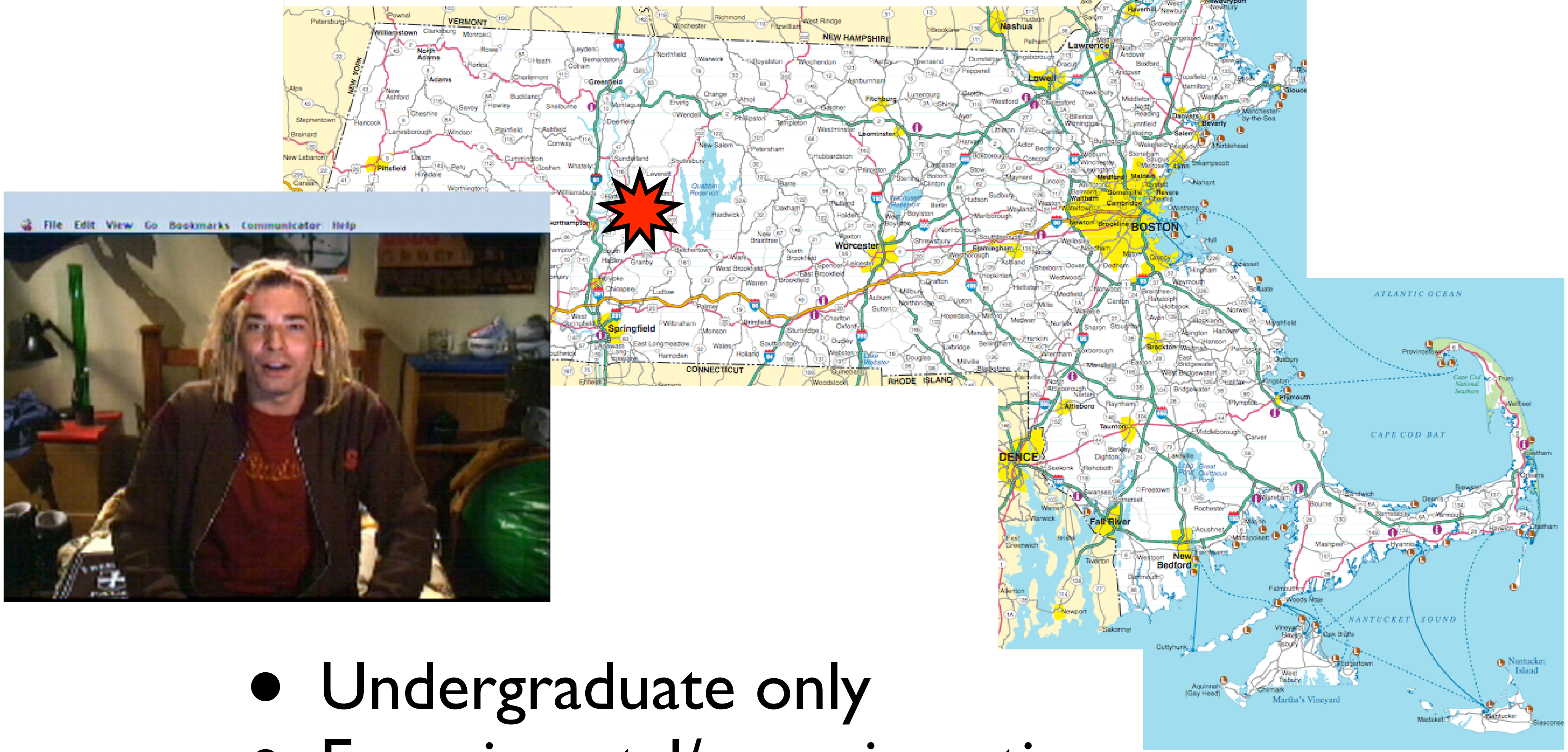
The Evolution of Identity and Modularity in Nature and Computation

Lee Spector
Cognitive Science
Hampshire College
Oberlin class of 1984
(Philosophy, TIMARA, WOBC, Tank Coop, ...)

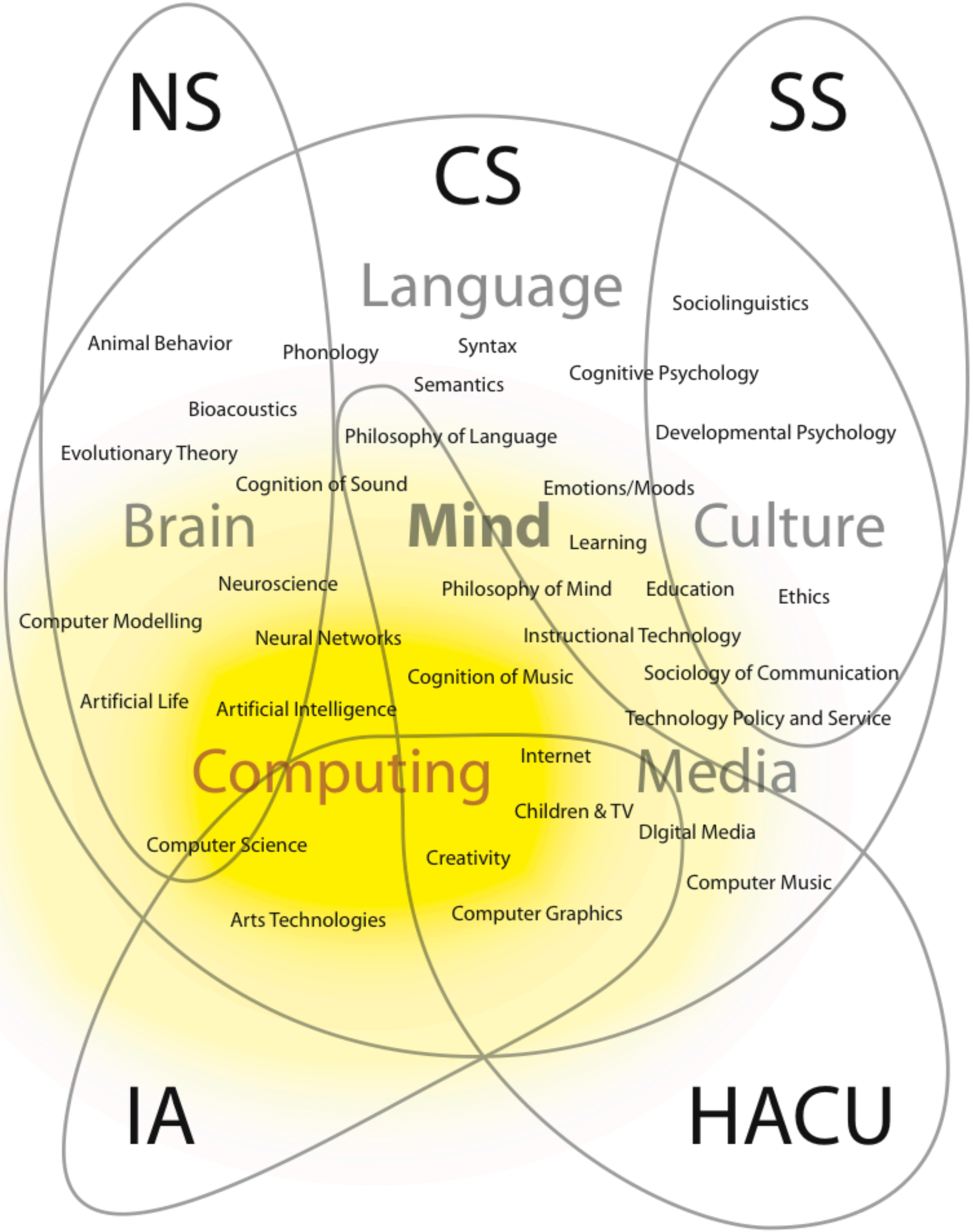
Overview

- Modularity
- Identity
- Evolving computer programs
- Evolving modular programs
- Implications

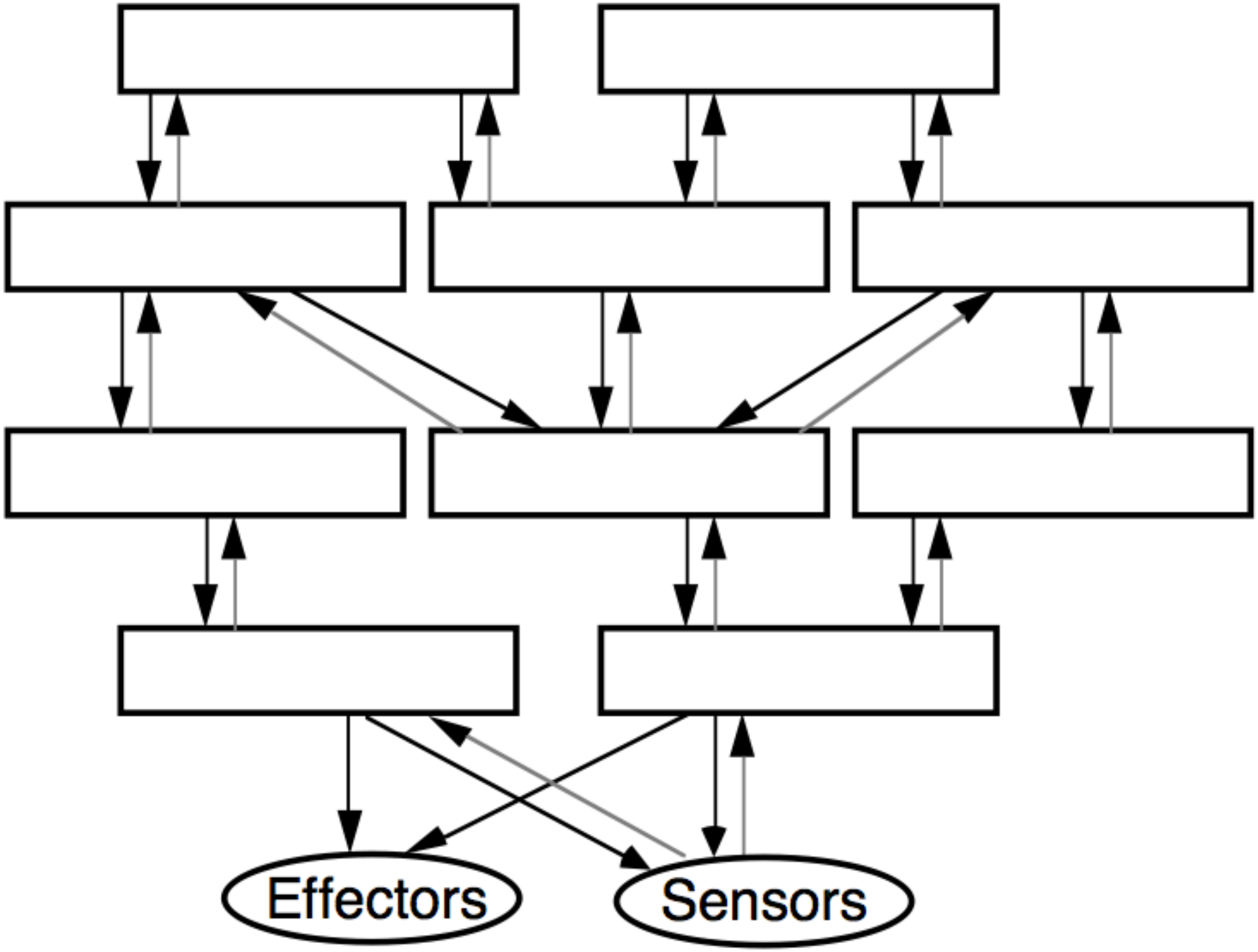
Hampshire College

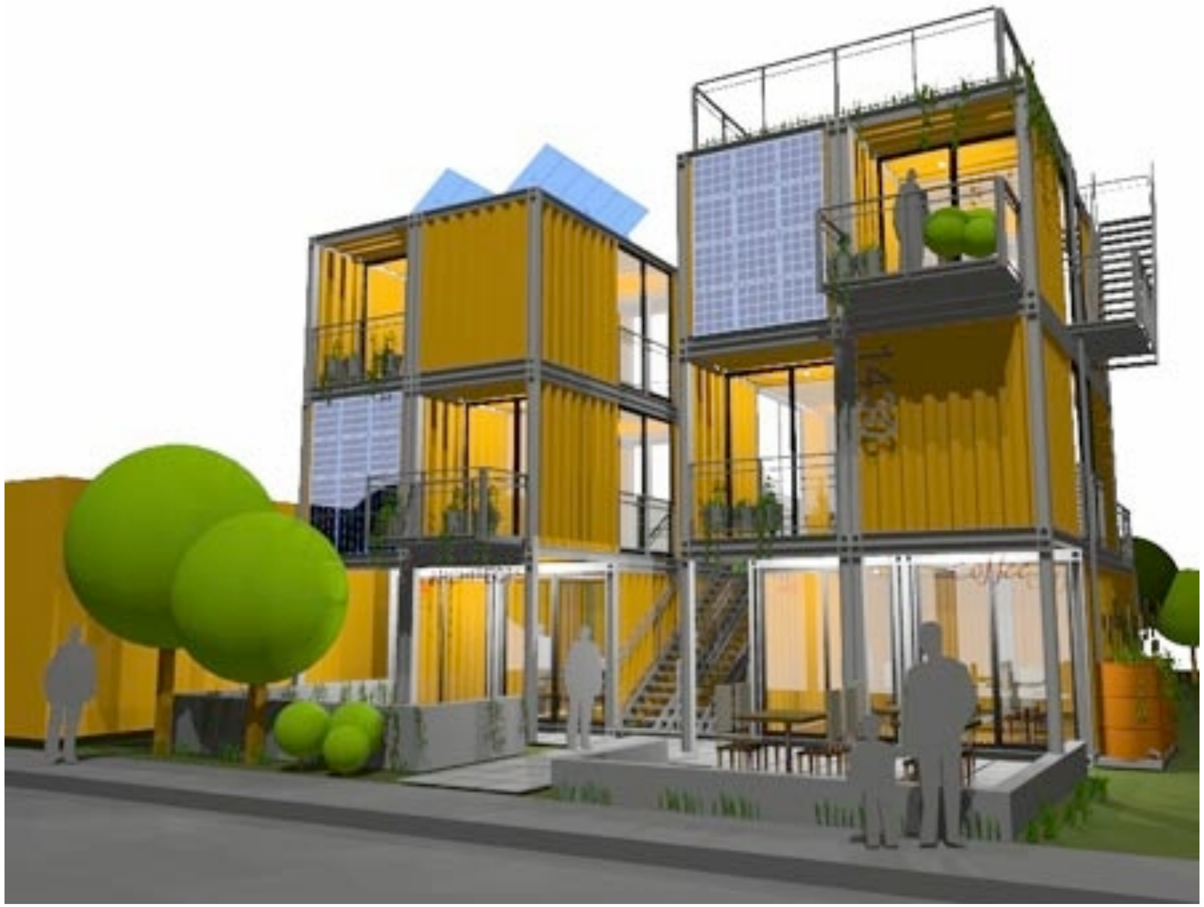


- Undergraduate only
- Experimental/experimenting
- Five Colleges consortium
- No grades, credits, majors, departments, ...
- School of Cognitive Science


NS
SS
CS
IA
HACU
Language
Brain
Mind
Culture
Computing
Media
Sociolinguistics
Animal Behavior
Phonology
Syntax
Semantics
Cognitive Psychology
Bioacoustics
Philosophy of Language
Developmental Psychology
Evolutionary Theory
Cognition of Sound
Emotions/Moods
Learning
Neuroscience
Philosophy of Mind
Education
Ethics
Computer Modelling
Neural Networks
Instructional Technology
Cognition of Music
Sociology of Communication
Artificial Life
Artificial Intelligence
Technology Policy and Service
Internet
Children & TV
DIgital Media
Computer Science
Creativity
Computer Music
Arts Technologies
Computer Graphics

Modularity is Everywhere

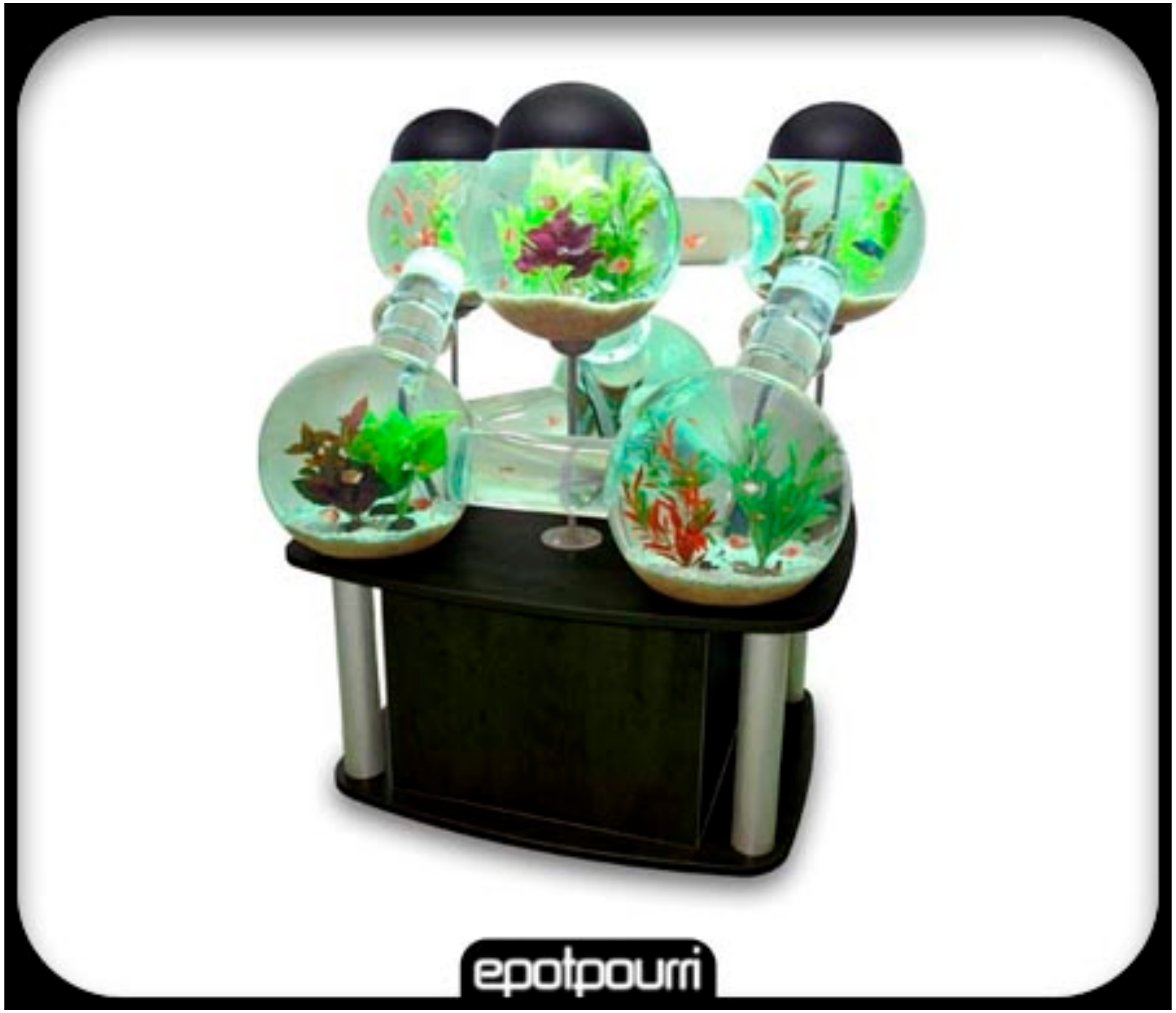




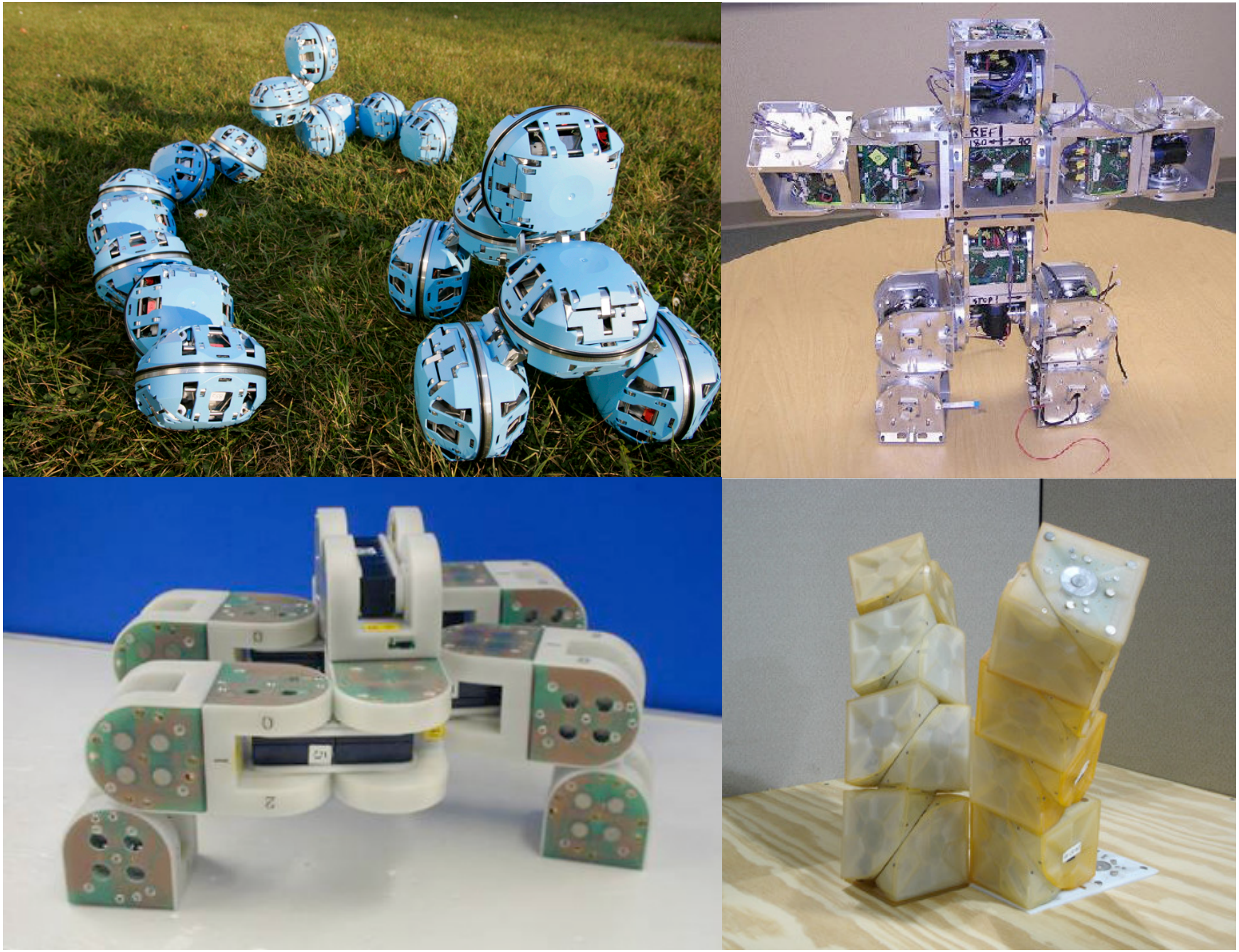
http://equitygreen.typepad.com/blog/2007/08/hybrid-seattle-.html#more

http://www.flickrfotos.com/modular-44-plastic-coffee-table-design/

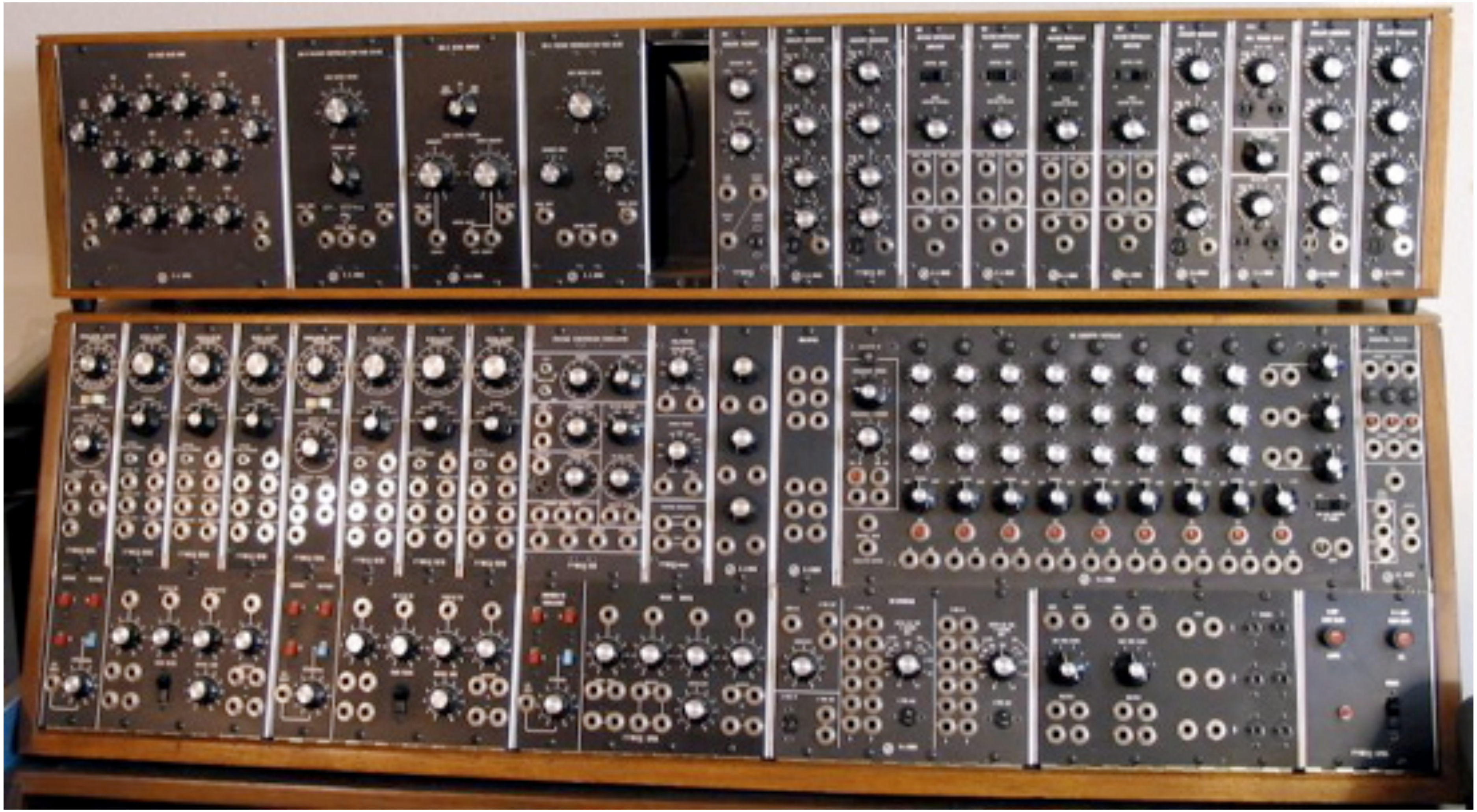
http://talkinterior.com/interior-design-vita-minimalist-modular-home/



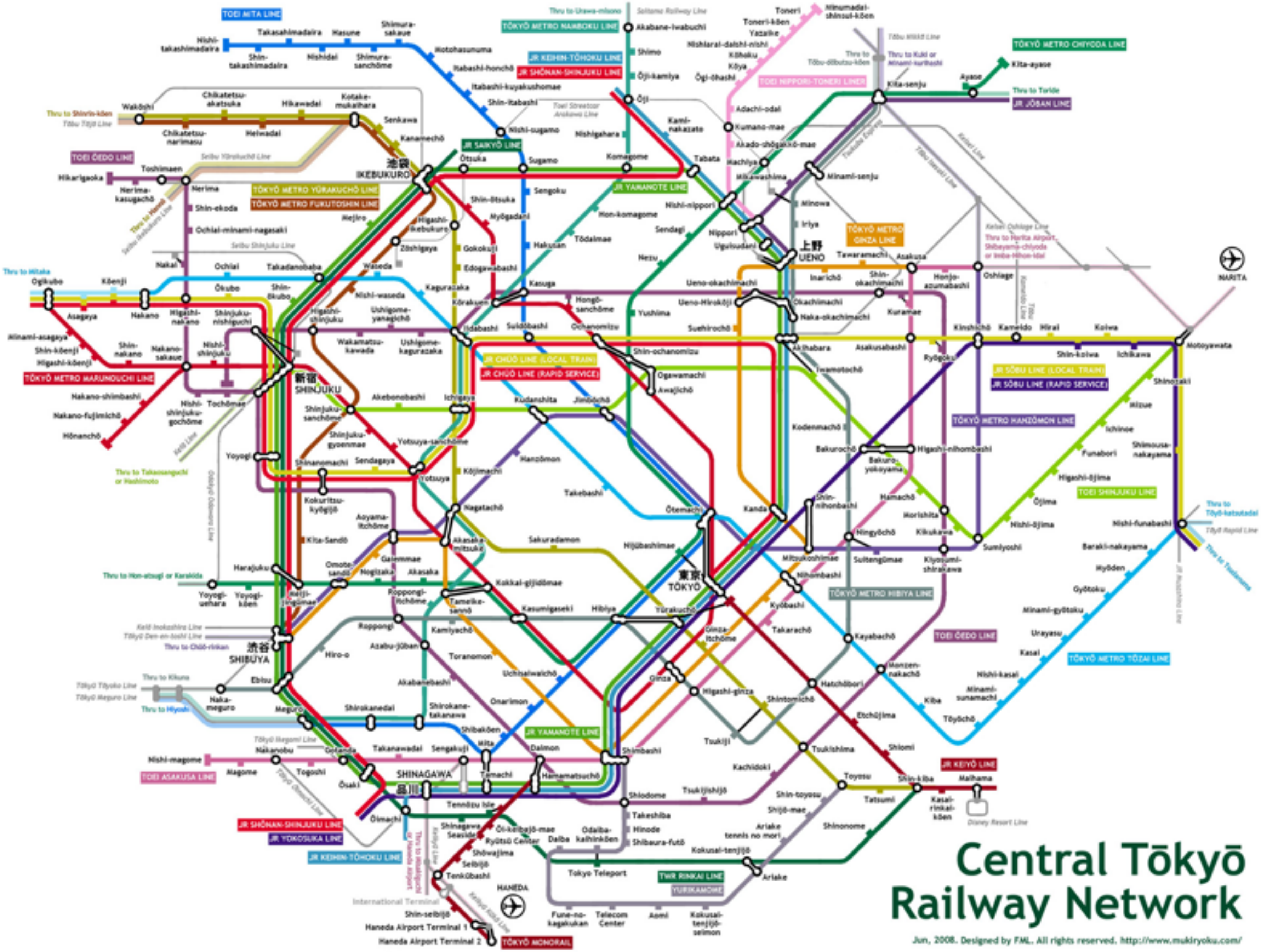
http://www.e-potpourri.com/index.php/2008/02/02/octopus-studios-silverfish-aquarium-boasts-modern-modular-design/



http://wyss.harvard.edu/viewevent/37/wyss-seminar-series-kasper-stoy
http://www.technovelgy.com/ct/Science-Fiction-News.asp?NewsNum=953
http://www.engadget.com/2005/03/26/m-tran-self-reconfigurable-modular-robot/
http://www.hizook.com/blog/2012/01/16/ted-talks-about-robots-and-robotics-part-1



http://www.synthtopia.com/content/2007/04/04/moog-55-modular-synthesizer/



http://mappery.com/map-of/Tokyo-Metro-Map

Modularity in Software

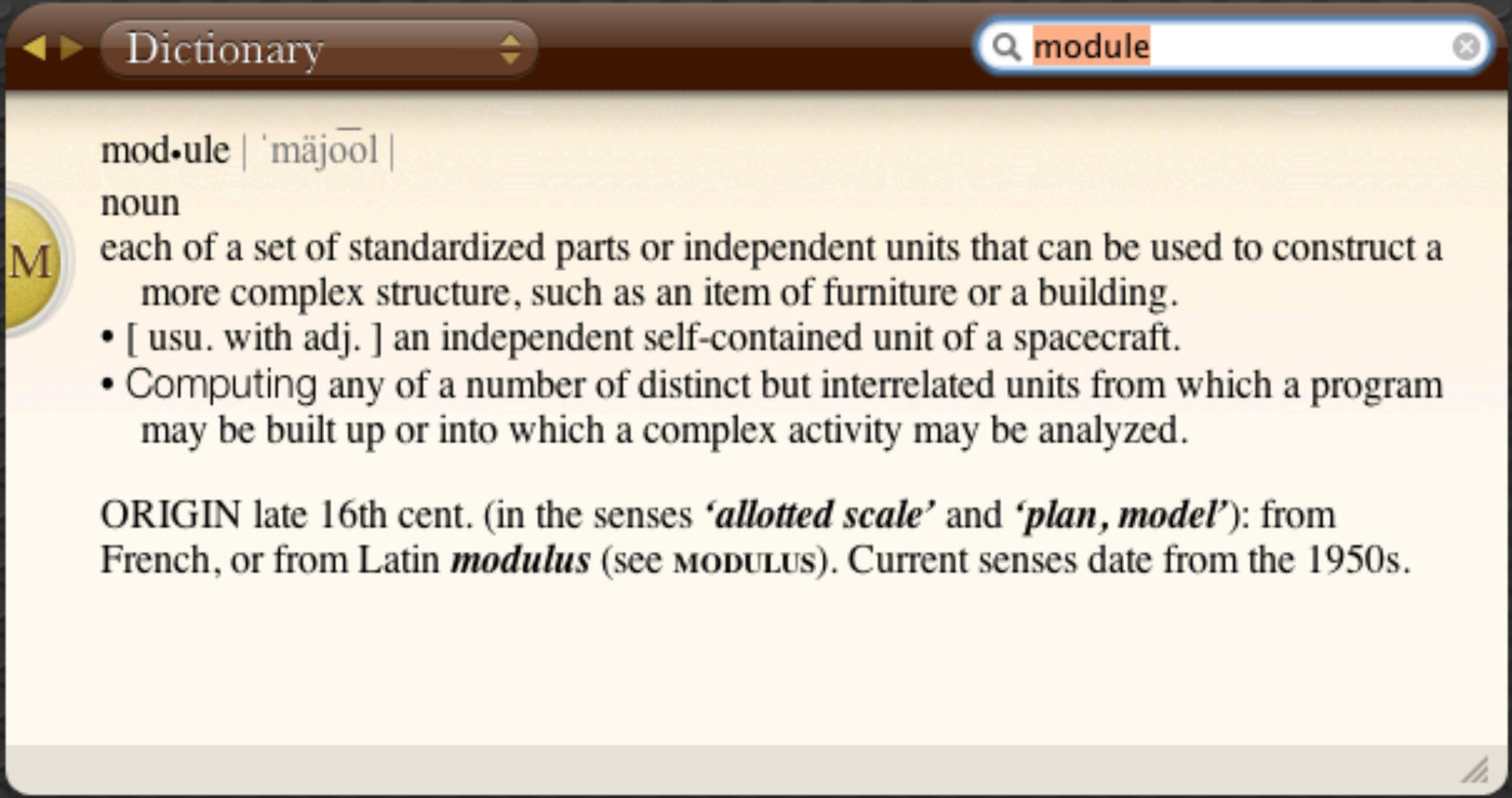
- Pervasive and widely acknowledged to be essential
- Modules may be functions, procedures, methods, classes, data structures, interfaces, etc.
- Modularity measures include coupling, cohesion, encapsulation, composability, etc.

http://en.wikipedia.org/wiki/File:Sa-fern.jpg

http://a-z-animals.com/animals/centipede/

Cognitive Science

- Long history of modularity theories: Gall, ... Simon, ... Fodor, ... Cermak and Craik, ... Gardner, ... Jackendoff, ... Grafman, ...
- Simon's "nearly decomposable systems"
- Fodor's features: domain specific, mandatory, fast, encapsulated, fixed architecture, characteristic patterns of ontogeny and failure
- Central vs. input systems
- Modest vs. massive



mod•ule | ˈmäjo͞ol |

noun

each of a set of standardized parts or independent units that can be used to construct a more complex structure, such as an item of furniture or a building.

- [usu. with adj.] an independent self-contained unit of a spacecraft.
- Computing any of a number of distinct but interrelated units from which a program may be built up or into which a complex activity may be analyzed.

ORIGIN late 16th cent. (in the senses ***'allotted scale'*** and ***'plan, model'***): from French, or from Latin ***modulus*** (see MODULUS). Current senses date from the 1950s.

Questions

- **Why** are modules everywhere?
- What are they good for?
- Where do they come from?
- What conditions permit or facilitate their emergence?

Identity

- How are modules recognized by other components of a system?
- Where do module identities come from?
- How can module identity co-evolve with modular architecture?

Holland's Tags

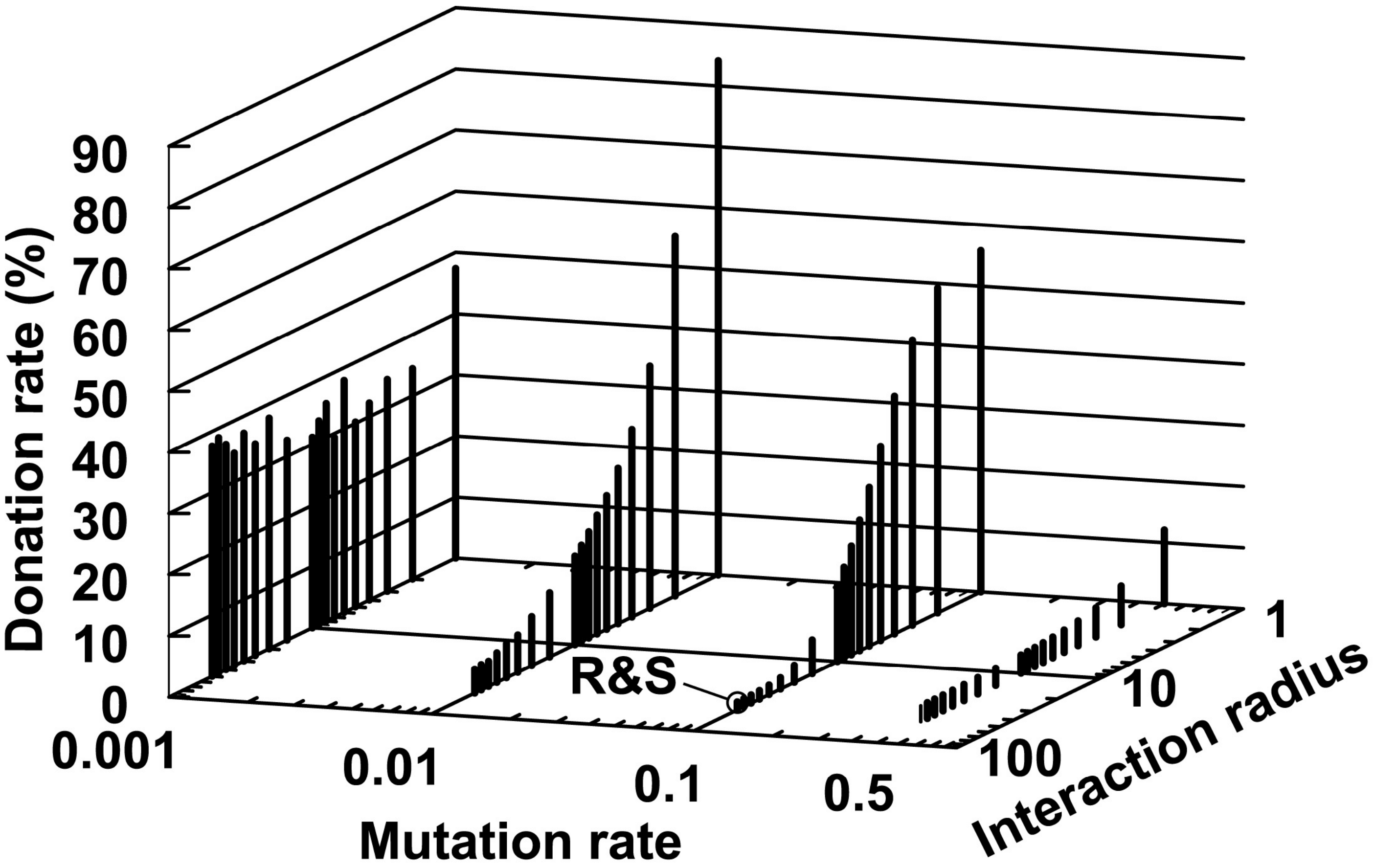
- Initially arbitrary identifiers that come to have meaning over time
- Appear to be present in some form in many different kinds of complex adaptive systems
- Examples range from immune systems to armies on a battlefield
- A general tool for the support of emergent complexity

Evolution of Altruism

- Puzzles/challenges/results since Darwin
- Explanations of altruism toward:
  - Kin
  - Reciprocating partners
  - Agents with good reputations

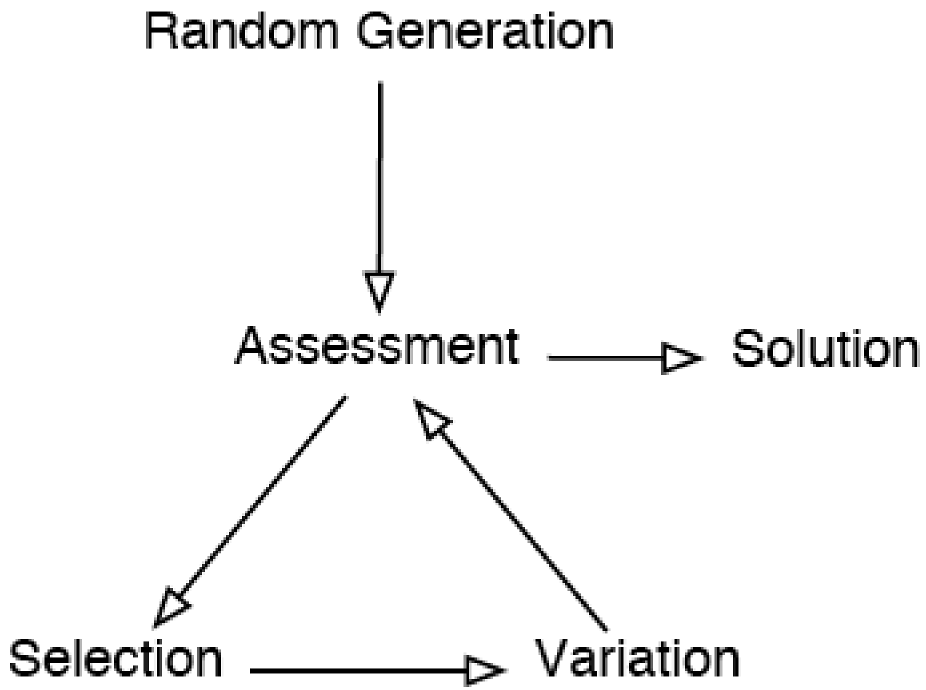
Tag-Based Altruism

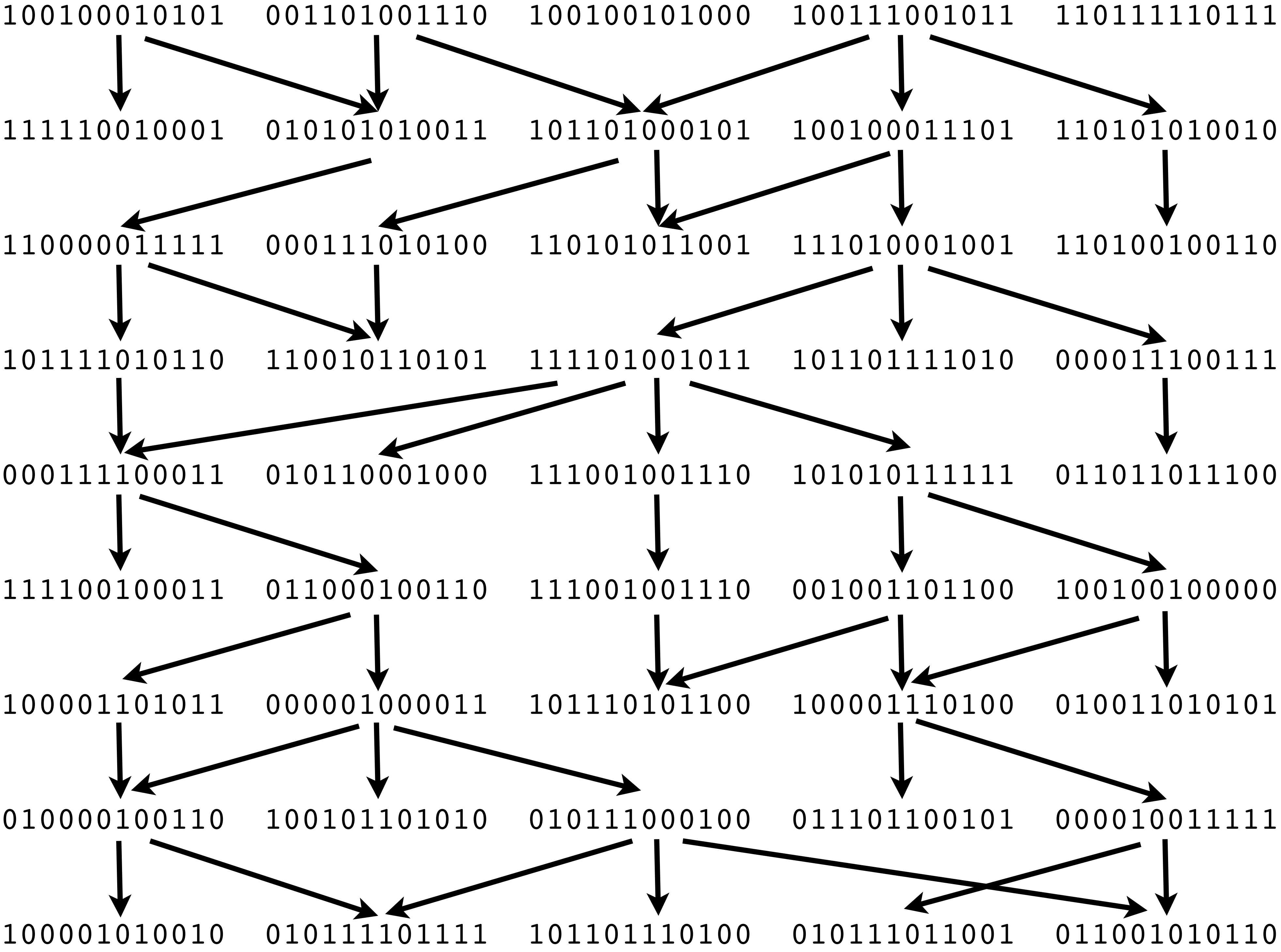
- Individuals have tags and tag-difference tolerances
- Donate when Δtags ≤ tolerance
- Riolo *et al.* (*Nature*, 2001) showed that tag-based altruism can evolve; Roberts & Sherratt (*Nature*, 2002) claimed it would not evolve under more realistic conditions



Spector, L., and Klein, J. Genetic stability and territorial structure facilitate the evolution of tag-mediated altruism. In *Artificial Life*.

Evolutionary Computation



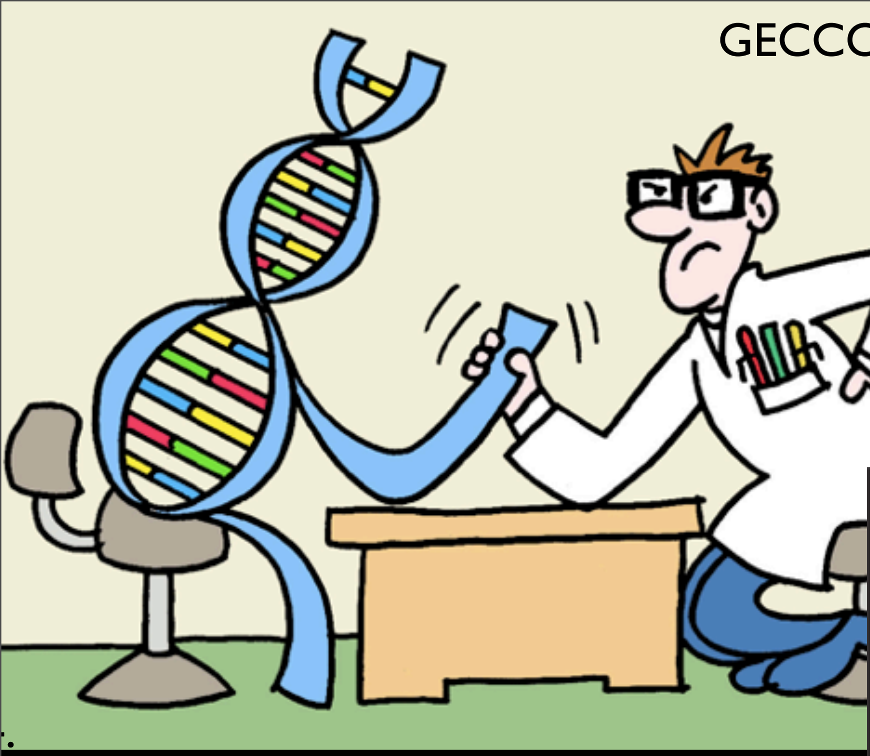

100100010101
001101001110
100100101000
100111001011
110111110111
111110010001
010101010011
101101000101
100100011101
110101010010
110000011111
000111010100
110101011001
111010001001
110100100110
101111010110
110010110101
111101001011
101101111010
000011100111
000111100011
010110001000
111001001110
101010111111
011011011100
111100100011
011000100110
111001001110
001001101100
100100100000
100001101011
000001000011
101110101100
100001110100
010011010101
010000100110
100101101010
010111000100
011101100101
000010011111
100001010010
010111101111
101101110100
010111011001
011001010110

Traditional Genetic Algorithms

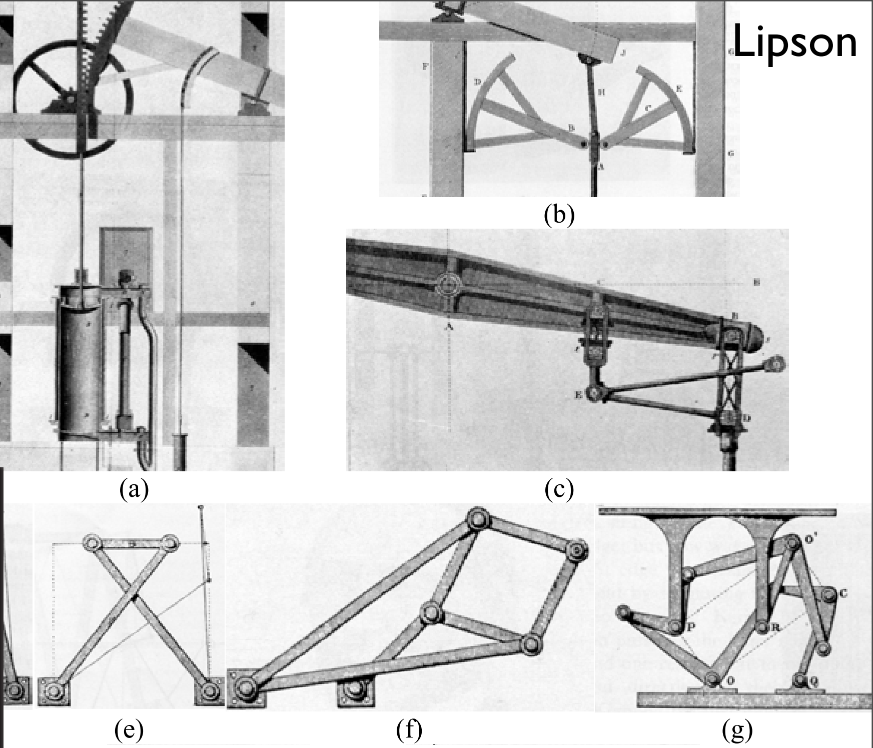
- Interesting dynamics
- Rarely solve interesting hard problems

Genetic Programming

- Evolutionary computing to produce executable computer programs.
- Programs are tested by executing them.



GECCO Humies



Lipson

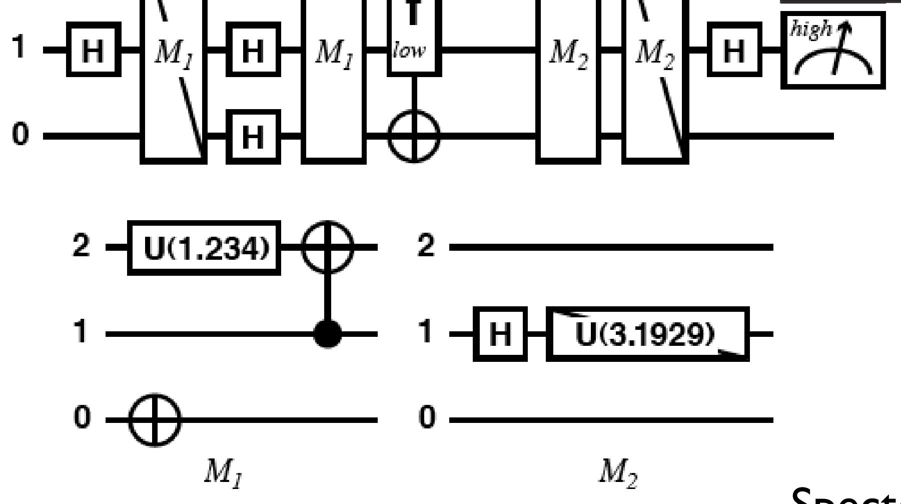
Volume 22 Number 3
Summer 2008

AIEDAM

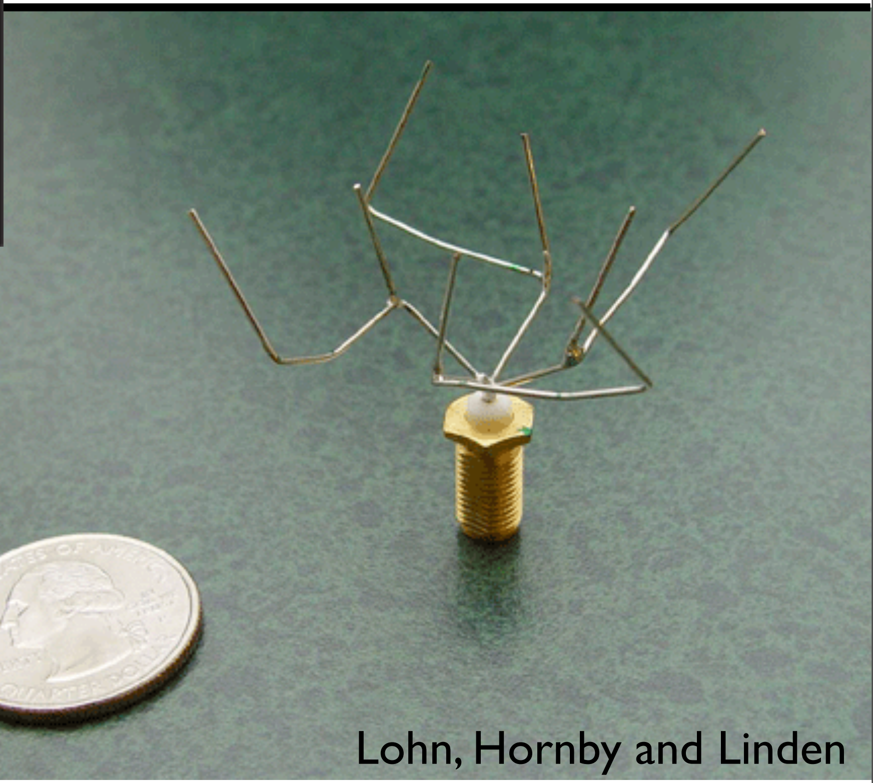
Artificial Intelligence for Engineering Design, Analysis and Manufacturing

David C. Brown, Editor

Cambridge University Press



Spector



Lohn, Hornby and Linden

Program Representations

- Lisp-style symbolic expressions (Koza, ...).
- Purely functional/lambda expressions (Walsh, Yu, ...).
- Linear sequences of machine/byte code (Nordin et al., ...).
- Artificial assembly-like languages (Ray, Adami, ...).
- Stack-based languages (Perkis, Spector, Stoffel, Tchernev, ...).
- Graph-structured programs (Teller, Globus, ...).
- Object hierarchies (Bruce, Abbott, Schmutter, Lucas, ...)
- Fuzzy rule systems (Tunstel, Jamshidi, ...)
- Logic programs (Osborn, Charif, Lamas, Dubossarsky, ...).
- Strings, grammar-mapped to arbitrary languages (O'Neill, Ryan, ...).

Mutating Lisp

```
(+ (* X Y)
   (+ 4 (- Z 23)))
```

```
(+ (* X Y)
   (+ 4 (- Z 23)))
```

```
(+ (- (+ 2 2) Z)
   (+ 4 (- Z 23)))
```

Recombining Lisp

```
Parent 1: (+ (* X Y)
              (+ 4 (- Z 23)))
Parent 2: (- (* 17 (+ 2 X))
             (* (- (* 2 Z) 1)
                (+ 14 (/ Y X))))

Child 1:  (+ (- (* 2 Z) 1)
             (+ 4 (- Z 23)))
Child 2:  (- (* 17 (+ 2 X))
             (* (* X Y)
                (+ 14 (/ Y X))))
```

Symbolic Regression

Given a set of data points, evolve a program that produces *y* from *x*.

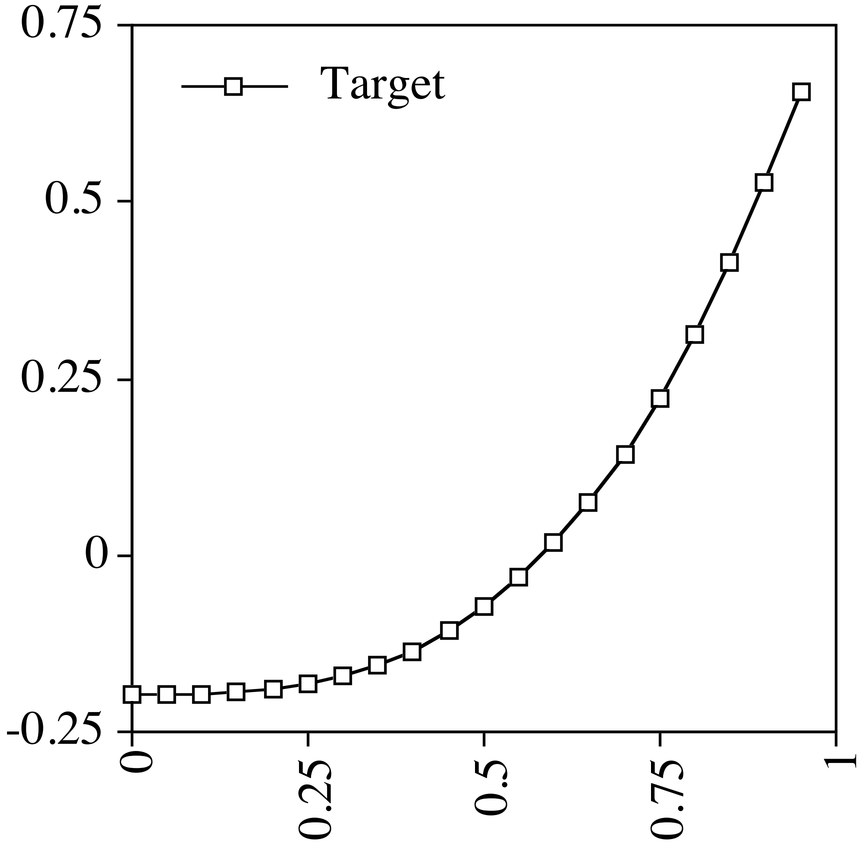
Primordial ooze: +, -, *, %, *x*, 0.1

Fitness = error (smaller is better)

GP Parameters

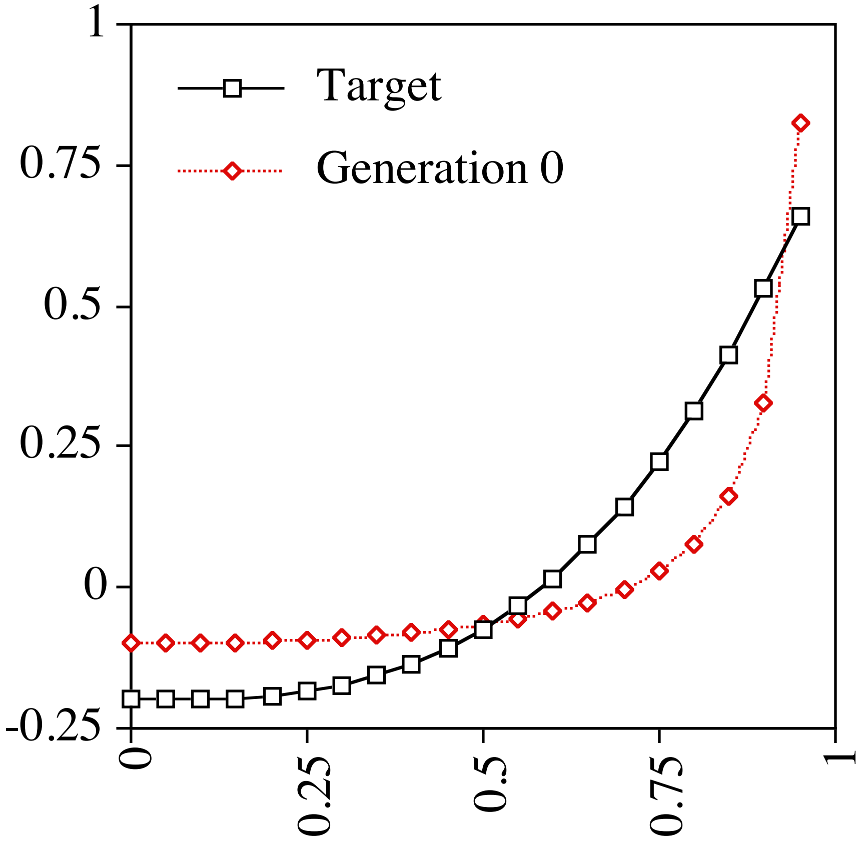
Maximum number of Generations: 51
Size of Population: 1000
Maximum depth of new individuals: 6
Maximum depth of new subtrees for mutants: 4
Maximum depth of individuals after crossover: 17
Fitness-proportionate reproduction fraction: 0.1
Crossover at any point fraction: 0.3
Crossover at function points fraction: 0.5
Selection method: FITNESS-PROPORTIONATE
Generation method: RAMPED-HALF-AND-HALF
Randomizer seed: 1.2

Evolving $y = x^3 - 0.2$



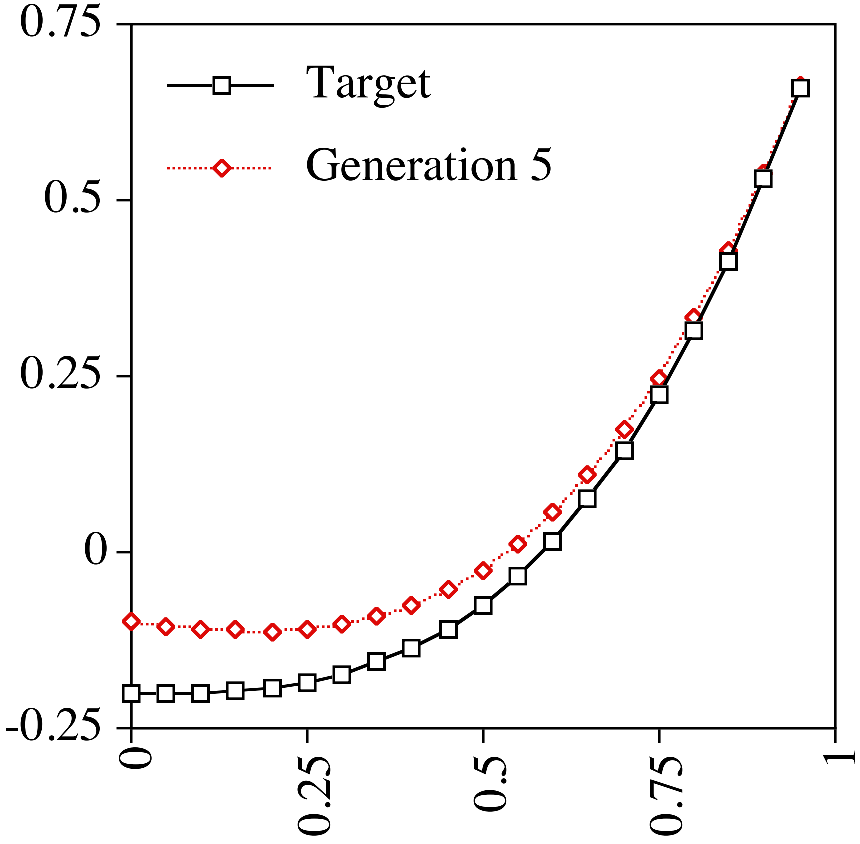
Best Program, Gen 0

```
(- (% (* 0.1
         (* X X))
      (- (% 0.1 0.1)
         (* X X)))
   0.1)
```



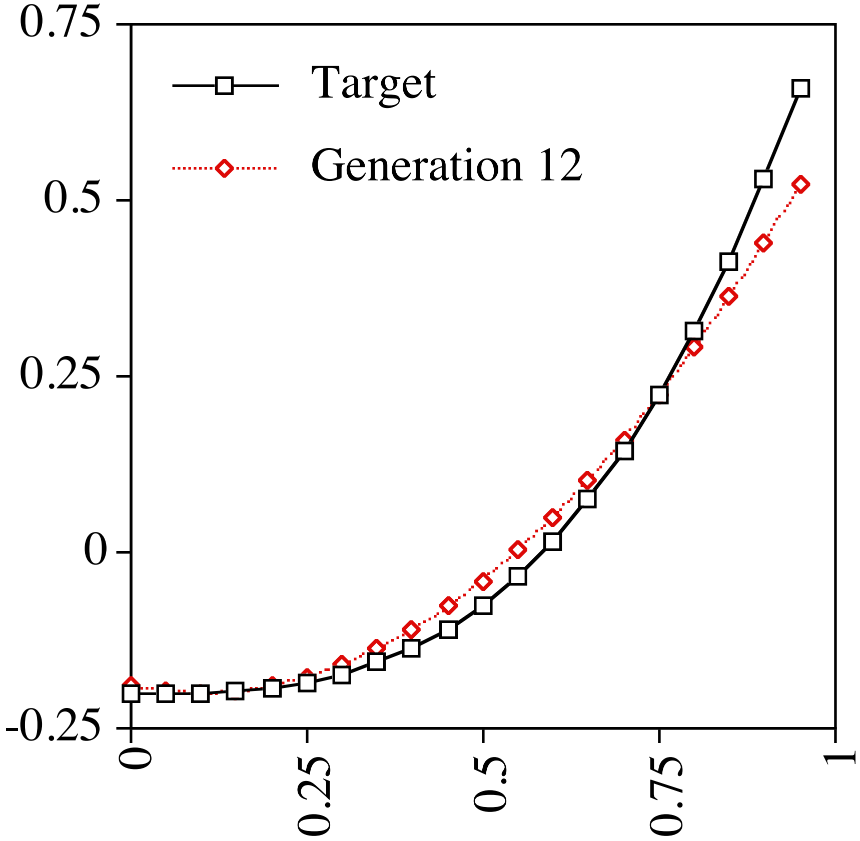
Best Program, Gen 5

```
(- (* (* (% X 0.1)
         (* 0.1 X))
      (- X
         (% 0.1 X)))
   0.1)
```



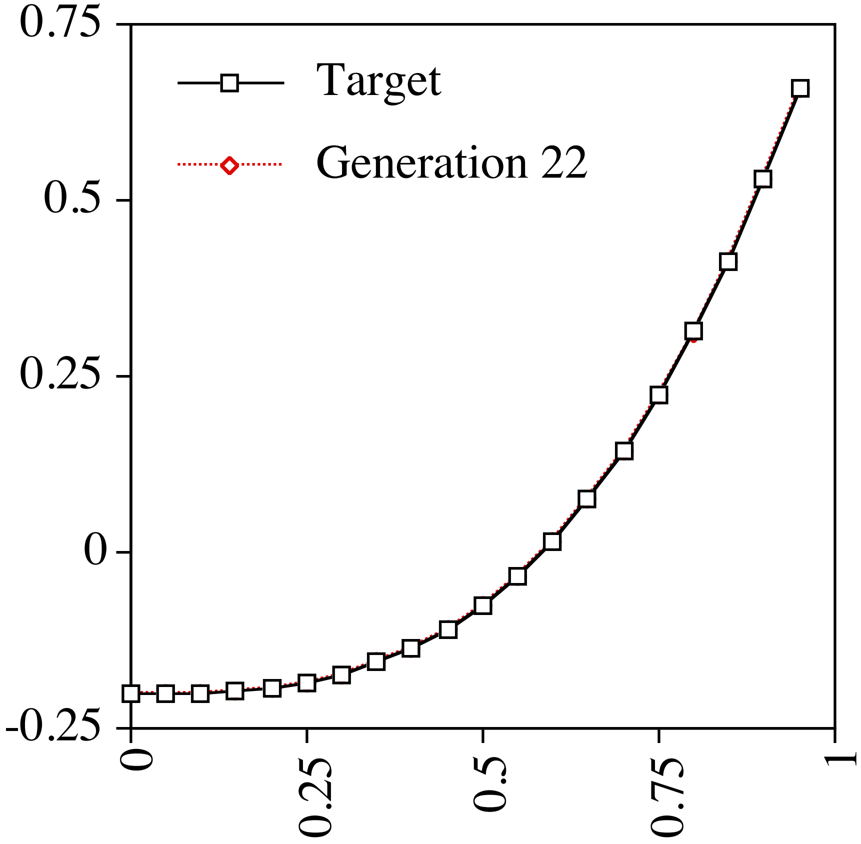
Best Program, Gen 12

```
(+ (- (- 0.1
         (- 0.1
            (- (* X X)
               (+ 0.1
                  (- 0.1
                     (* 0.1
                        0.1))))))
      (* X
         (* (% 0.1
               (% (* (* (- 0.1 0.1)
                        (+ X
                           (- 0.1 0.1)))
                     X)
                  (+ X (+ (- X 0.1)
                          (* X X)))))
            (+ 0.1 (+ 0.1 X)))))
   (* X X))
```



Best Program, Gen 22

```
(- (- (* X (* X X))
      0.1)
   0.1)
```



Genetic Programming for Finite Algebras

Lee Spector
Cognitive Science
Hampshire College
Amherst, MA 01002
lspector@hampshire.edu

David M. Clark
Mathematics
SUNY New Paltz
New Paltz, NY 12561
clarkd@newpaltz.edu

Ian Lindsay
Hampshire College
Amherst, MA 01002
iml04@hampshire.edu

Bradford Barr
Hampshire College
Amherst, MA 01002
bradford.barr@gmail.com

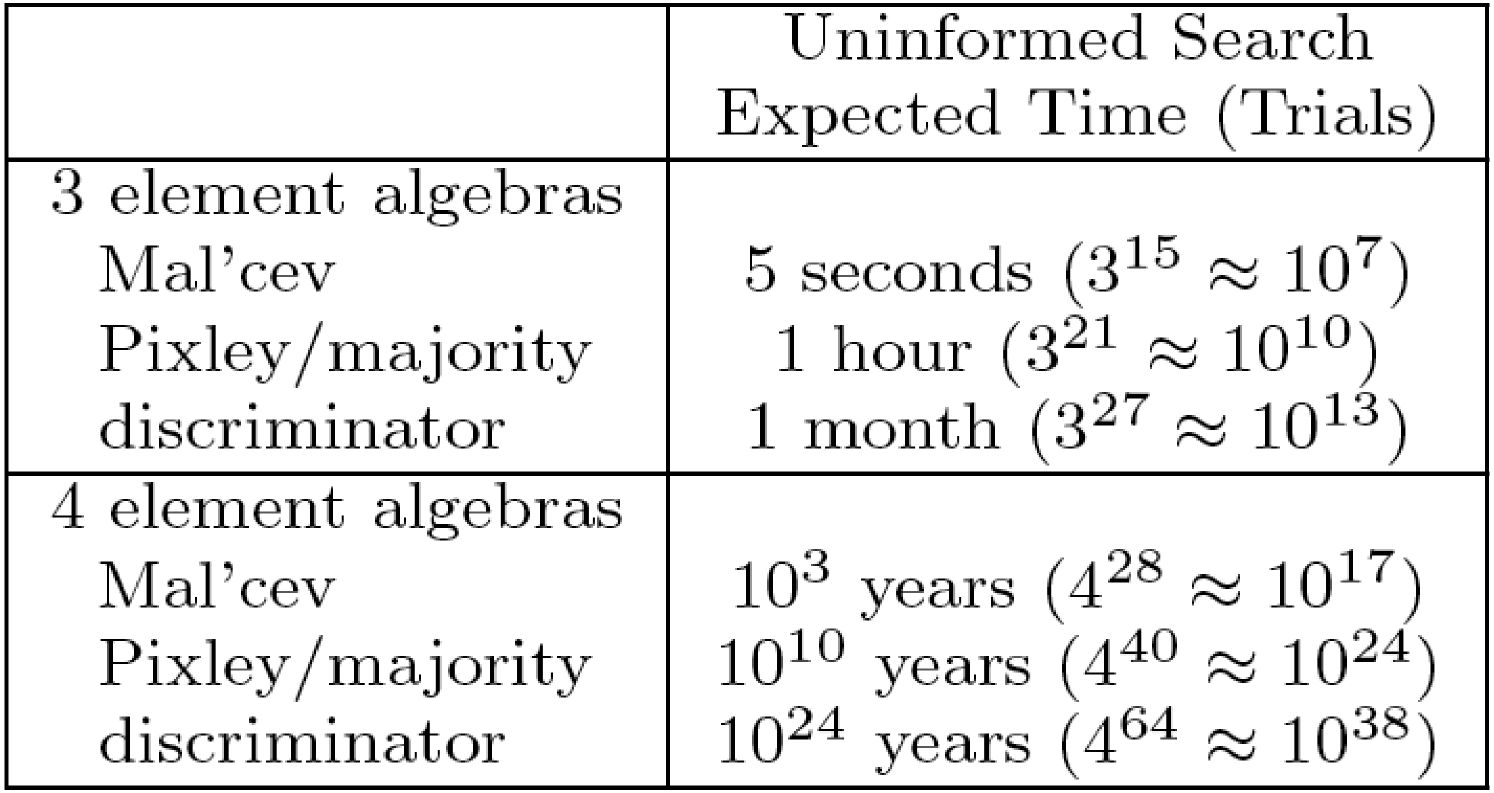
Jon Klein
Hampshire College
Amherst, MA 01002
jk@artificial.com

Humies 2008
GOLD MEDAL

Goal

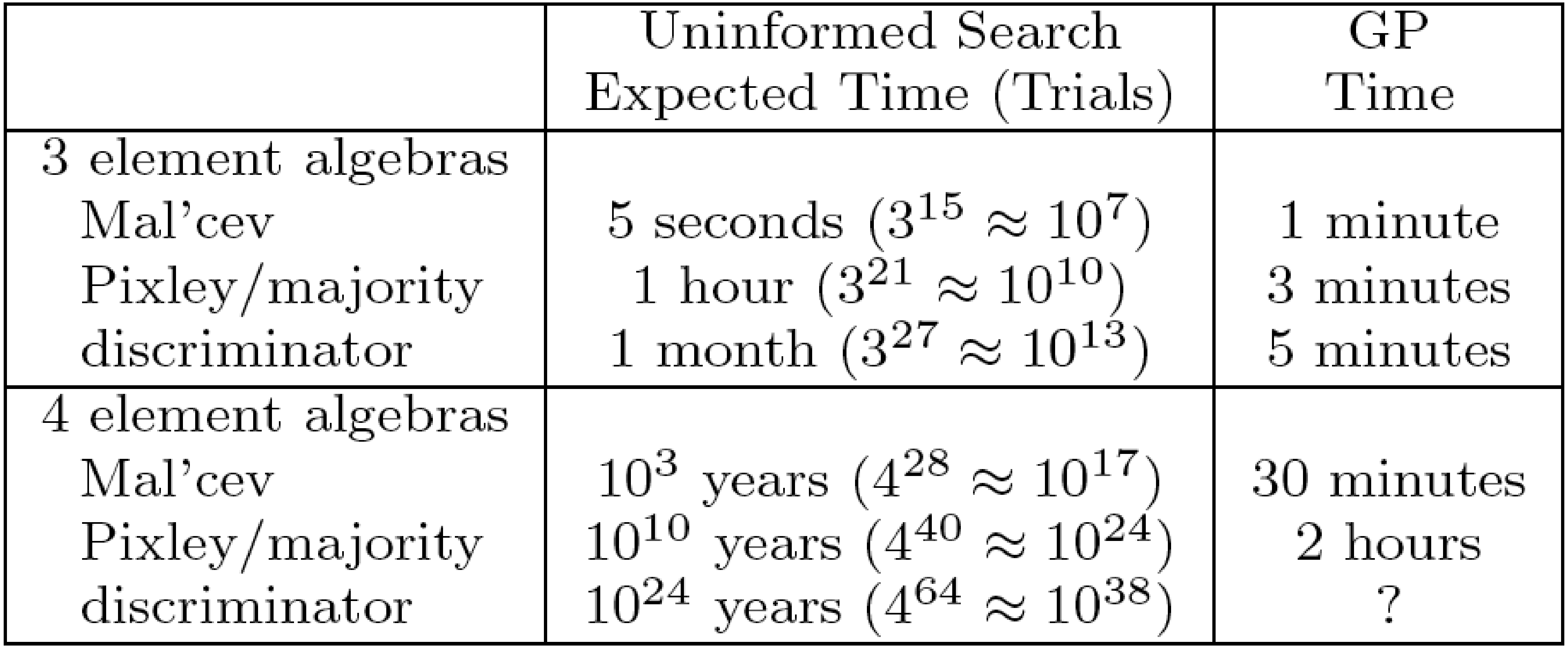
- Find finite algebra terms that have certain special properties
- For decades there was no way to produce these terms in general, short of exhaustive search
- Current best methods produce enormous terms

Significance, Time



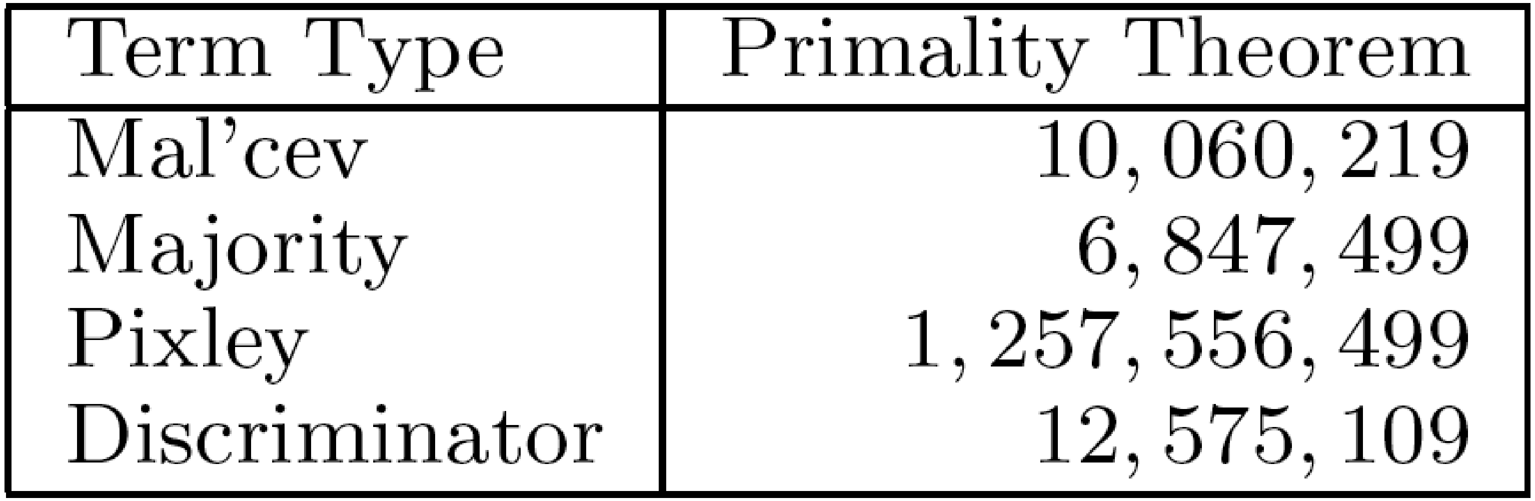
	Uninformed Search Expected Time (Trials)
3 element algebras	
Mal'cev	5 seconds ($3^{15} \approx 10^7$)
Pixley/majority	1 hour ($3^{21} \approx 10^{10}$)
discriminator	1 month ($3^{27} \approx 10^{13}$)
4 element algebras	
Mal'cev	10^3 years ($4^{28} \approx 10^{17}$)
Pixley/majority	10^{10} years ($4^{40} \approx 10^{24}$)
discriminator	10^{24} years ($4^{64} \approx 10^{38}$)

Significance, Time



	Uninformed Search Expected Time (Trials)	GP Time
3 element algebras		
Mal'cev	5 seconds $(3^{15} \approx 10^7)$	1 minute
Pixley/majority	1 hour $(3^{21} \approx 10^{10})$	3 minutes
discriminator	1 month $(3^{27} \approx 10^{13})$	5 minutes
4 element algebras		
Mal'cev	10^3 years $(4^{28} \approx 10^{17})$	30 minutes
Pixley/majority	10^{10} years $(4^{40} \approx 10^{24})$	2 hours
discriminator	10^{24} years $(4^{64} \approx 10^{38})$	?

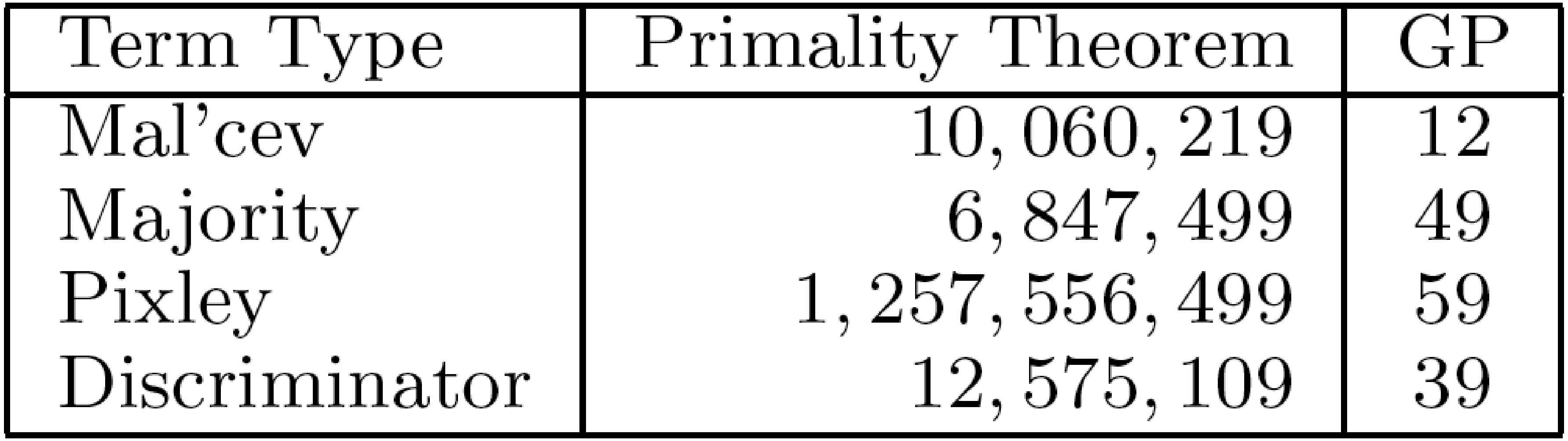
Significance, Size



Term Type	Primality Theorem
Mal'cev	$10,060,219$
Majority	$6,847,499$
Pixley	$1,257,556,499$
Discriminator	$12,575,109$

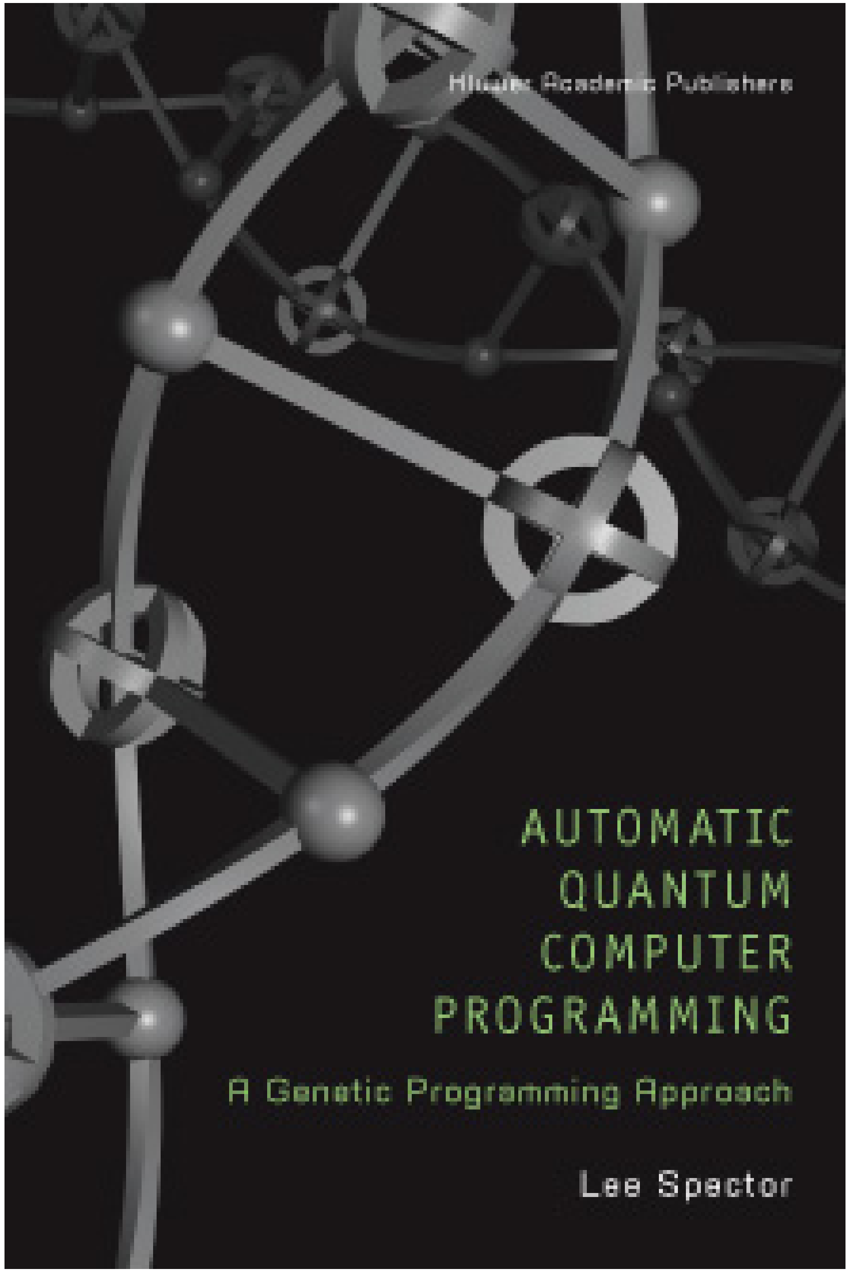
(for A_I)

Significance, Size



Term Type	Primality Theorem	GP
Mal'cev	10, 060, 219	12
Majority	6, 847, 499	49
Pixley	1, 257, 556, 499	59
Discriminator	12, 575, 109	39

(for A_I)



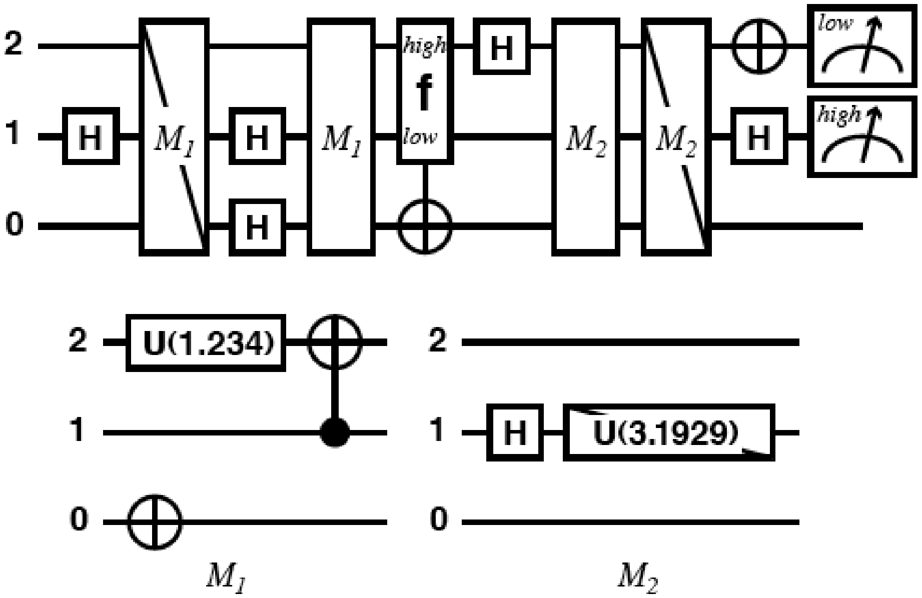


Figure 8.7. A gate array diagram for an evolved version of Grover's database search algorithm for a 4-item database. The full gate array is shown at the top, with M_1 and M_2 standing for the smaller gate arrays shown at the bottom. A diagonal line through a gate symbol indicates that the matrix for the gate is transposed. The "f" gate is the oracle.

Humies 2004 GOLD MEDAL

Evolving Modular Programs

With “automatically defined functions”

- All programs in the population have the same, pre-specified architecture
- Genetic operators respect that architecture
- Complicated, brittle, limited...
- Architecture-altering operations: more so

Evolving Modular Programs

With “execution stack manipulation”

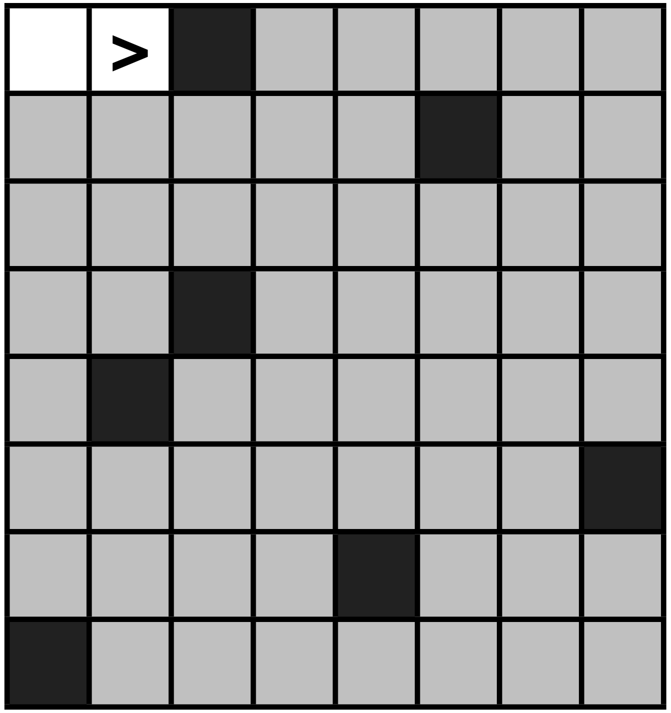
- Code queued for execution is stored on an “execution stack”
- Allow programs to duplicate and manipulate code that on the stack
- Simple types and uses of modules can be evolved easily
- Does not scale well to large/complex systems

Evolving Modular Programs

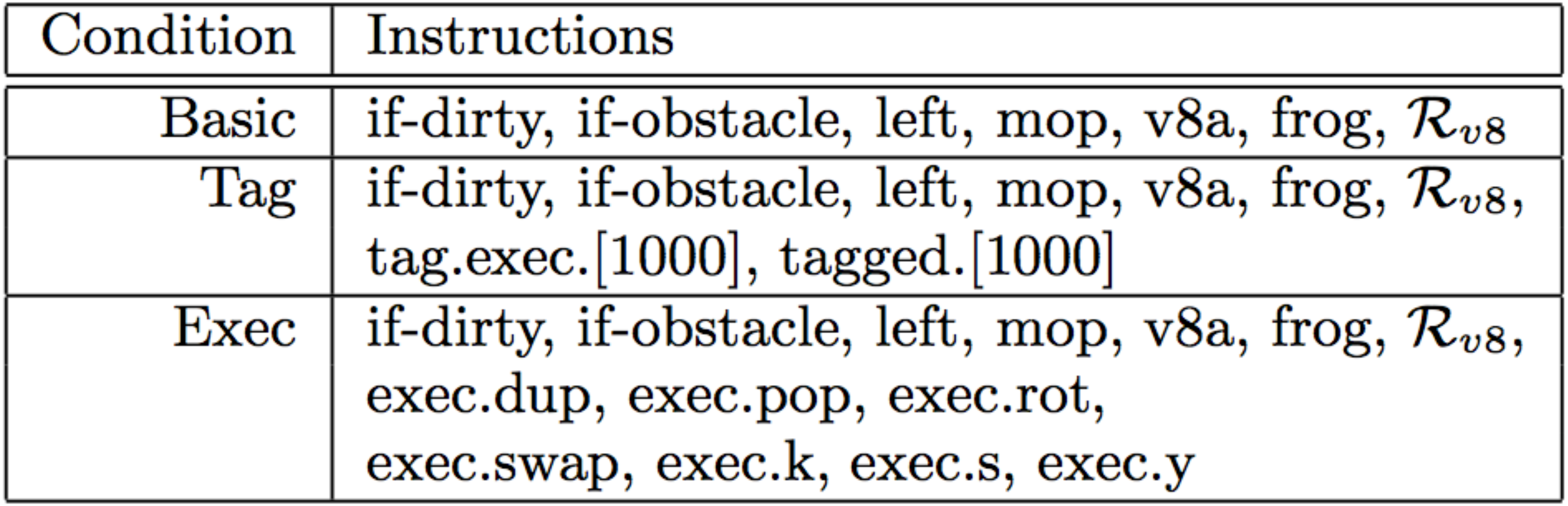
With tags

- Include instructions that tag code (modules)
- Include instructions that recall and execute modules by *closest matching* tag
- If a single module has been tagged then all tag references will recall modules
- The number of tagged modules can grow incrementally over evolutionary time

Dirt-Sensing, Obstacle-Avoiding Robot Problem

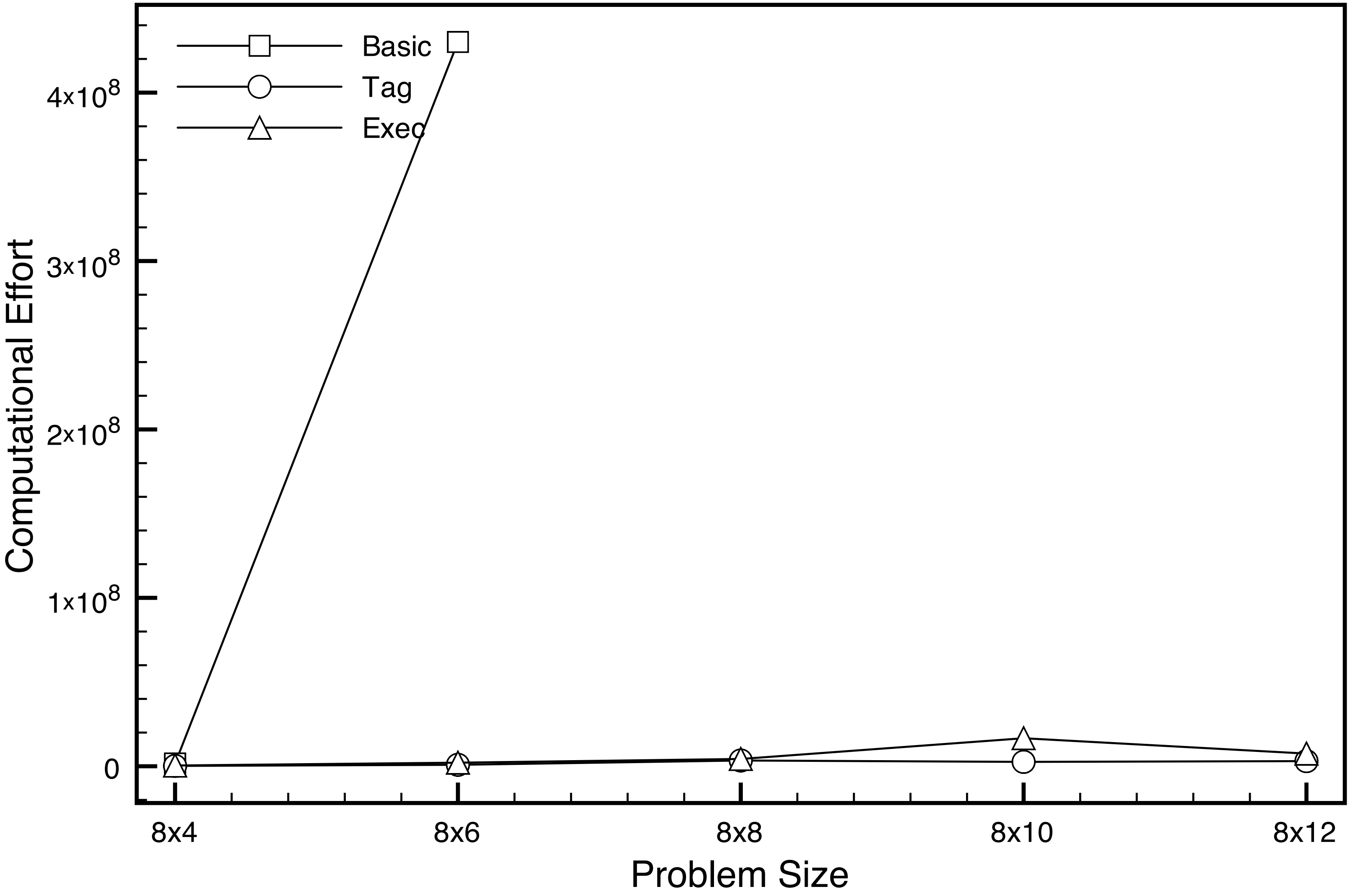


DSOAR Instructions

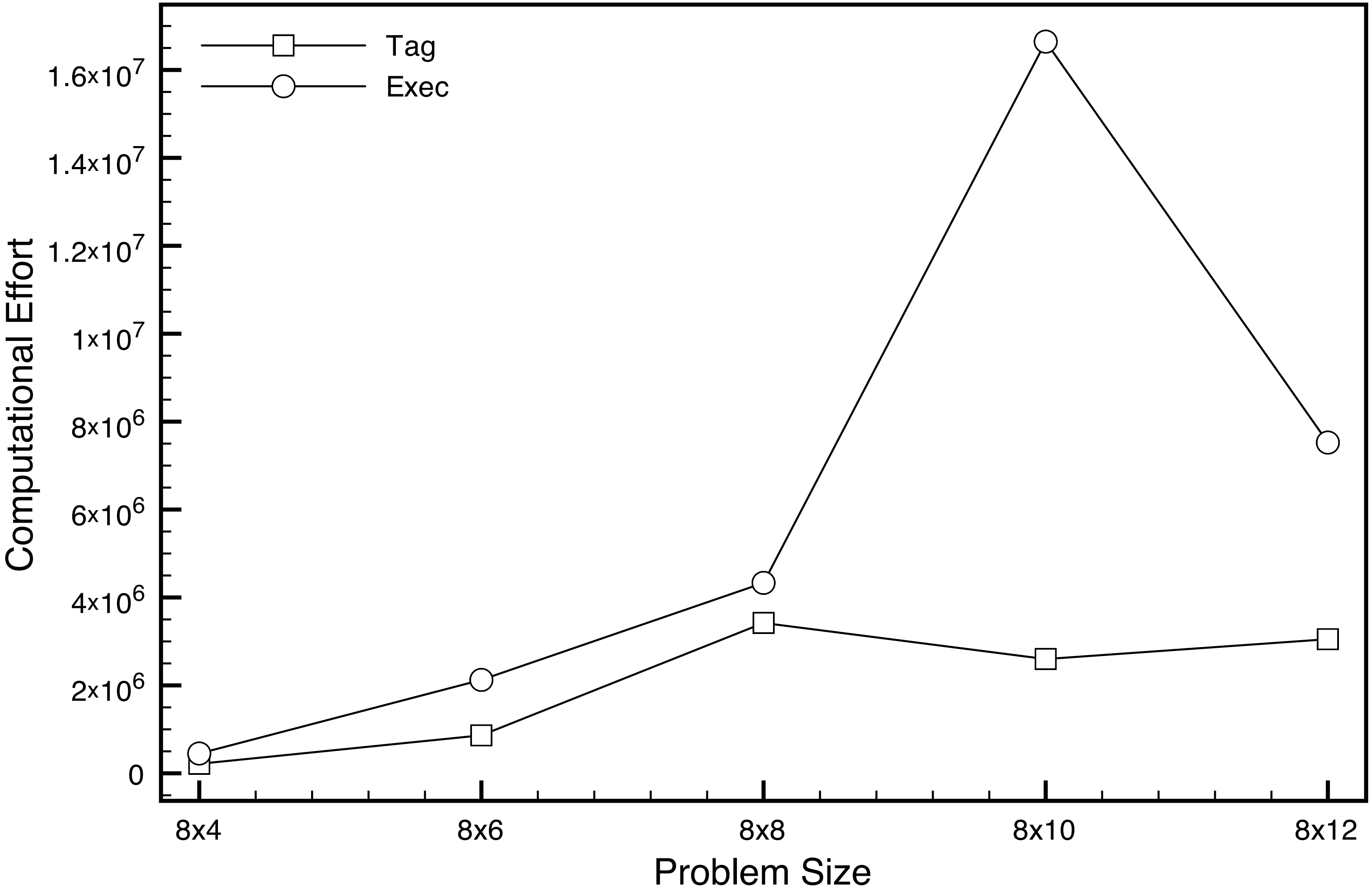


Condition	Instructions
Basic	if-dirty, if-obstacle, left, mop, v8a, frog, $\mathcal{R}_{v8}$
Tag	if-dirty, if-obstacle, left, mop, v8a, frog, $\mathcal{R}_{v8}$, tag.exec.[1000], tagged.[1000]
Exec	if-dirty, if-obstacle, left, mop, v8a, frog, $\mathcal{R}_{v8}$, exec.dup, exec.pop, exec.rot, exec.swap, exec.k, exec.s, exec.y

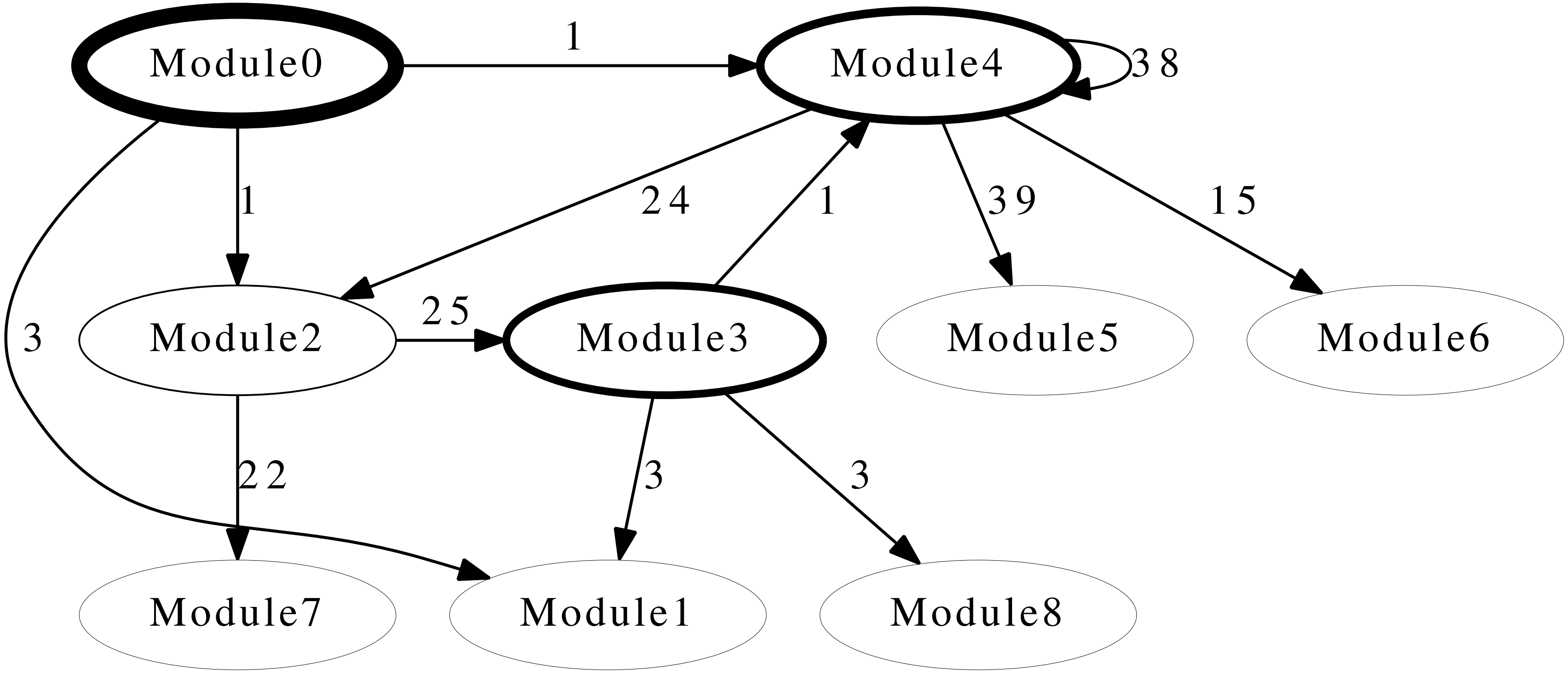
DSOAR Effort



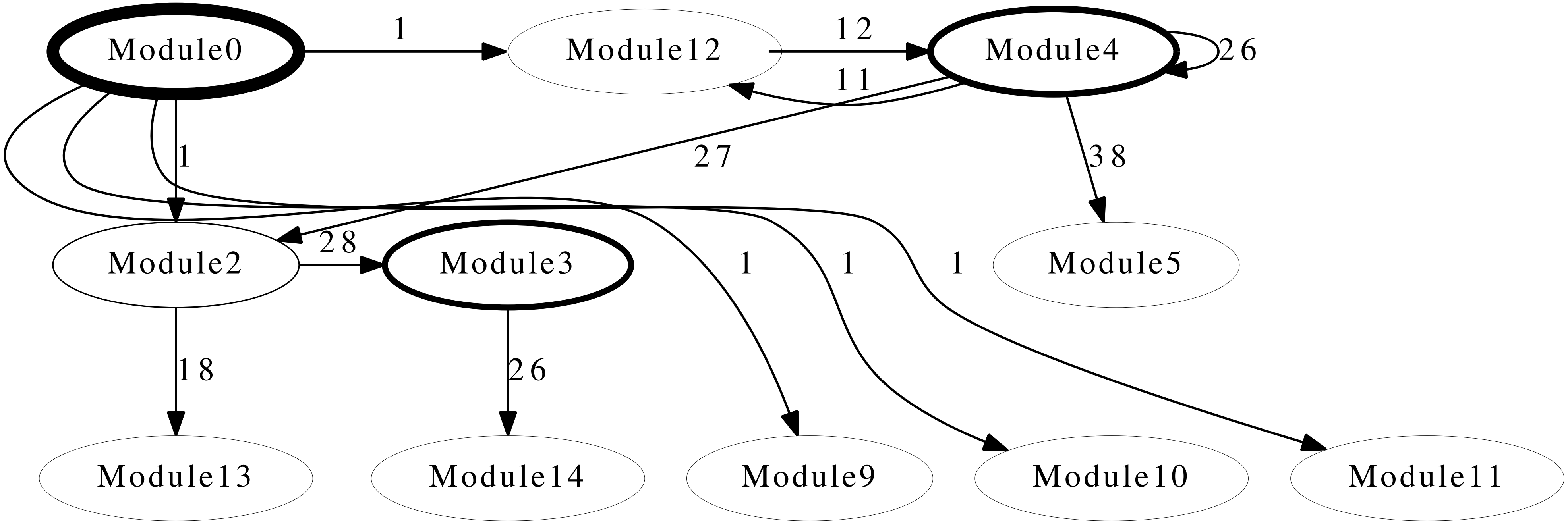
DSOAR Effort



Evolved DSOAR Architecture (in one environment)



Evolved DSOAR Architecture (in another environment)



Conclusions

- Tags provide an effective mechanism for the evolution of modular programs that solve difficult problems
- Tags may provide or explain mechanisms that support the evolution of modularity in a range of other systems, both natural and artificial