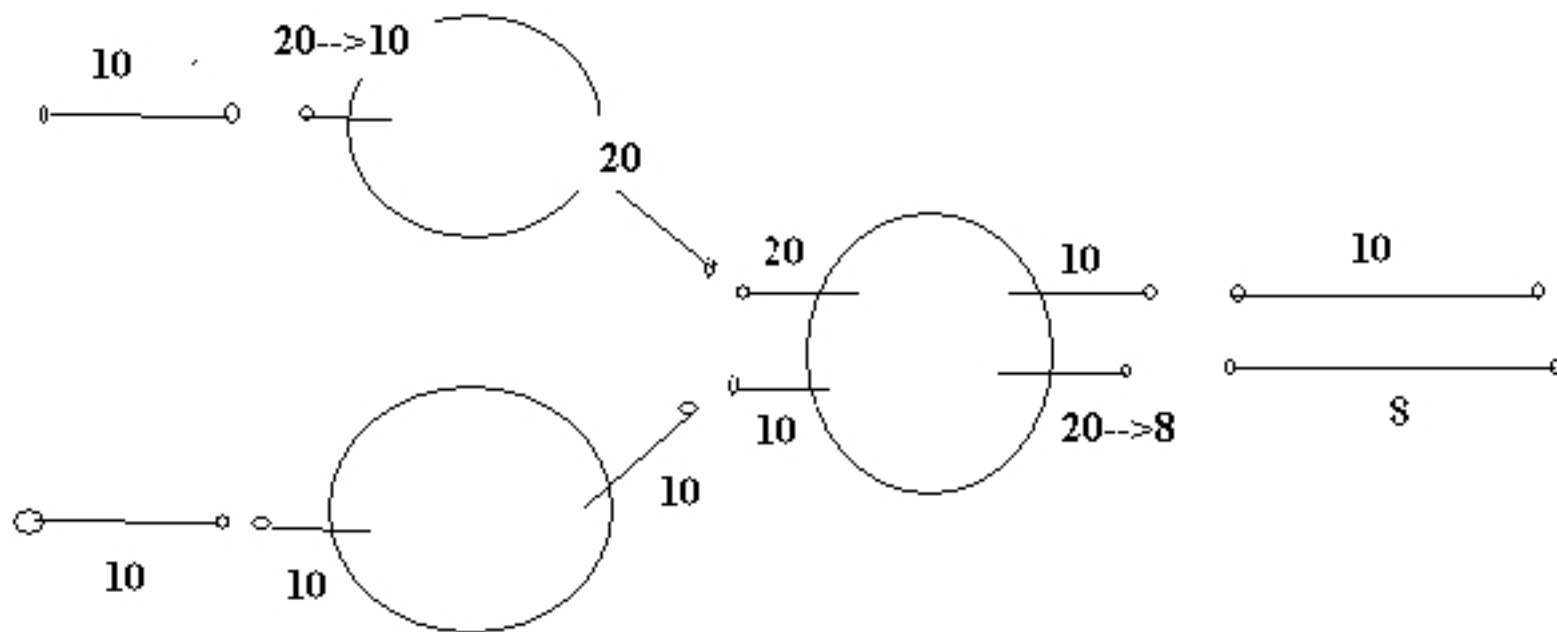


### Clustered Hox

-- example how this would be used with a repressor

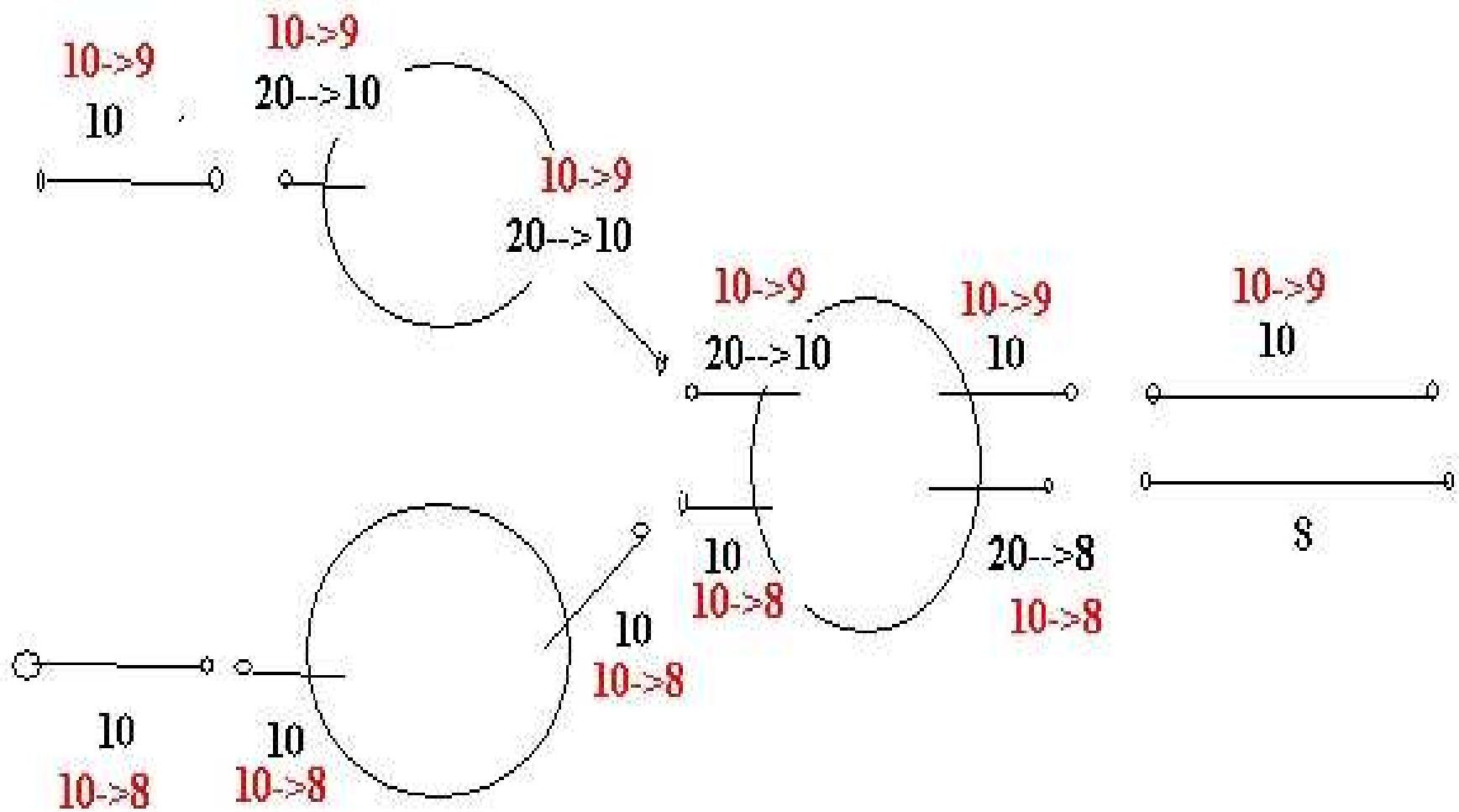
--repressor insures no balancing through pool until path cuts

Step 1



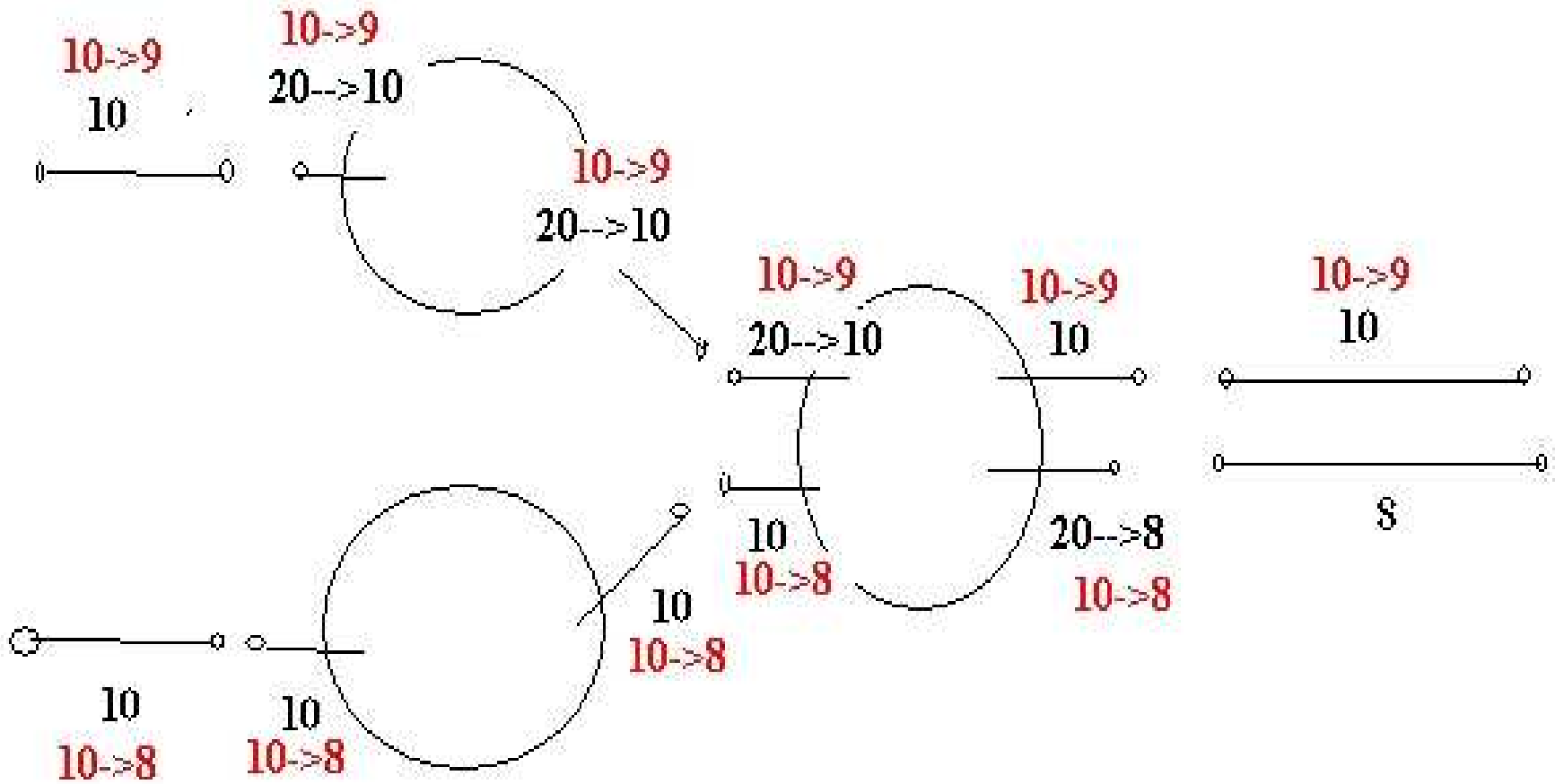
Step 2 balance through pools

total volume moved: 17



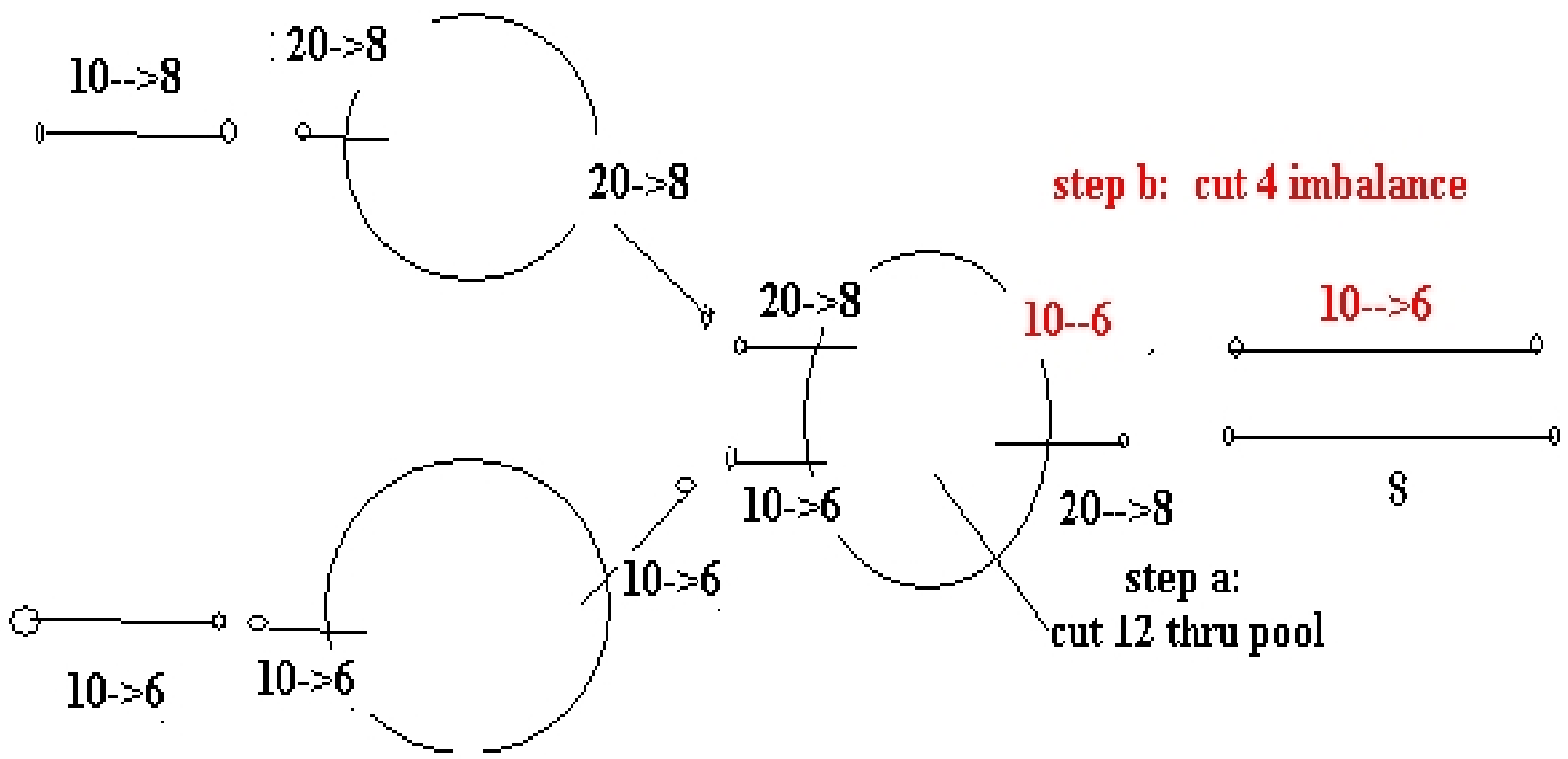
Step 2 balance through pools

total volume moved: 17



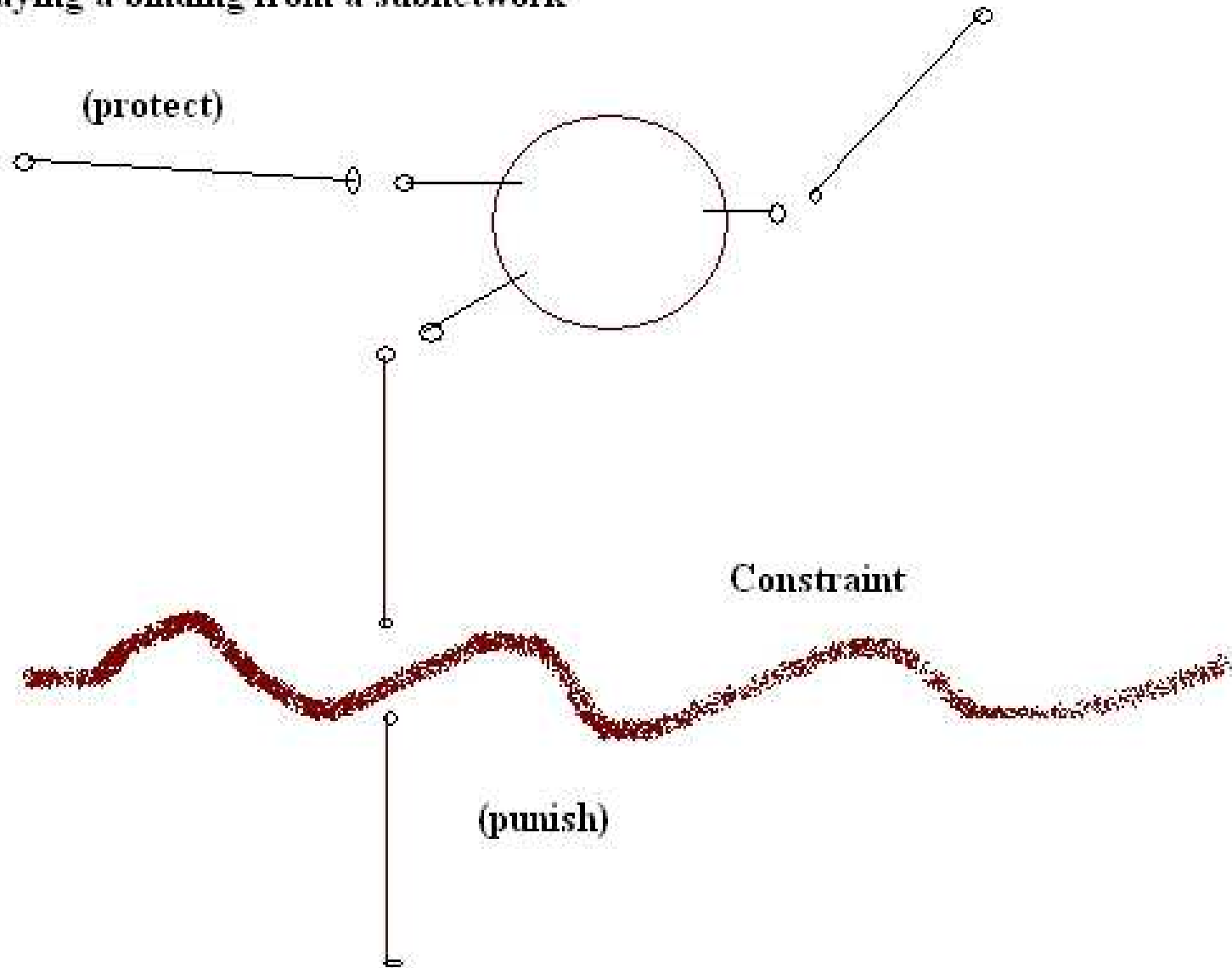
# Without Repressor:

total volume moved: 14



## Extended Hox

Hox displaying a binding from a subnetwork



# Coupling -- results

- Tight coupling -- where each program module relies on each of the other modules
- Content coupling --when one module modifies or relies on the internal workings of another module
- Control coupling --one module controlling the logic of another, by passing it information on what to do
- Message coupling --exchanges of messages or events (state)

-primarily message/data coupling

# Disadvantages...

Not obvious what next action will be, as this will be decided by neighbors and current state

Solution is specialized browsers

# What about the Program?

70 classes , 20 of which are test classes

300 hours / 6 months

roughly 1/2 megabyte of source

roughly 20 papers (mostly written for myself, others for NNG)

more than a prototype, less than a core