W. B. Langdon, UCL, London

email W.Langdon@cs.ucl.ac.uk

Genetic Programming:
On the **Programming** of Computers by Natural Selection — John Koza

Algorithms + **Data Structures** = Programs — Niklaus Wirth

Using GP to Evolve **Data Structures**

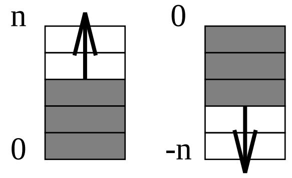
Why Evolve Data structures

- Programs require memory
- Can't specify memory $\Rightarrow$
  evolve how memory is used

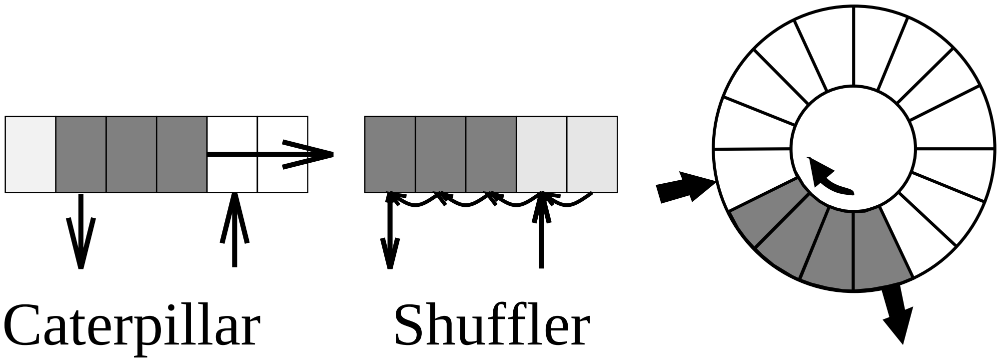
- Data structures more powerful
- and less brittle
- $\leadsto$ help evolution

Data Structures Evolved

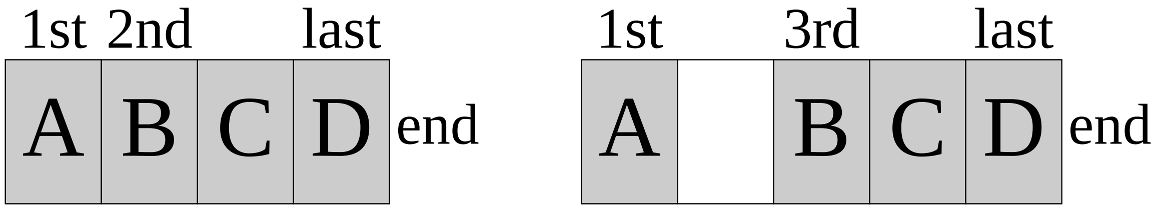
- Stack: Work up or down

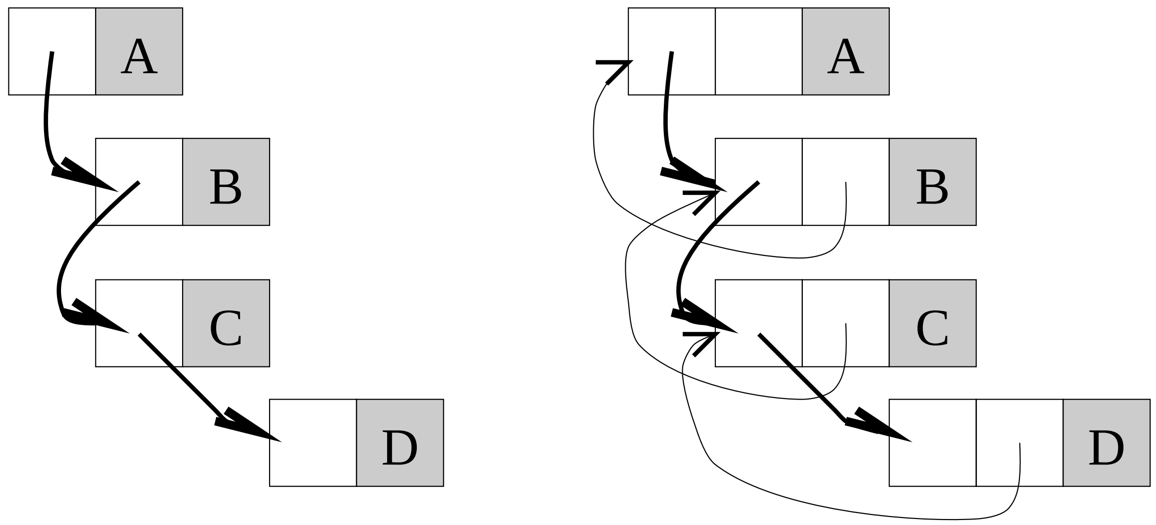


- Queue: Caterpillar, shuffler, circular



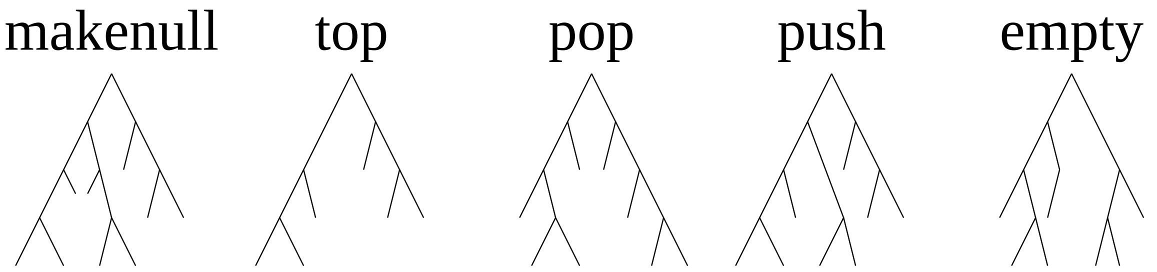
- List: Array, linked chain, double chain





Representation

- Data structure = memory + ops
- Individual is trial data structure
- Each operation has own tree

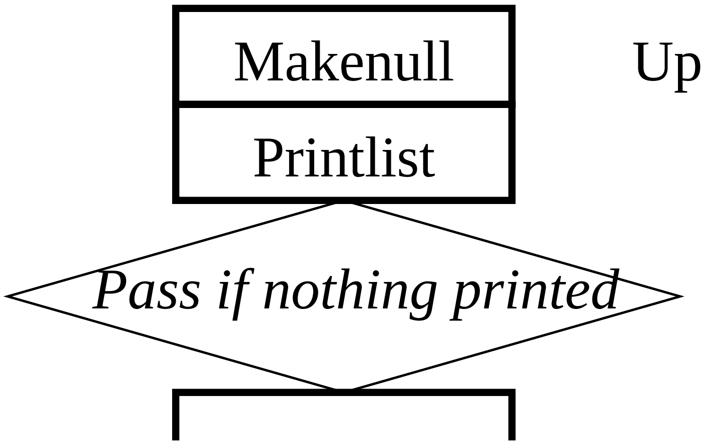


One individual within the population
(Stack - 5 trees)

- Arithmetic, inputs, constants, adfs,
  Indexed memory

Fitness function

- Test interface NOT internals
- Test operations in small groups



Up to 575

- Is operation's answer correct? Consistent with other answers?
- Delayed payoff

  E.g. Push no reward for saving input, until Pop

  But such programs do evolve!

Multiple Solutions
Do you have this problem?

- Cross Shuffler with Circular expect poor offspring
- Population needs to agree
  But converge $\Rightarrow$ lost diversity
- Population contain multiple solutions:
  Demes, Niches, IFLTE, NROG
- Mechanism to breed like with like
  genotype, phenotype or both?
- $\leadsto$ Multiple species,
  each solving problem differently

Summary

- Benefits of structured use of memory
- How simple data structures can be evolved
- General problem of multiple solutions

- To be general GP must solve any type of problem
- $\Rightarrow$ evolve any type of program
- Can't ignore data!

W. B. Langdon, UCL, London

email W.Langdon@cs.ucl.ac.uk
WWW http://www.cs.ucl.ac.uk/staff/W.Langdon/

Try it yourself!

ftp site `cs.ucl.ac.uk`		
directory	file	contents
`genetic/gp-code`	`GPdata-1-dec-94.tar.Z`	**C++ code**
`genetic/gp-code`	`gpquickie/*`	Interactive environment
`genetic`	`simple-gp.c`	generate C code
`genetic`	`park-miller.cc`	random numbers
`genetic/biblio`	`gp-bibliography.bib`	
`genetic/papers`		