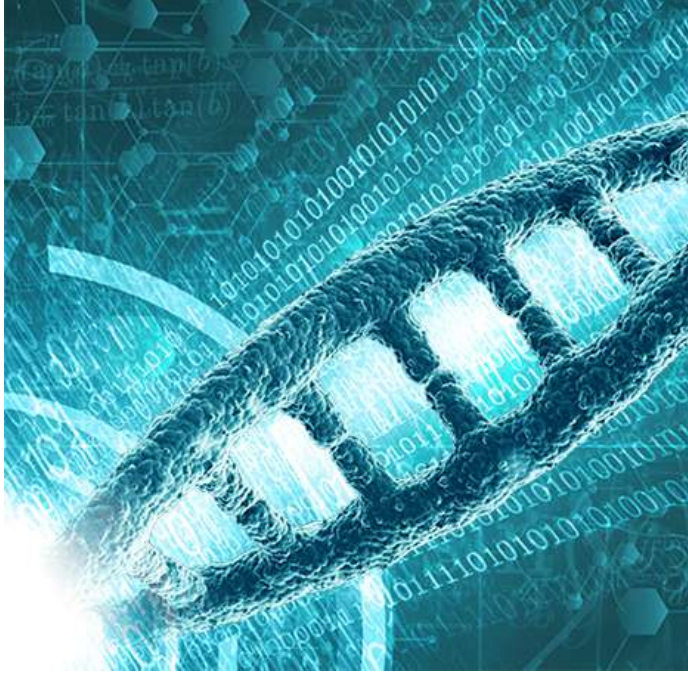


Genetic Programming and Evolving Complex Genomes

Alberto TONDA

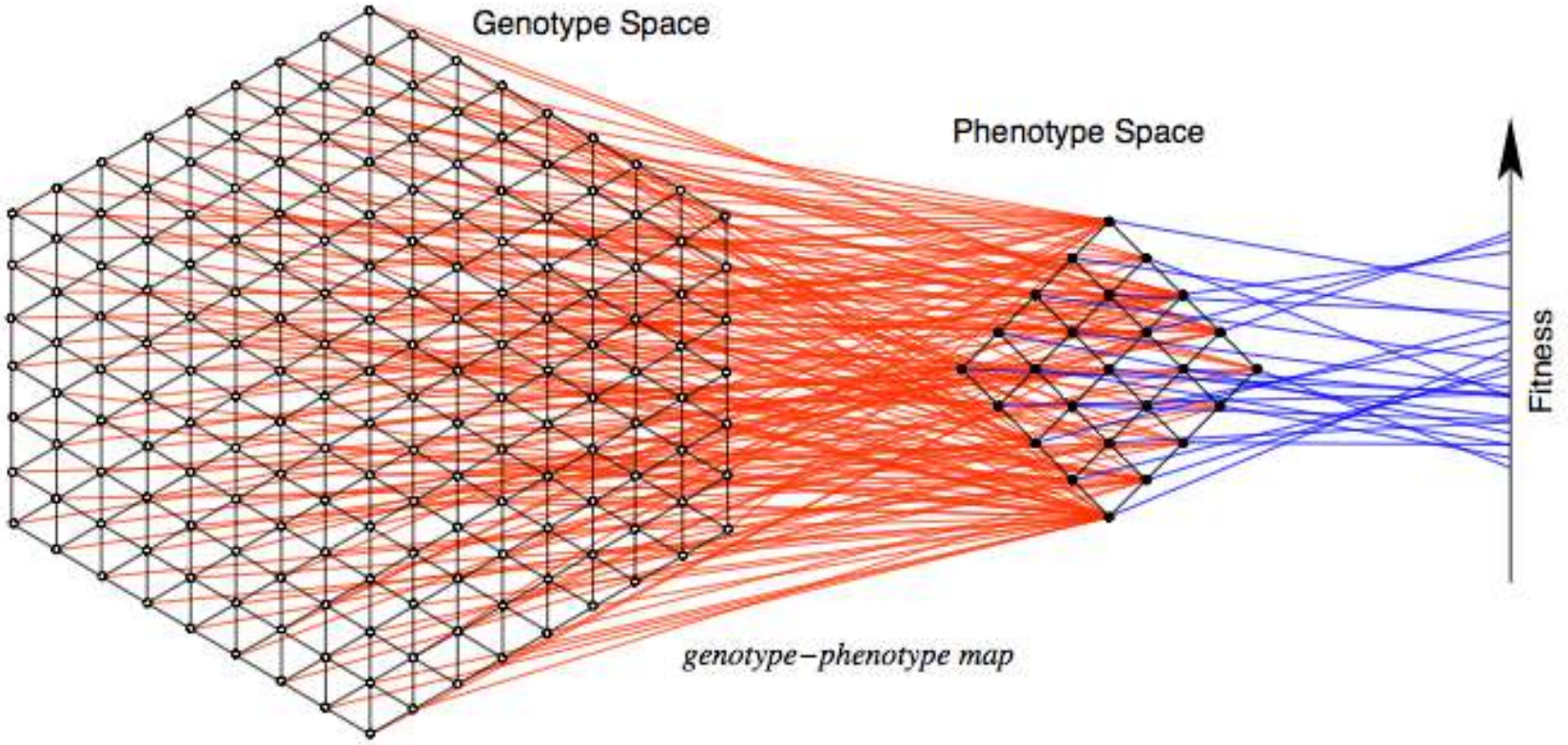
Permanent Researcher (CR)
UMR MIAPS, INRAe and Université Paris-Saclay
Équipe EKINOCS
91120, Palaiseau
alberto.tonda@inrae.fr

Outline

- Complex Genomes
- Genetic Programming
- Linear Genetic Programming
- Examples
- Evolve Everything!

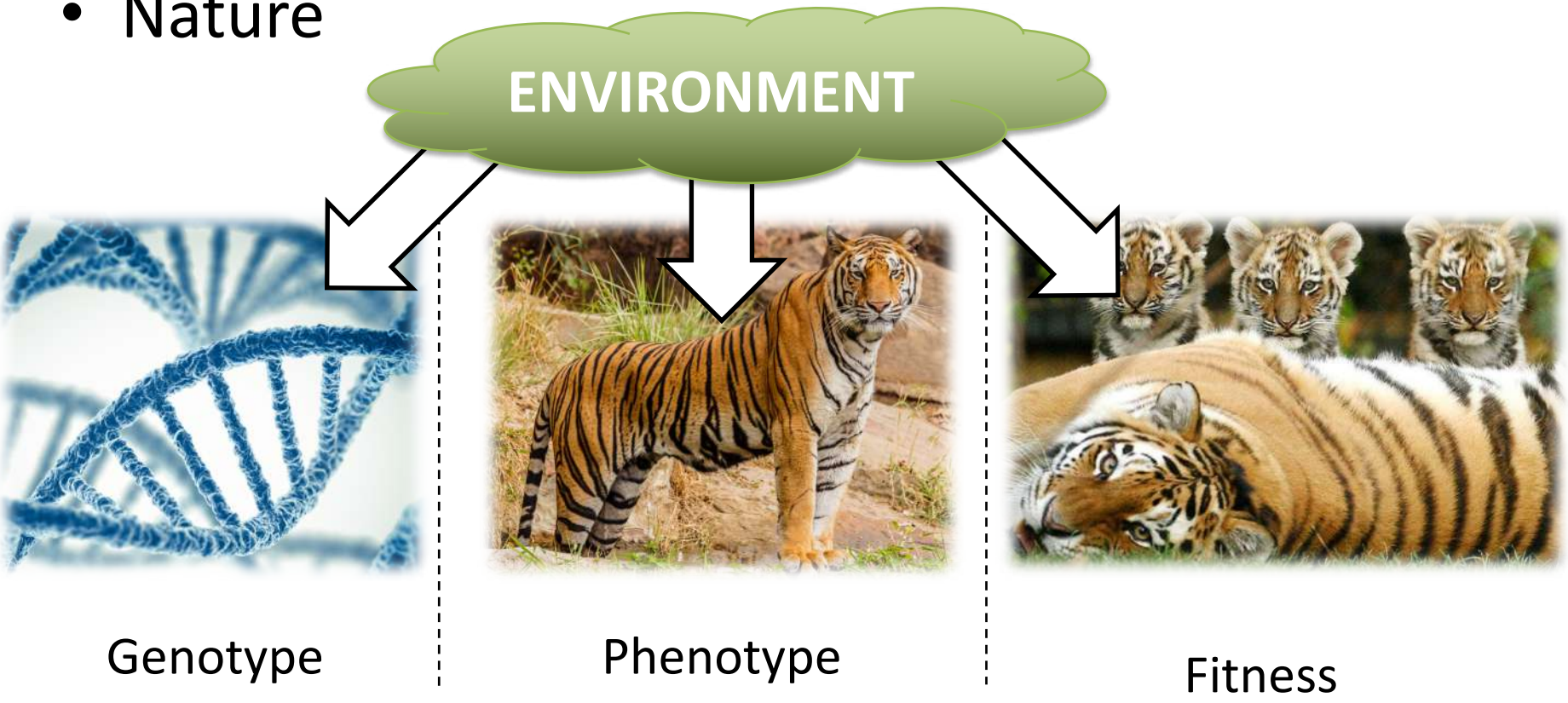
Complex Genomes

- Nature and EAs



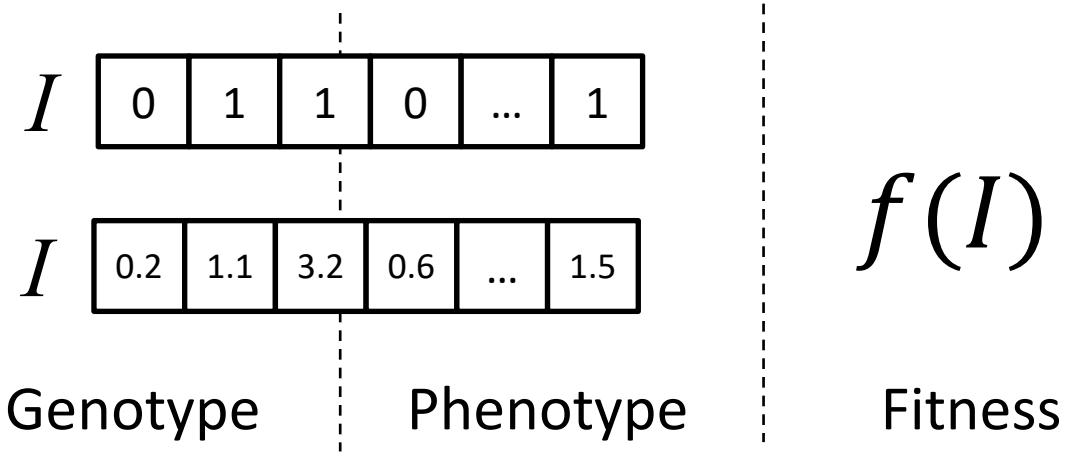
Complex Genomes

- Nature



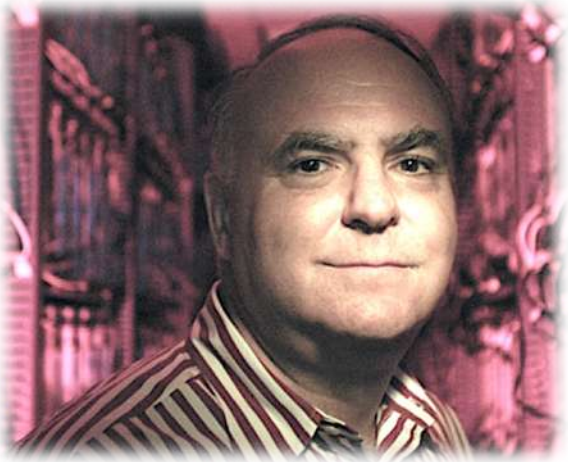
Complex Genomes

- Genetic Algorithms/Evolution Strategies…



Genetic Programming

- Genetic Programming
  - John Koza, 1992
  - Extend GA/ES to anything
  - Focus on computer programs



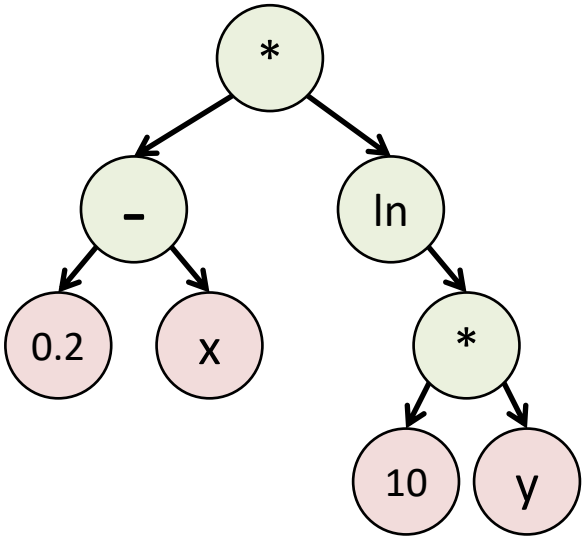
- Internal representation
  - Binary trees
  - Specialized mutations and crossovers

Genetic Programming

- General idea: EVOLVE ALL THE THINGS
  - If you can *describe* a candidate solution to a problem…
  - …and *variations* (e.g. mutations, crossovers)…
  - …and you can *define a fitness function*…
  - …EAs can explore the *space of all possible solutions*!

- It worked for *life on Earth*!
  - DNA is pretty complicated
  - Genome doesn't need to be *just numbers*

Genetic Programming

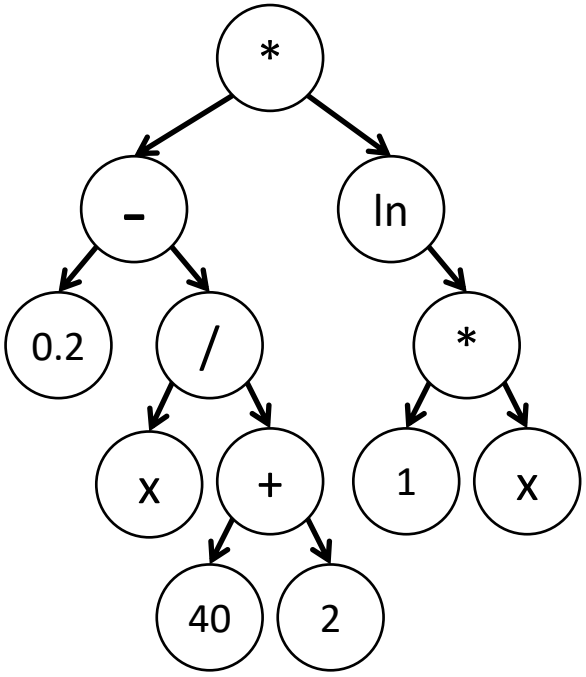


Operators: +, -, *, /, ln…

Terminals: reals, ints, vars, …

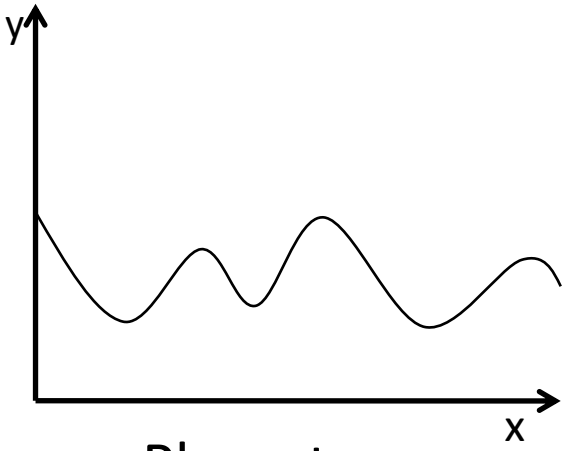
$$f(x, y) = (0.2 - x) * \ln(10 * y)$$

Symbolic Regression



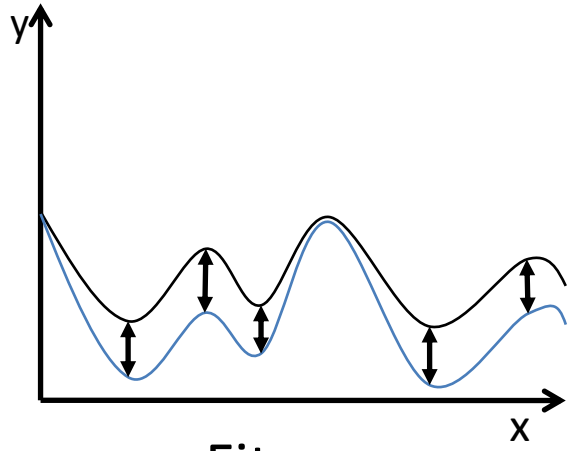
Genotype

$f(x) = [0.2 - (x/42)] * \ln(x)$



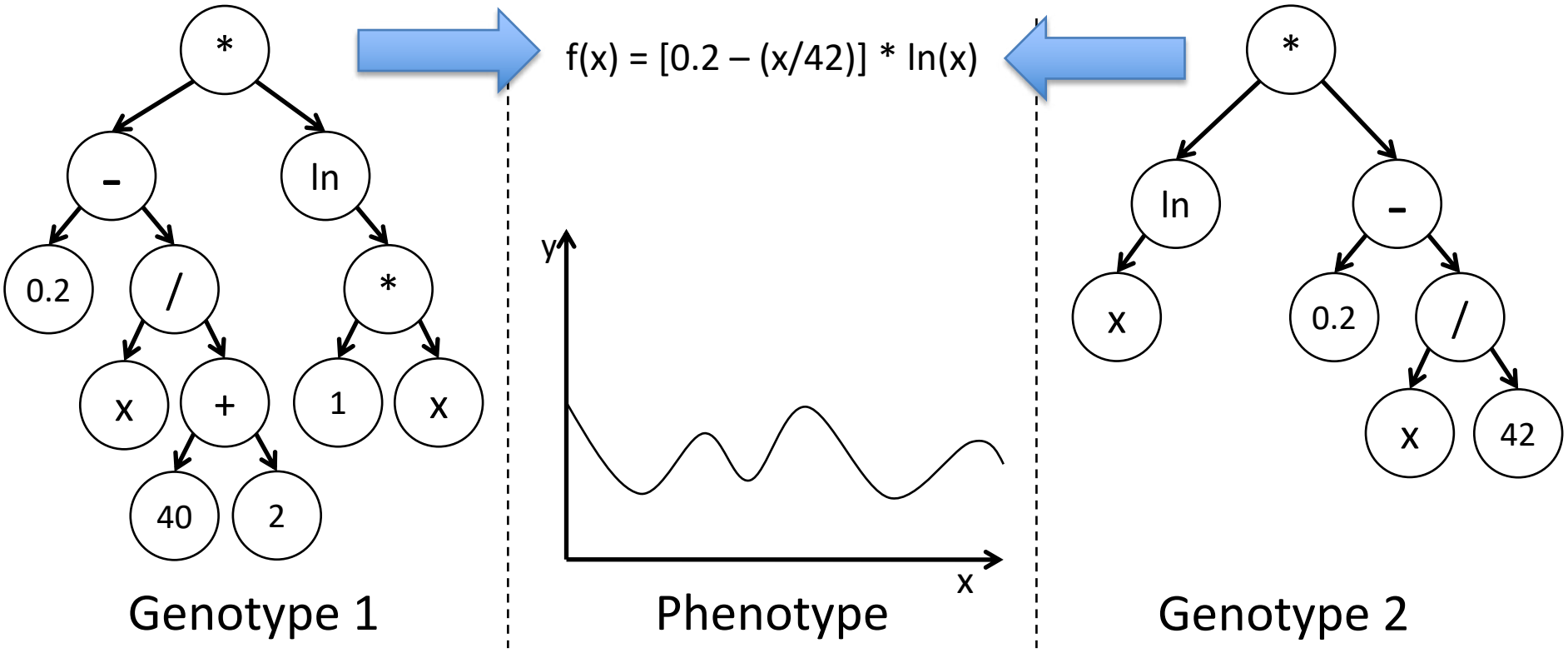
Phenotype

Fitness = $\sum_{i=0}^{N} \text{abs}(f(xi) - g(xi))$

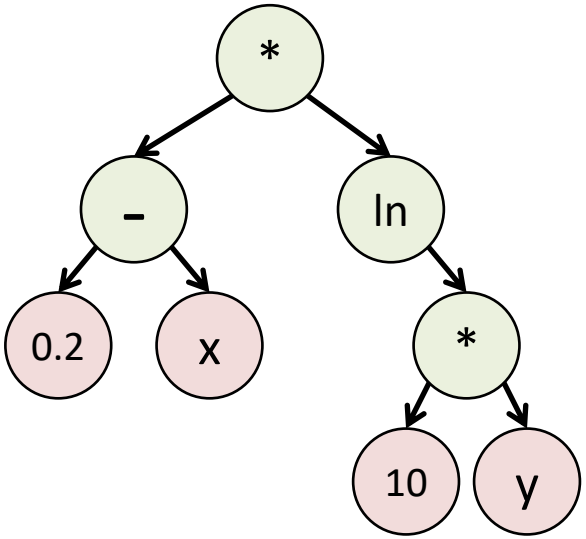


Fitness

Symbolic Regression



Genetic Programming

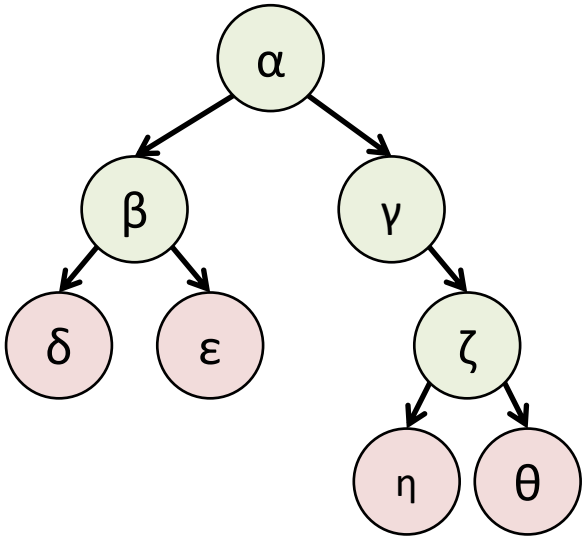


Operators: +, -, *, /, ln…

Terminals: reals, ints, vars, …

$$f(x, y) = (0.2 - x) * \ln(10 * y)$$

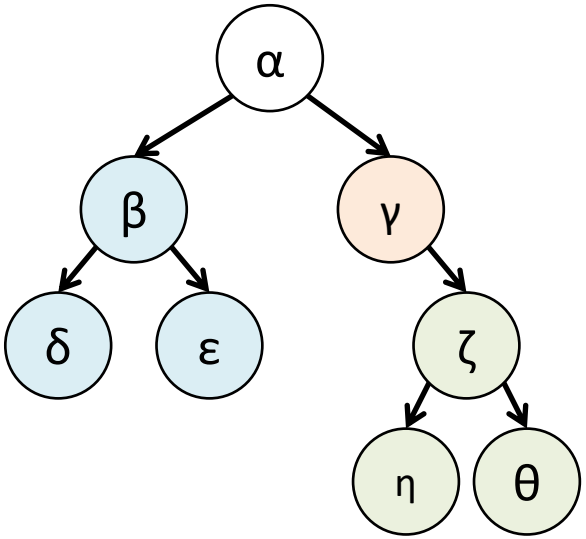
Genetic Programming



Operators: α, β, γ, ζ, λ, π, ς, …

Terminals: δ, ε, η, θ, ρ, σ, τ, …

Genetic Programming



Genotype

$I = \alpha($
$\beta(\delta, \varepsilon),$
$\gamma(\zeta(\eta, \theta))$
$)$

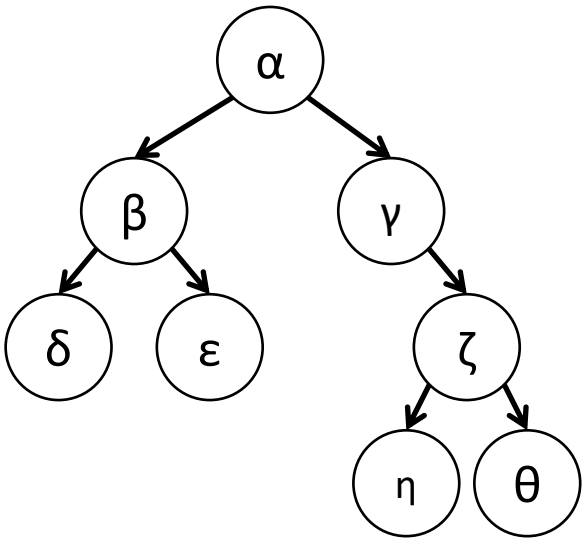
Phenotype

$f(I)$

Fitness

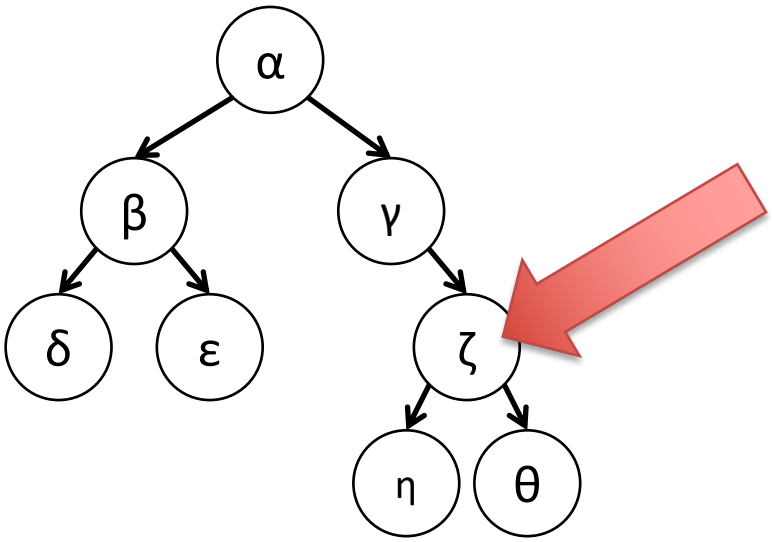
Genetic Programming

- Mutation(s)



Genetic Programming

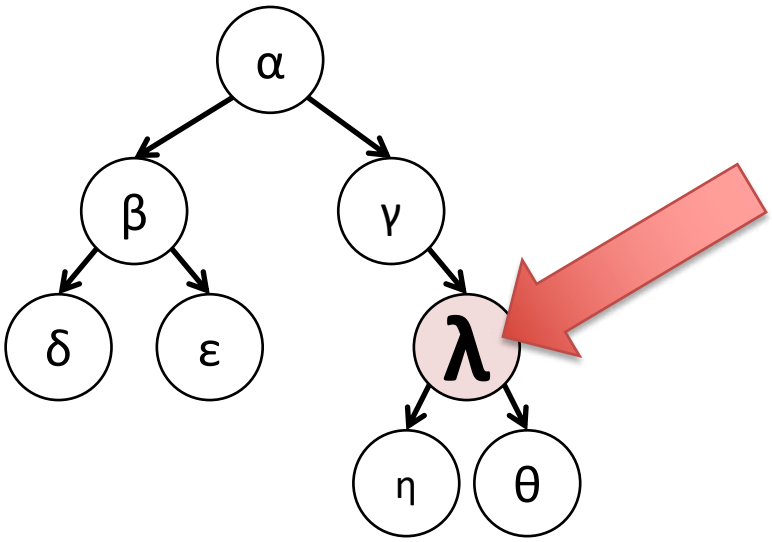
- Mutation(s)



POINT MUTATION

Genetic Programming

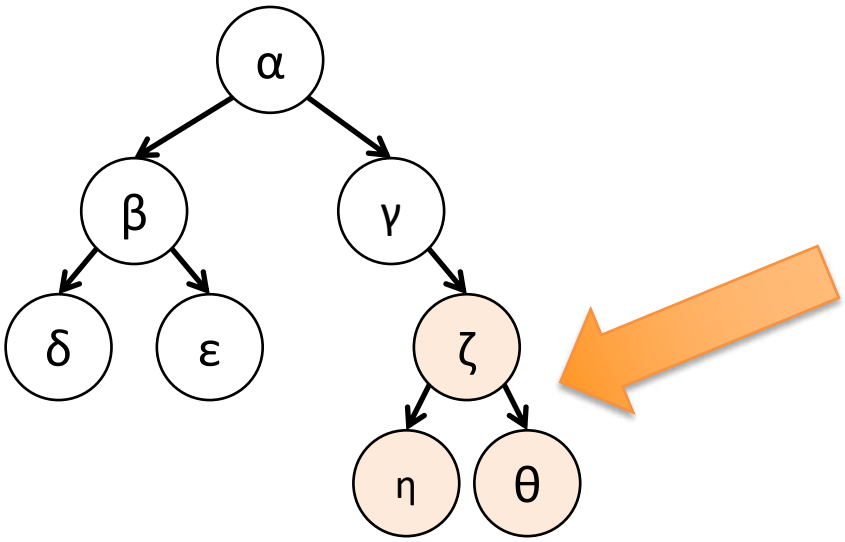
- Mutation(s)



POINT MUTATION

Genetic Programming

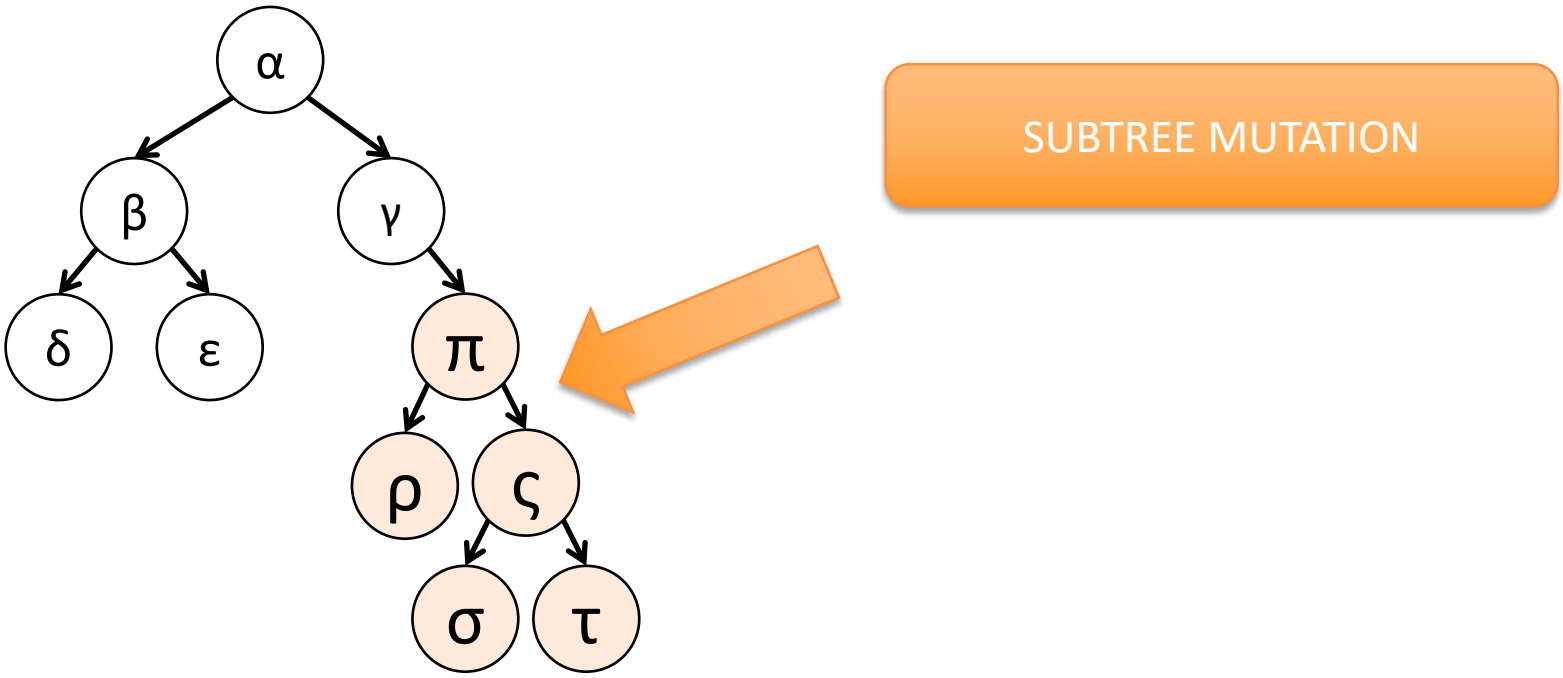
- Mutation(s)



SUBTREE MUTATION

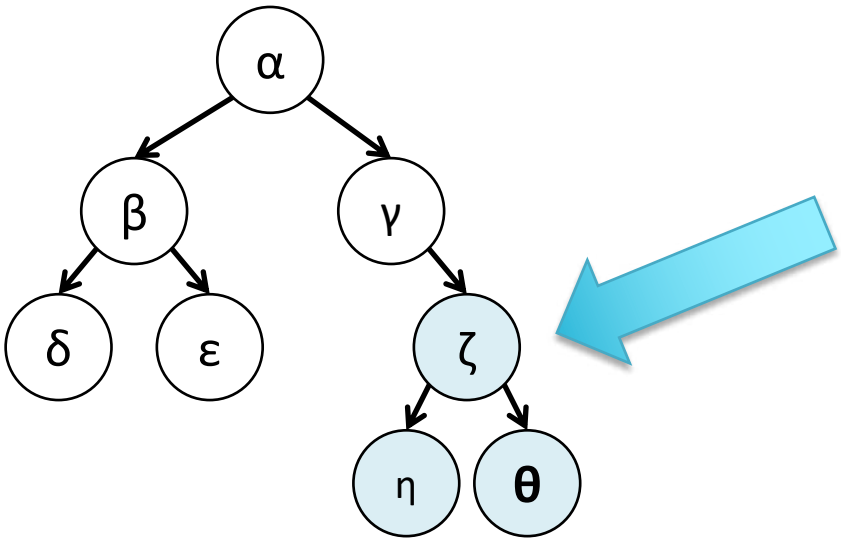
Genetic Programming

- Mutation(s)



Genetic Programming

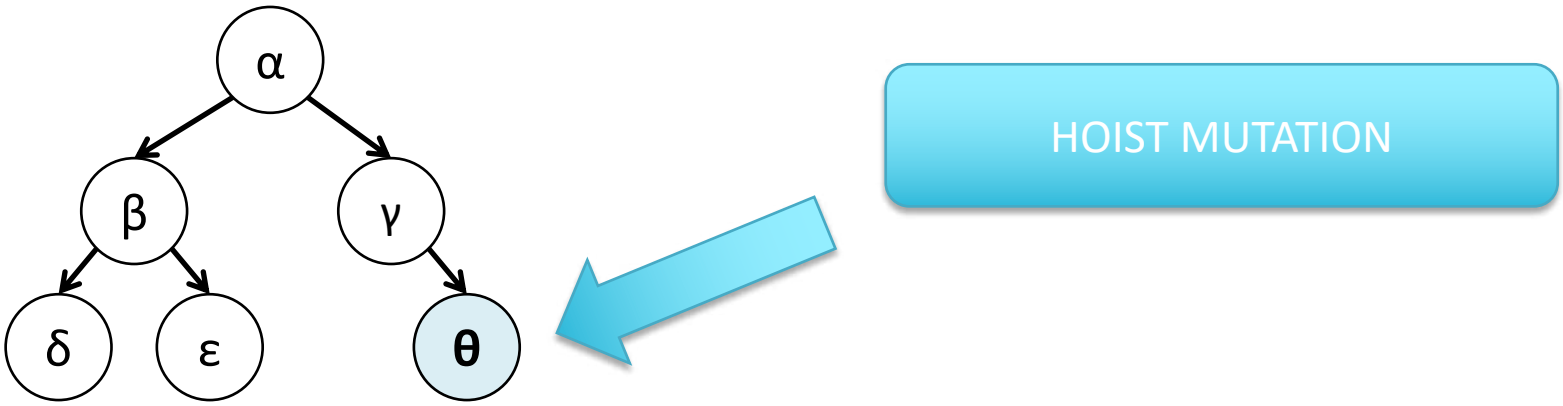
- Mutation(s)



HOIST MUTATION

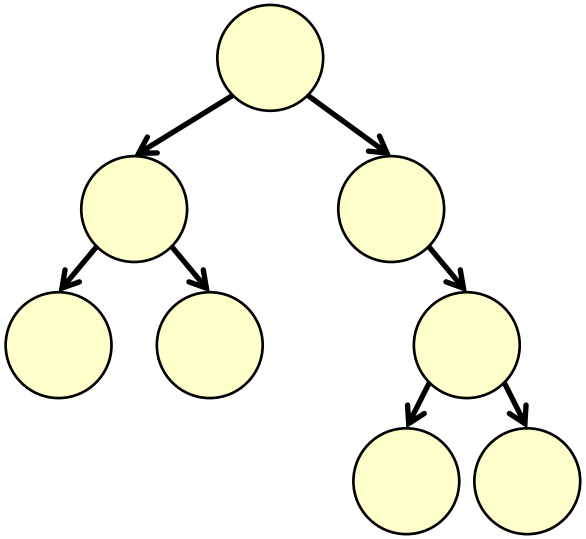
Genetic Programming

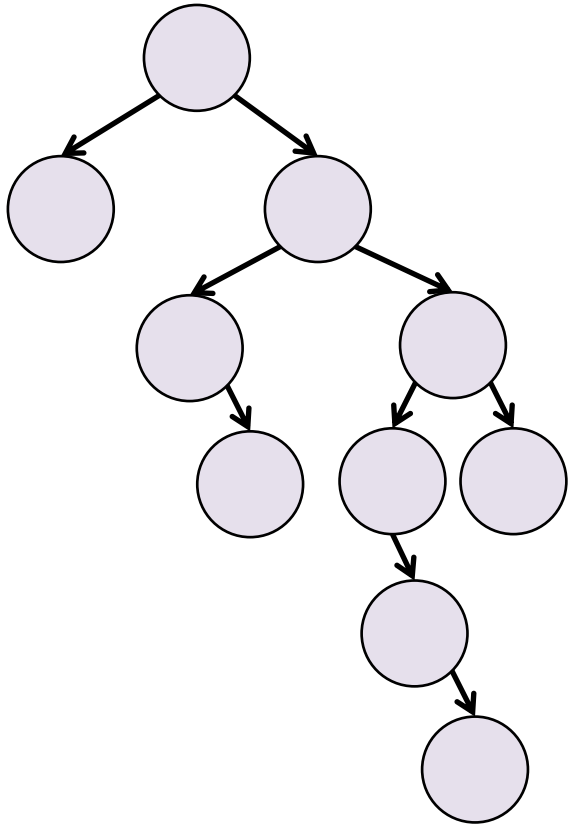
- Mutation(s)



Genetic Programming

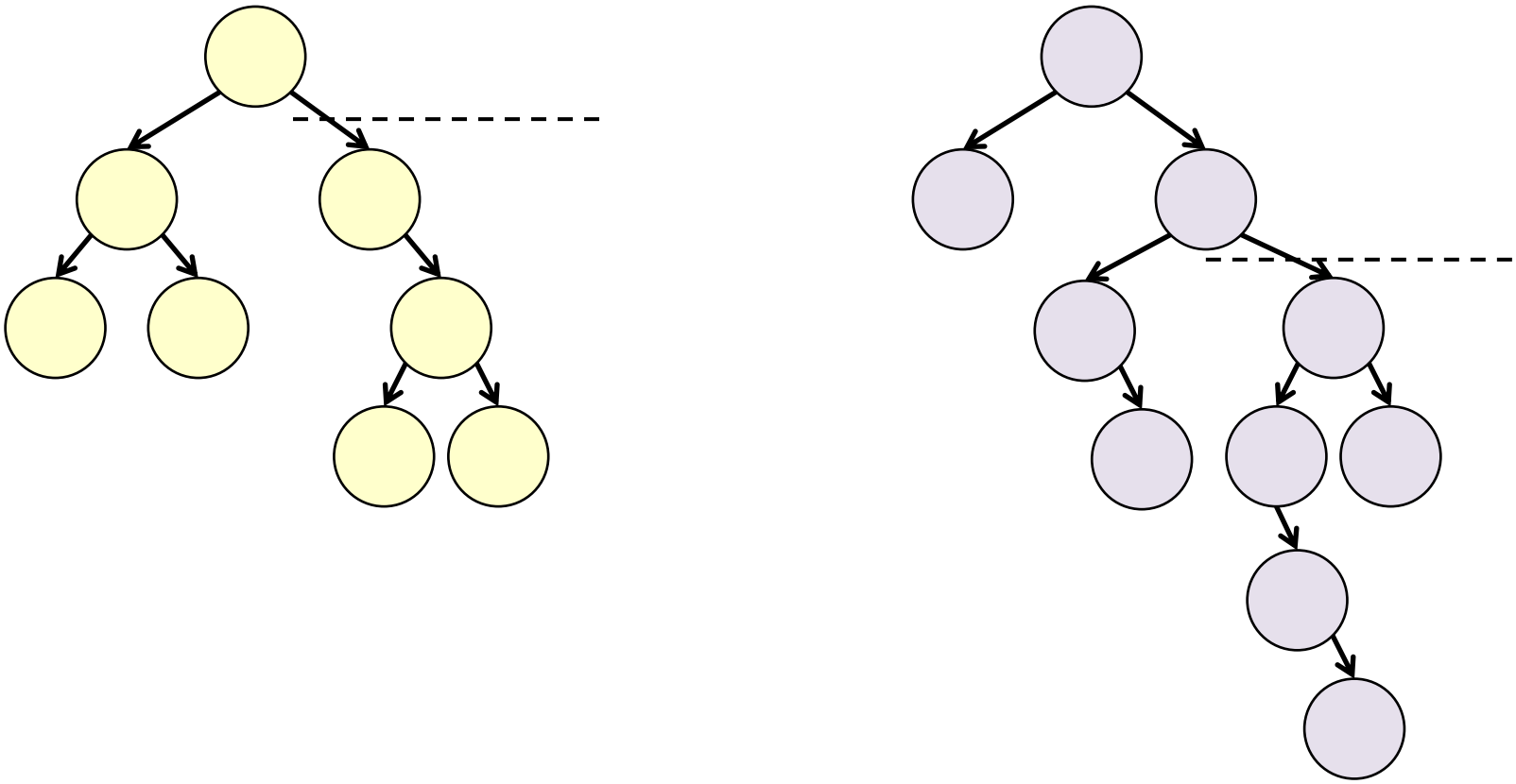
- Crossover(s)





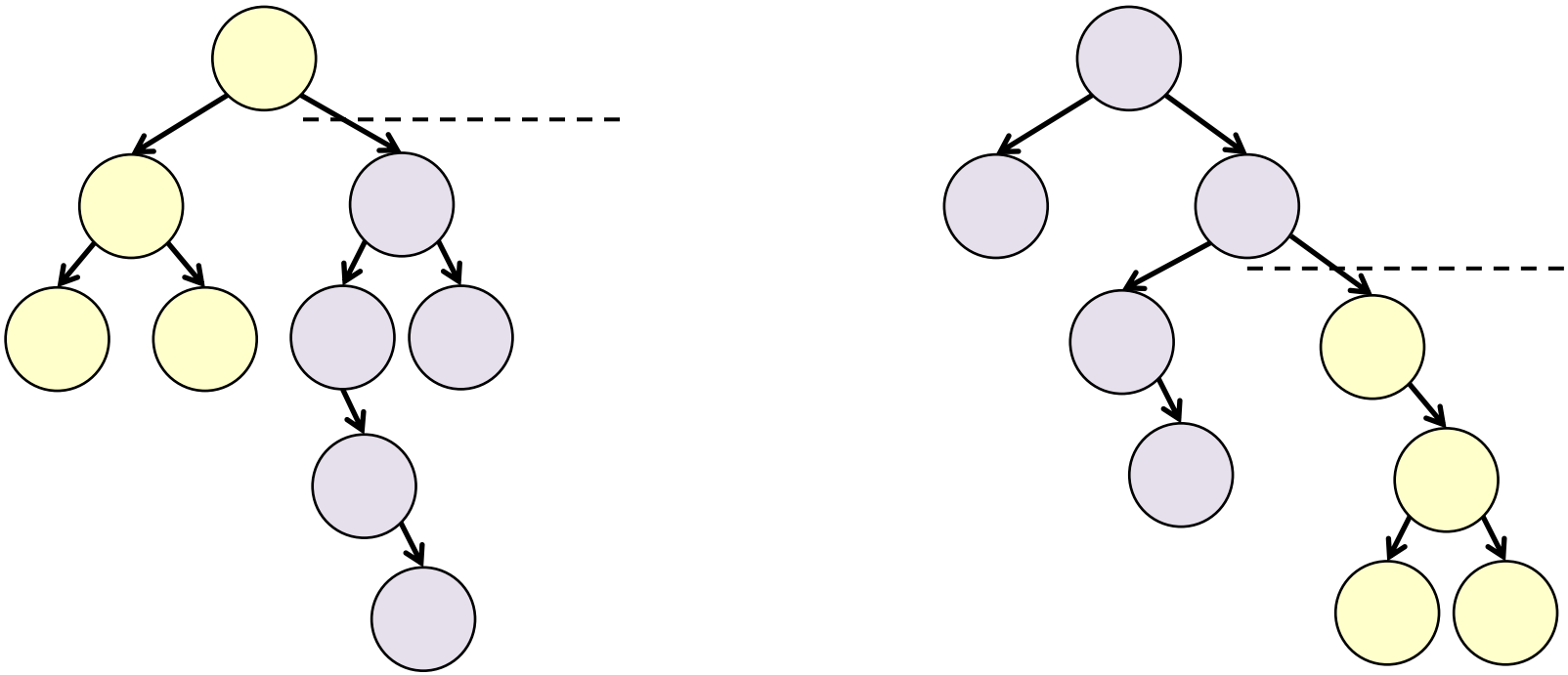
Genetic Programming

- Crossover(s)



Genetic Programming

- Crossover(s)



Genetic Programming: Issues

- Issues
  - Bloating
  - Introns
  - Destructive recombination

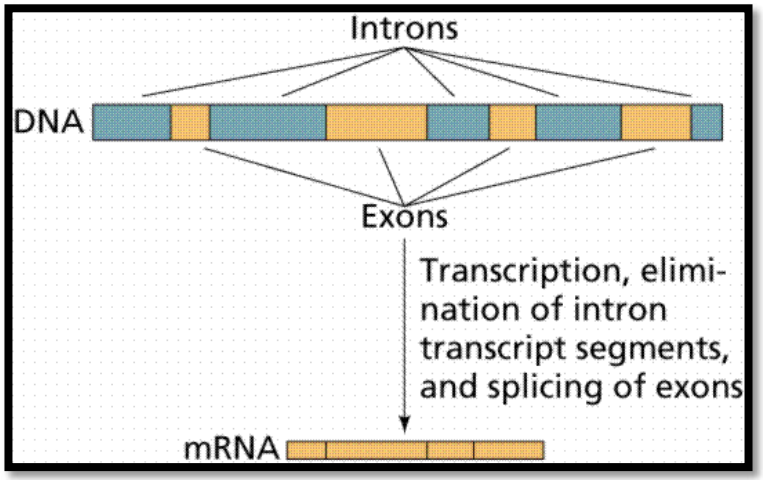
Genetic Programming: Bloating

- Over the generations
  - Individuals tend to increase in size…
  - …with no benefit on their fitness!
- Solutions
  - Penalize large trees (penalty coefficient)
  - *Fitness hole* (randomly, tournament selection on other criteria, such as size or diversity)
  - Simplify (problem-dependent)

Genetic Programming: Introns

- "Useless" parts inside individuals
  - Non-coding genome (y+1-1/1…)
  - Increase in size

- Are they REALLY useless?
  - Introns exist in natural DNA
  - "Protect" important code from destructive recombination



GP: Destructive Recombination

- Issue with crossovers
  - Important information is destroyed
  - How to choose a proper cut point?

- Heuristic crossover (problem-dependent)
- Ignore the problem (hope introns solve it)

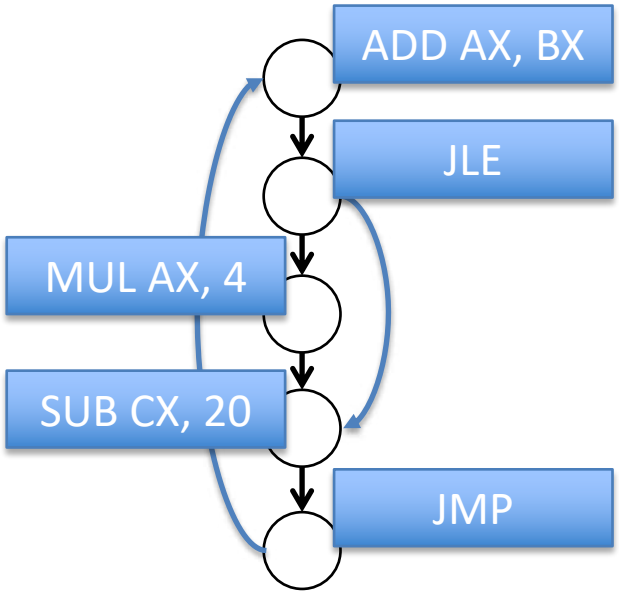
Genetic Programming

- What are we doing?
- Blending optimization and machine learning?
  - If your candidate solution is a *model*…
  - …then you are (arguably) doing *machine learning*!

- Terminology is still *uncertain*

Linear Genetic Programming

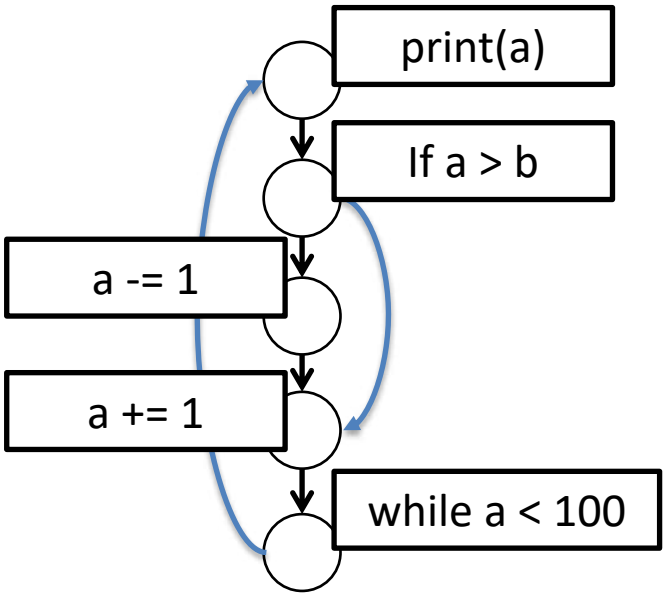
- Evolving linear graphs
- Used for evolving computer programs
- Backward/forward arcs interpreted as jumps




```
label1: ADD AX, BX
JLE label2
MUL AX, 4
label2: SUB CX, 20
JMP label1
```

Linear Genetic Programming

- Evolving linear graphs
- Used for evolving computer programs
- Backward/forward arcs interpreted as jumps



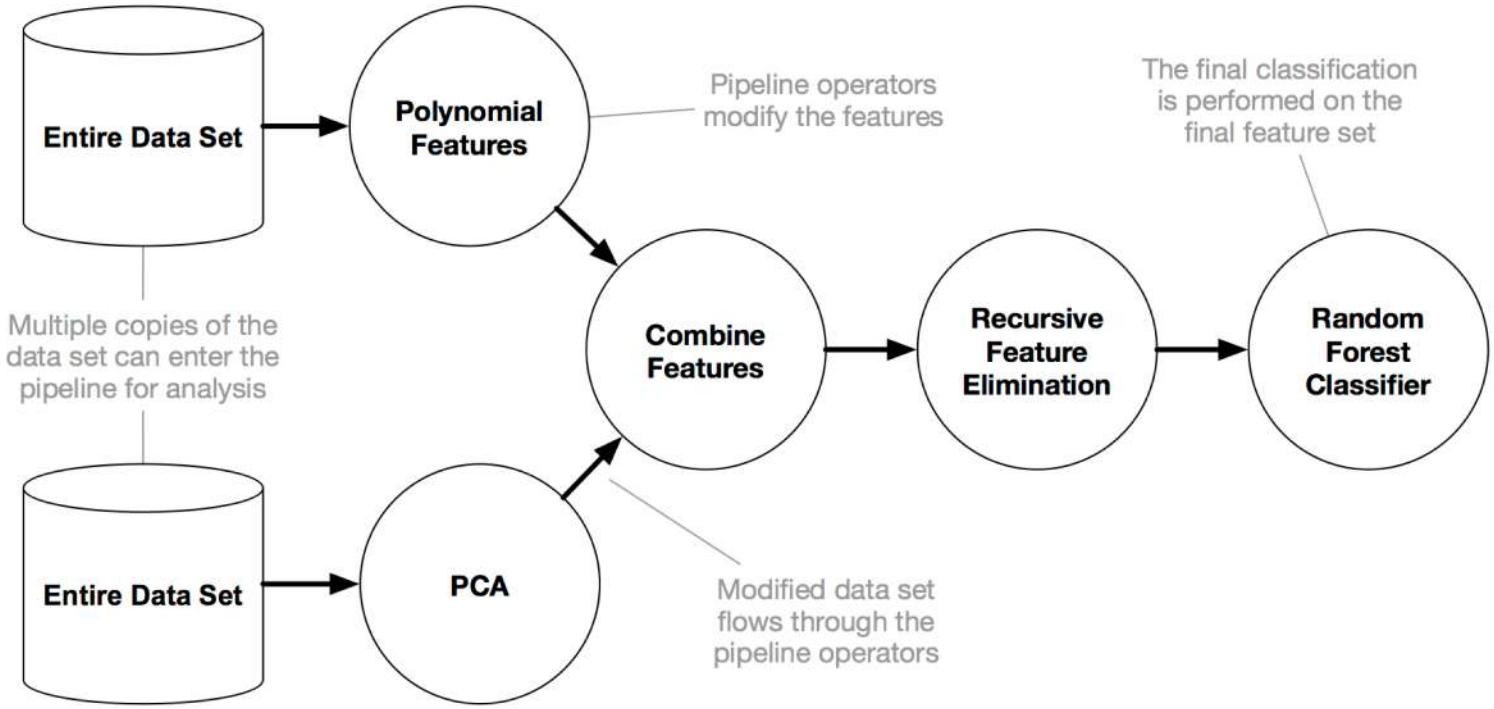

```
while a < 100 :
    print(a)
    if a > b :
        a += 1
    else :
        a -= 1
```

Evolving AIs

- Real-time strategy (RTS)
  - Planet Wars (Google)
  - StarCraft
  - Student StarCraft AI Tournament
- Trade-off
  - ANNs are better
  - You can read GP trees

Evolving ML Workflows

- TPOT
  - https://epistasislab.github.io/tpot/



Evolving Neural Networks

- Design of Artificial Neural Networks (ANNs)
  - Most are built by copying literature
  - Trial and error, human-designed
  - But can we find something better?
- Neuroevolution!
  - The concept exists since ~1995
  - Wider adoption since 2016, with success of Deep Learning (improved ANNs)

Software Testing

- Generate sequences of inputs to find bugs
  - Motorola tests mobile phones
  - Facebook tests graphical user interfaces (**Sapienz**)
- Fitness landscape seems impossible!
  - Few points with bugs, everything else is 0
  - Smoothing landscape with domain knowledge
  - Reward individuals that explore functions

Genetic Improvement

- Automatic correction of software bugs
- Individual: series of code modifications

- Fitness
  - A series of test cases
  - They still have to work

Comment line 52
Swap lines 3 and 22
Change variables lines 42 and 11
…

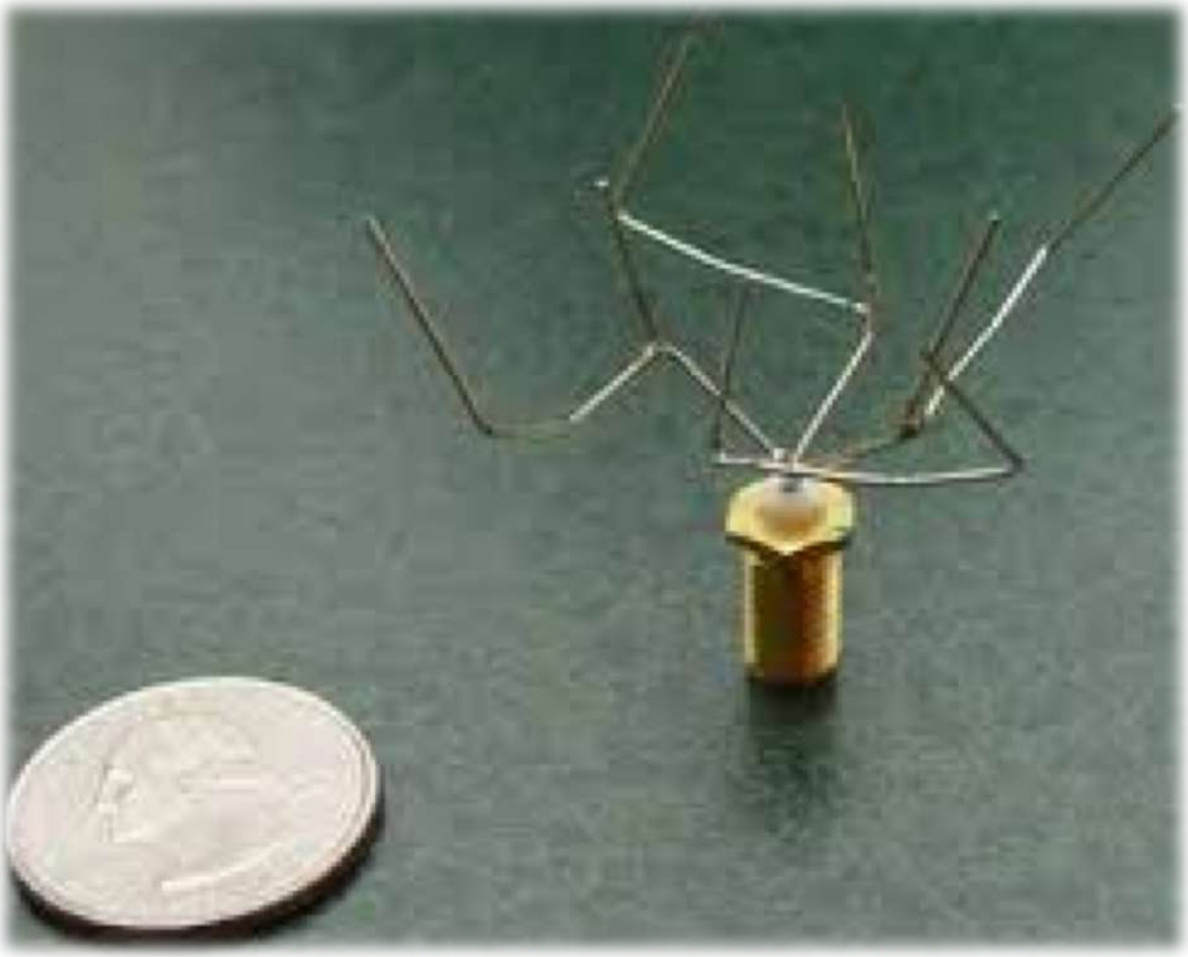
Genetic Improvement

- But does it *really* work?
  - Aren't EAs introducing other bugs?
  - Aren't HUMANS introducing other bugs?
  - In the end, you just need to be *as good as* the average programmer, and you save time
  - Still experimental

Langdon, William B. **Genetically Improved Software**

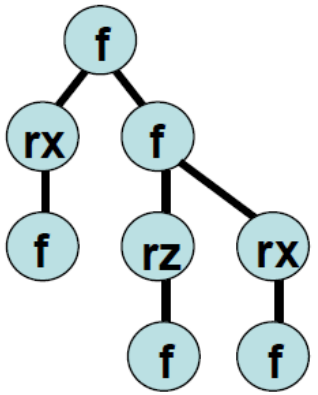
Justyna Petke and Saemundur O. Haraldsson and Mark Harman and William B. Langdon and David R. White and John R. Woodward. **Genetic Improvement of Software: a Comprehensive Survey**

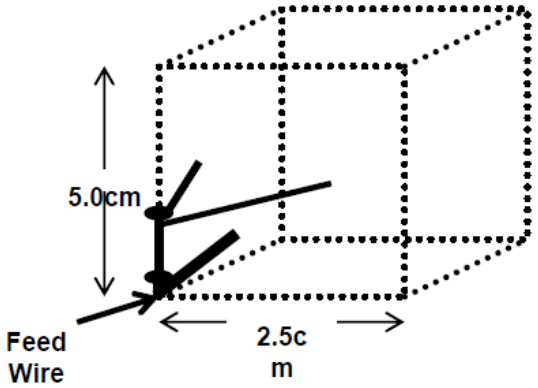
Evolve Everything



Evolve Everything: Antennas

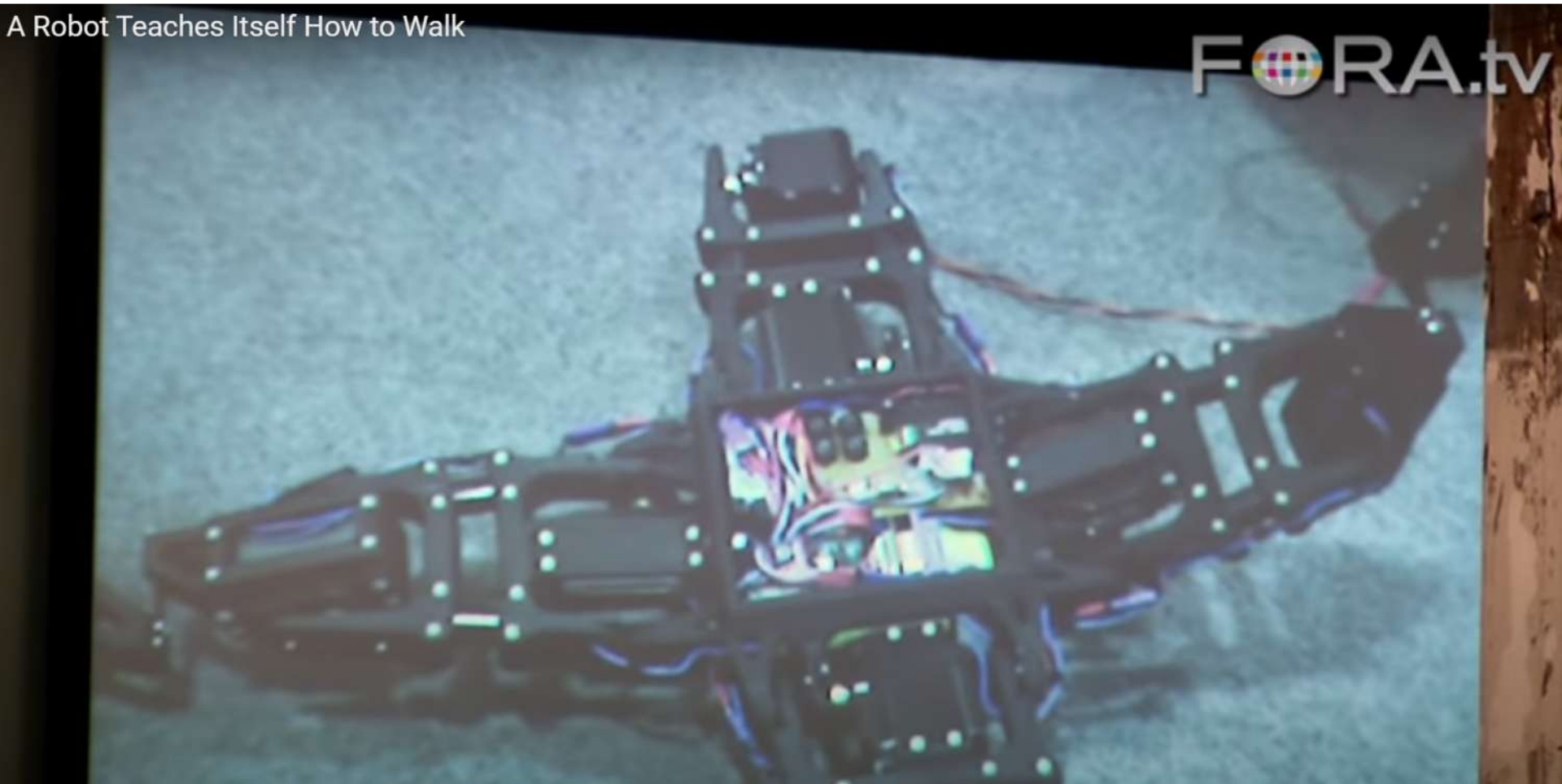
- Design of antennas for satellite ST5 (2006)
- Lots of constraints: weight, size, efficiency…
- Genome
  - Forward (length, radius)
  - Rotate_x (angle)
  - Rotate_y (angle)
  - Rotate_z (angle)
- It worked!





Evolve Everything: Robot Movement

https://www.youtube.com/watch?v=iNL5-0_T1D0



Evolve Everything: Movement

Evolved Electrophysiological Soft Robots

Nick Cheney[1]
Jeff Clune[2]
Hod Lipson[1]

[1] Creative Machines Lab, Cornell University
[2] Evolving AI Lab, University of Wyoming