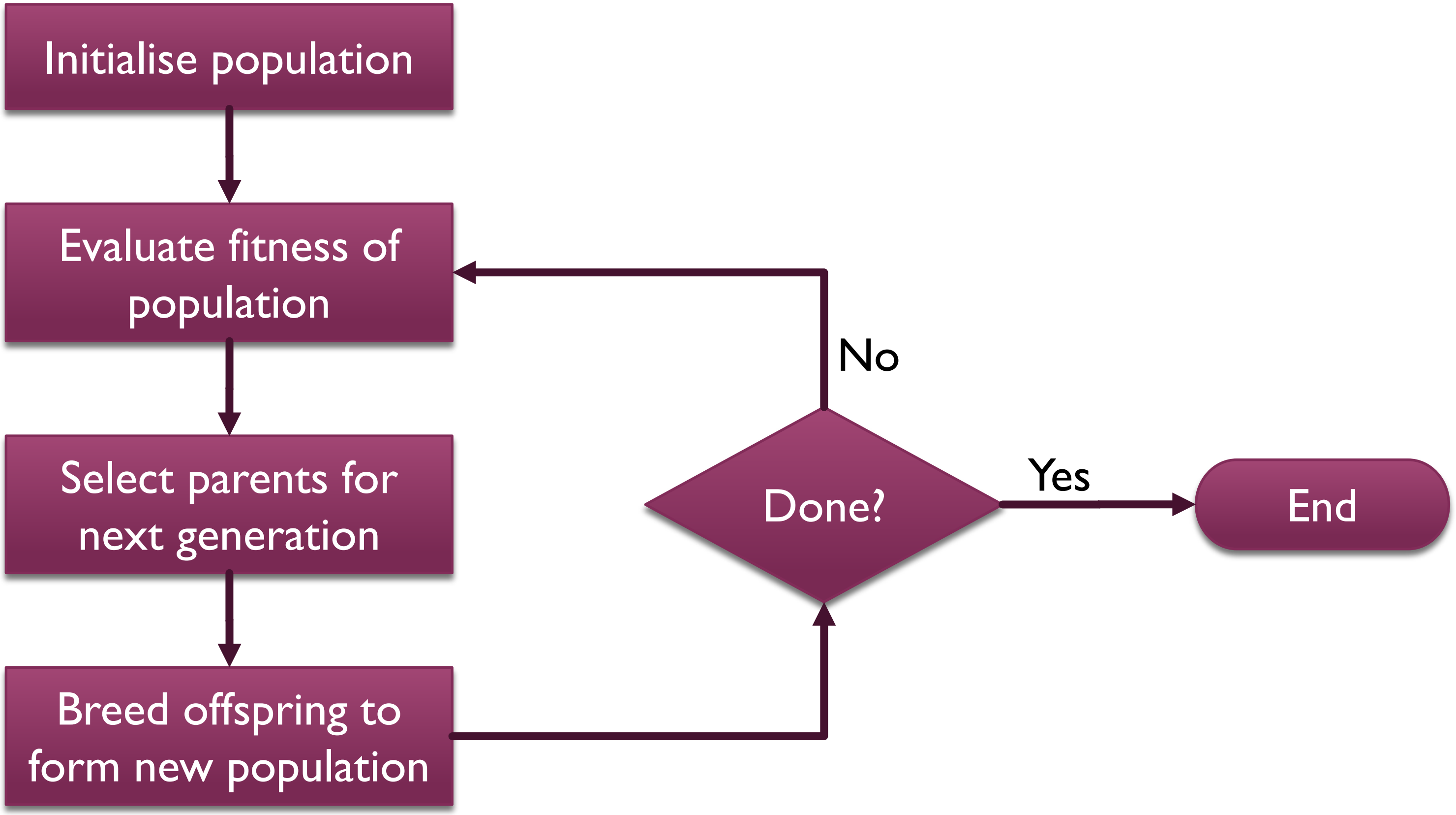
11: GENETIC PROGRAMMING

COMP704: MACHINE LEARNING

EVOLUTIONARY ALGORITHMS (EAS)
REVISION FROM LECTURE 7

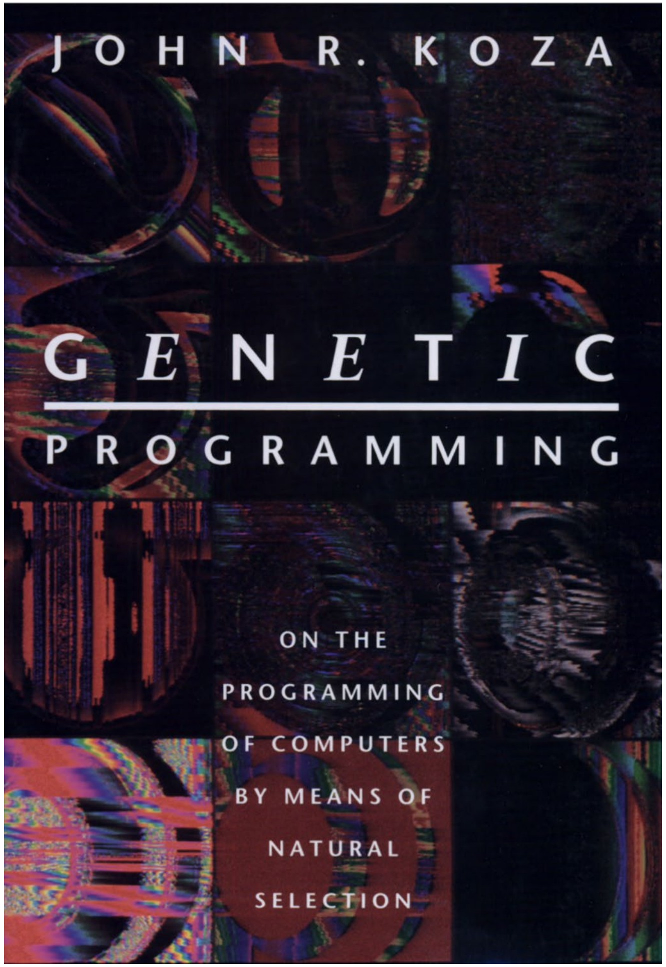
- **Inspired by** biological evolution
- A family of **population-based search** algorithms
- **Fittest** individuals are used to produce new individuals via **mutation** and **crossover**
- **Genotype**: search space representation
- **Phenotype**: solution space representation

BASIC EVOLUTIONARY ALGORITHM
REVISION FROM LECTURE 7


Initialise population
Evaluate fitness of population
Select parents for next generation
Breed offspring to form new population
Done?
No
Yes
End

GENETIC PROGRAMMING (GP)

- A term given to EA approaches where the **genotype** is a **computer program**
- The **phenotype** is the result of **executing** the program

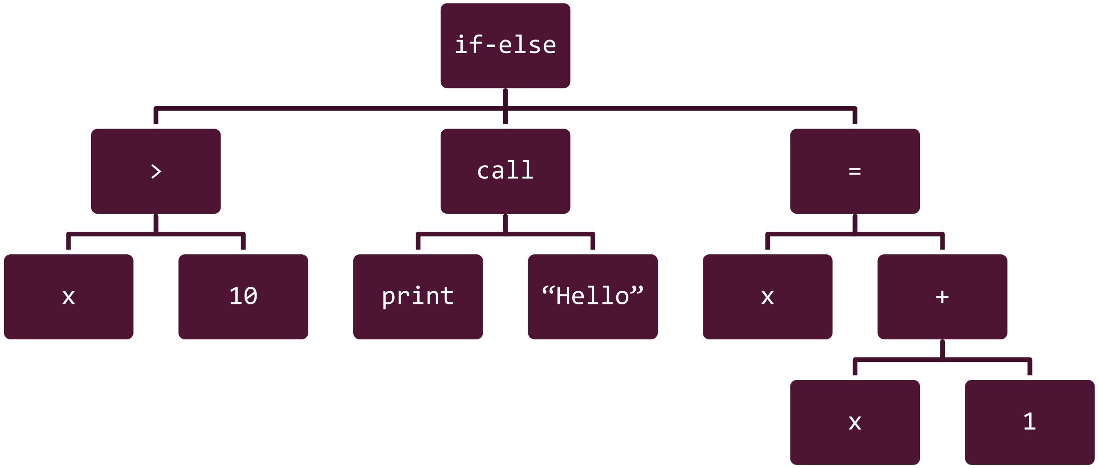

JOHN R. KOZA
GENETIC
PROGRAMMING
ON THE PROGRAMMING OF COMPUTERS BY MEANS OF NATURAL SELECTION

GENOTYPE REPRESENTATION

- We could represent a program as a sequence of ASCII characters…
- … but the vast majority of individuals would **not** correspond to **syntactically correct** programs
- Can’t run the program → no phenotype!
- We need a representation where the genotype is certain (or at least very likely) to map to a valid phenotype
- Several options – the classic one is a **tree representation**

PROGRAMS AS TREES

```
if (x > 10)
  print(“Hello”)
else
  x = x + 1
```



ABSTRACT SYNTAX TREES

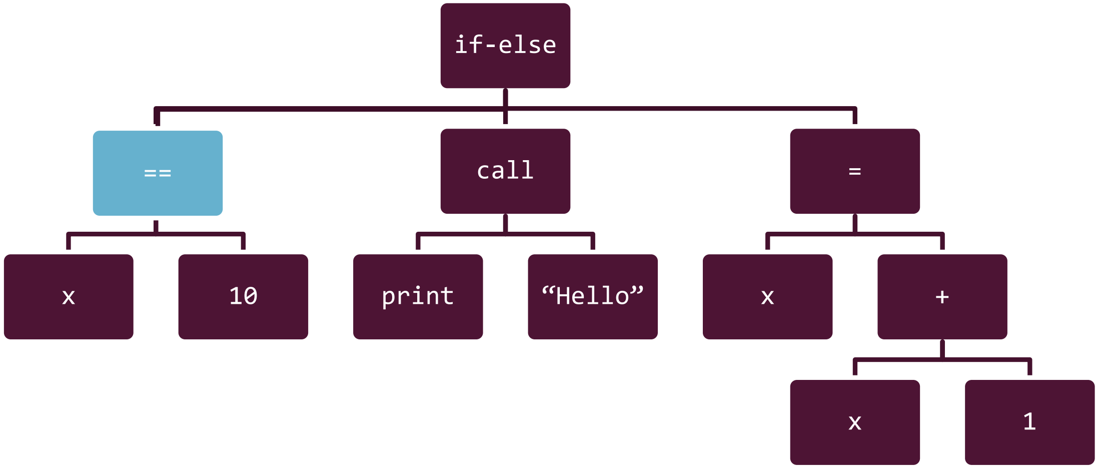
- Representing programs as trees is nothing unusual – it's used extensively inside compilers/interpreters
- Abstract Syntax Trees (ASTs)
- Language syntax has been **abstracted** away
- A valid tree is probably a valid program
- AST can be executed as-is, or translated back into source code

FITNESS EVALUATION

- Remember, **fitness** is a property of the **phenotype**
- In GP, the first step in measuring fitness is to **execute** the program
- Fitness is often measured based on the program’s **output**
- But could also measure **non-functional** properties e.g. speed, memory usage
- Fitness may be **multi-objective**

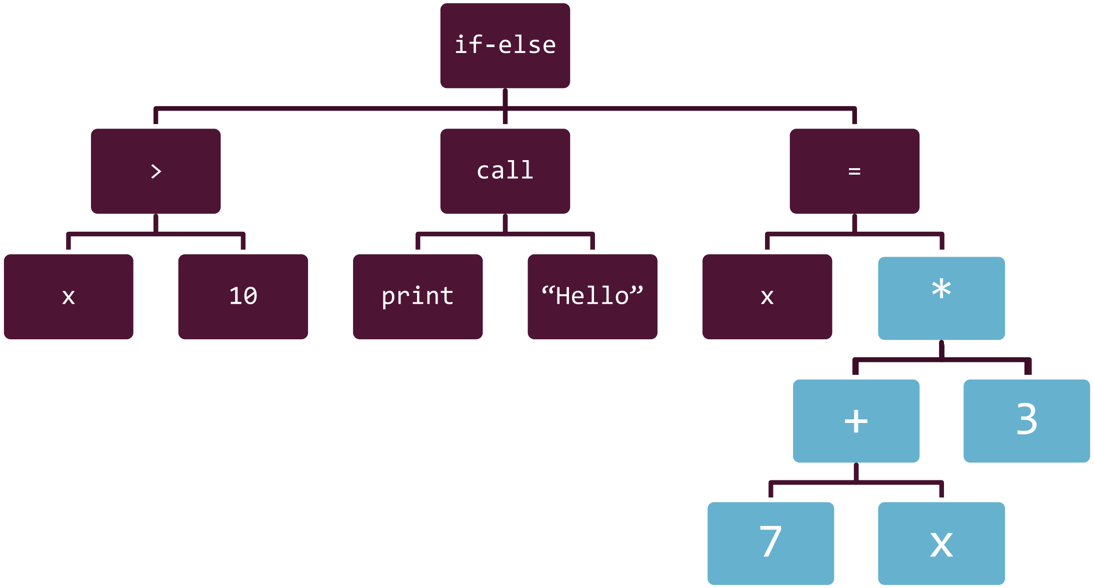
MUTATION

- Point mutation: replace a random node

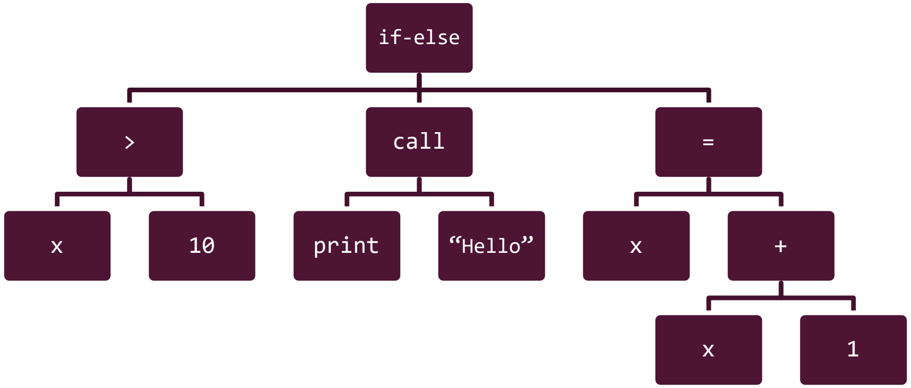


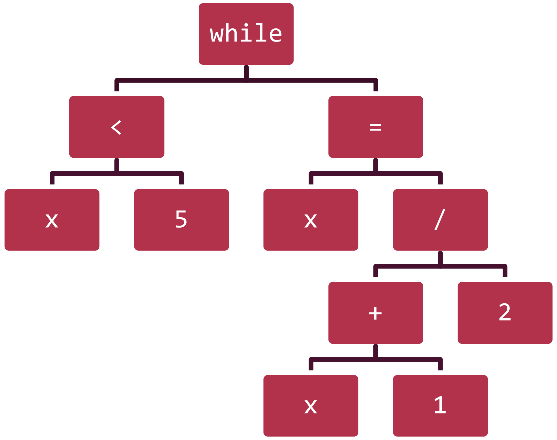
MUTATION

- Subtree mutation: replace a random subtree



CROSSOVER





BIOLOGICAL INSPIRATION

- The version of evolutionary algorithms we saw in week 7 take inspiration from biological genetics
  - Genotype is a fixed-length string of "DNA"
- GP moves away from this
  - Genotype is a variable-size tree structure
  - No real analogue of this in nature (that we know of)
- However GP still retains much of the rest of the biological inspiration behind EAs

WEAK VS STRONG TYPING

- Nodes generally correspond to **values** when the program is executed, therefore have a **type**
- **Weak typing**: the language rules are permissive about what type can go where in the tree
- **Strong typing**: nodes require their children to have specific types, and GP operators are careful to preserve type

DISRUPTION

- As you know, a small change in a program can have a huge impact on its execution!
- Genetic operators can be **disruptive**, destroying good individuals
- Much work on operators to minimise disruption and maintain goodness of solutions
- E.g. context preserving crossover
- E.g. Automatically Defined Functions (ADFs)

BLOAT

- GP operators can often allow trees to **grow** arbitrarily
- Often a lot of **bloat** resulting from "junk code"
  - Overly-convoluted code that could be simplified
  - Code that doesn't actually do anything, e.g. the `else` branch of an `if` that is always `true`, or a calculation that is eventually multiplied by zero

COMBATTING BLOAT

- Can put a hard limit on the size or depth of the tree
- Can modify GP operators to prefer smaller trees
- Can include tree size as part of the fitness measure, rewarding smaller trees
- However, bloat has similarities to **introns** in biology – genes which appear to do nothing, but are actually important
- Appears to be important for maintaining **diversity** in the population

CONCLUSION

- Genetic programming is a family of evolutionary algorithms where the genotype is a computer program
- Several possible genotype representations, but representations based on trees are popular
- As with all EAs, there is an art to making it work but it can be extremely powerful