

# **CSE/ECE 848**

## **Introduction to**

### **Evolutionary Computation**

**Module 5 - Lecture 23 - Part 1a (formerly 2 )**

## **From Genetic Programming**

### **to Genetic Improvement**

**Wolfgang Banzhaf, CSE**  
**John R. Koza Chair in Genetic Programming**

# Outline

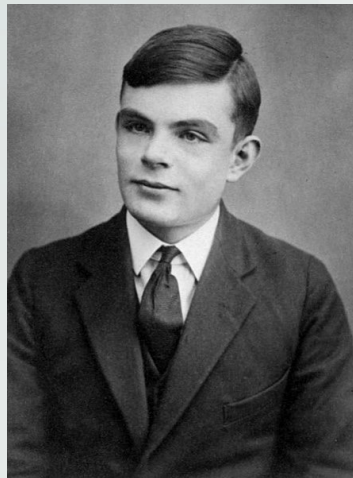
- From: Genetic Programming (GP)
- To: Genetic Improvement (GI)

# Genetic Programming - The Idea

“... the learning process may be regarded as a **search for a form of behaviour** which will satisfy the **teacher (or some other criterion)**. Since there is probably **a very large number** of satisfactory solutions the random method seems to be better than the systematic. It should be noticed that it is used in the analogous **process of evolution.**”

—Alan Turing, Computer Machinery and Intelligence (1950)

Alan Turing (1912-1958)



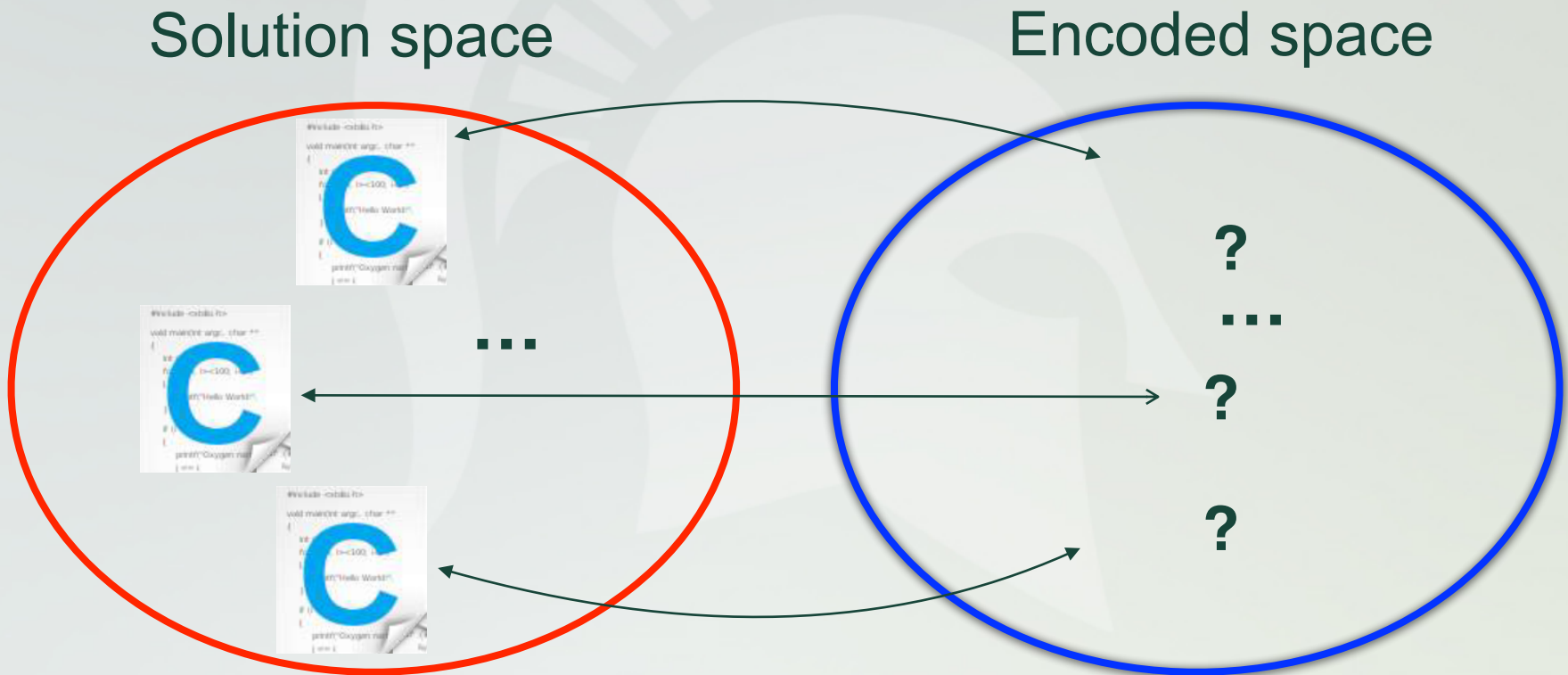
## Turing's Thoughts

# Genetic Programming

- Genetic Programming
  - Search for a form of behaviour => Search for programs
  - Satisfy the teacher (or some other criterion) => Fitness function
  - In the analogous process of evolution => Mimic nature evolution

# What is Genetic Programming

- Each solution in the solution space is a **computer program**



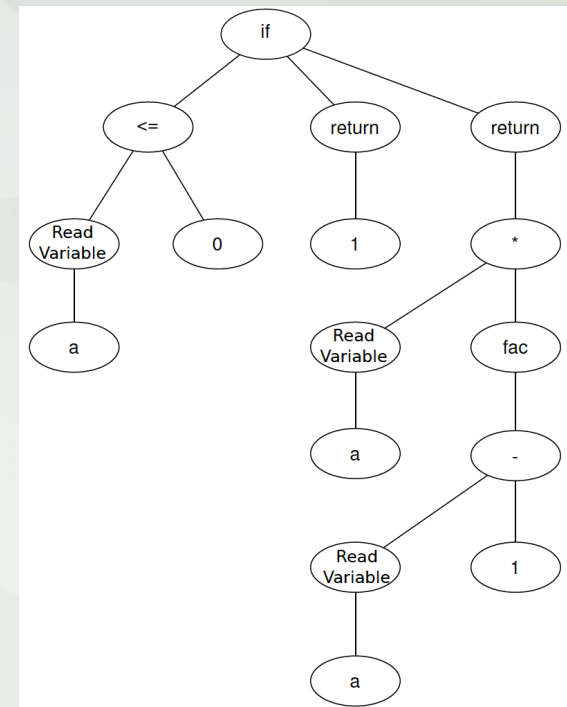
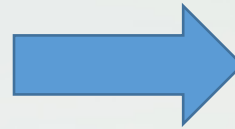
# Genetic Representation

- Encoding of Computer Programs

```

int Factorial(int a)
{
    if (a <= 0)
        return 1;
    else
        return (a * Factorial(a-1));
}
  
```

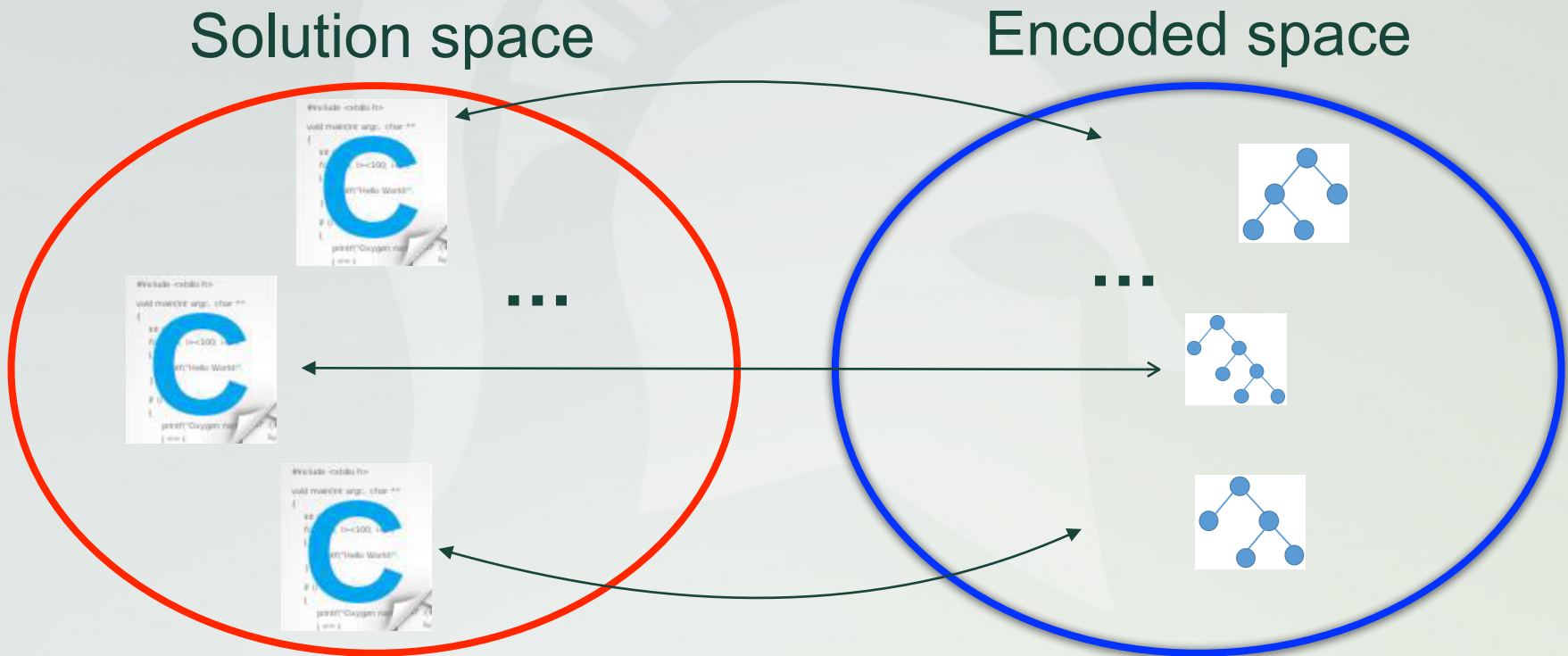
A computer program



AST Tree

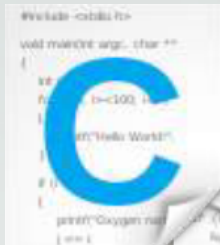
# Genotype and Phenotype

- Each solution in the solution space is a **computer program**



# Linear Genetic Programming

- LGP
  - Genetic representation: a sequence of instructions from imperative programs



...

Solution space

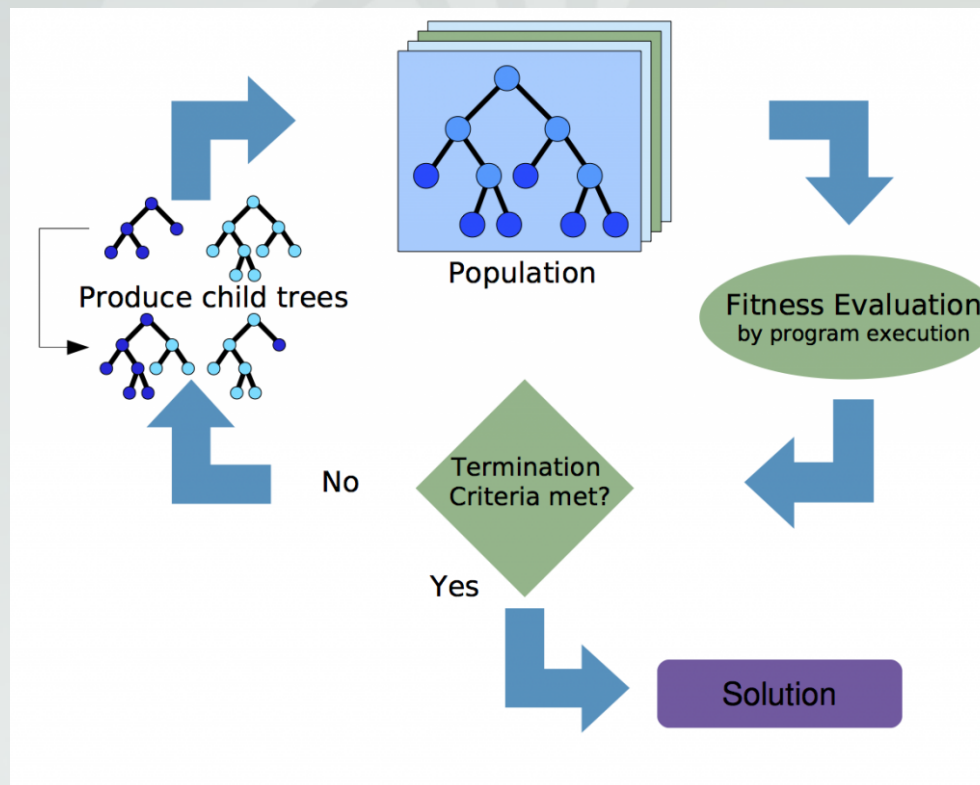
```
subl    $32, %esp
movl    $2, 20(%esp)
movl    $3, 24(%esp)
movl    24(%esp), %eax
movl    %eax, 4(%esp)
movl    20(%esp), %eax
movl    %eax, (%esp)
call    add
movl    %eax, 28(%esp)
movl    28(%esp), %eax
```

...

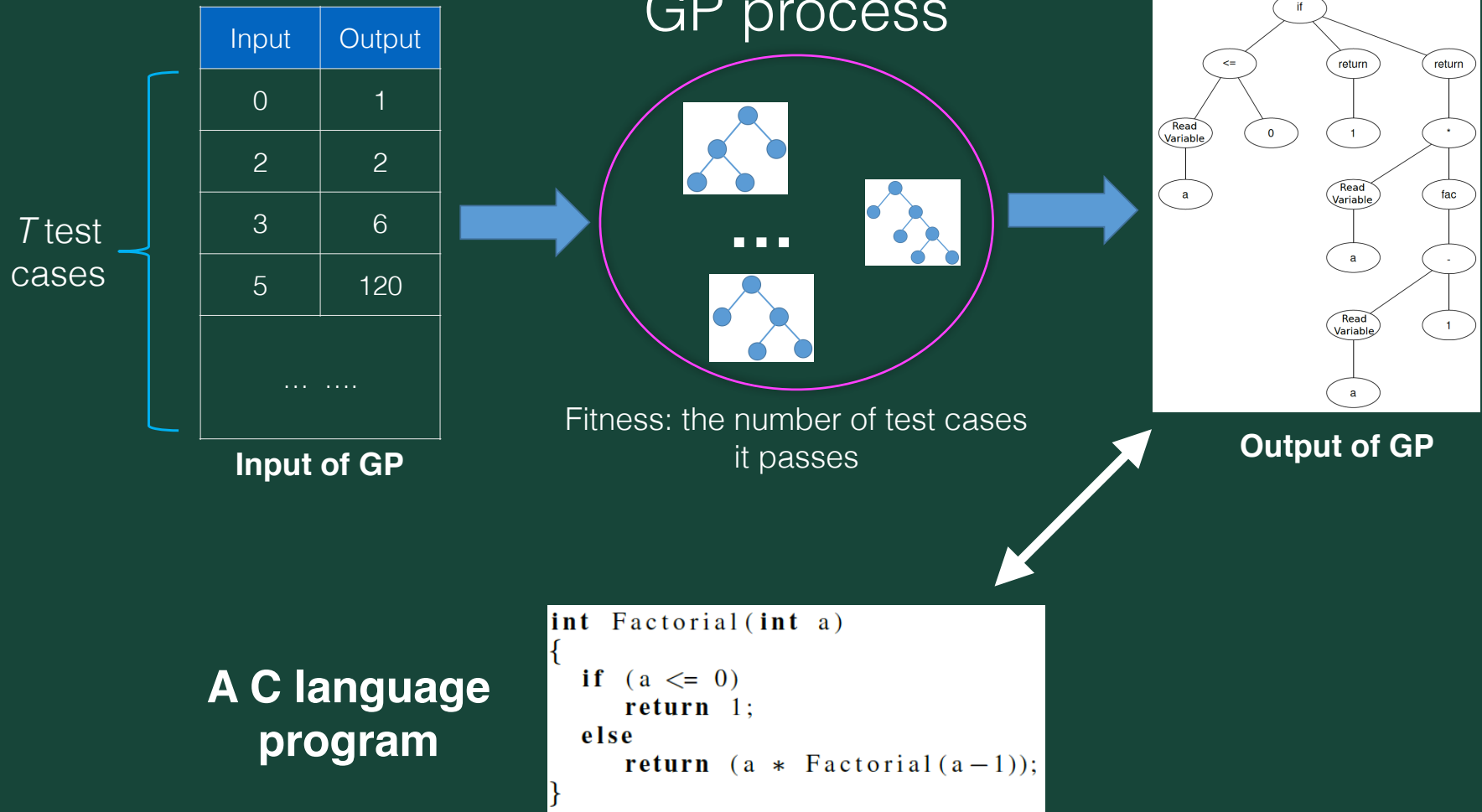
Encoded space



# Genetic Programming - Overall Process



# Genetic Programming



# Genetic Programming - The Abstract Syntax Tree

Here is a AST tree for a program containing about 10 lines of codes



If there are hundreds/thousands lines of code, can you imagine  
how complex the tree is?

Scalability is an issue!

# Genetic Programming for Software

- The ambition of writing larger pieces of software with GP has not been realized
- GP was **rarely** used to evolve **real-world** programs (Java/C/C++ ... programs) in the past several decades.
- The space of programs to be searched is **too large**
- Near-optimal solutions are usually not acceptable
- The distribution of the desired programs in the program space is **too sparse**
- Any search strategy would **fail**

# Genetic Programming for Machine Learning

- Even small/special programs can perform many difficult tasks
- GP can still do many exciting things
- GP can be applied to any kind of machine learning (i.e., regression and classification) tasks
- John R. Koza mentioned 76 instances where GP has been able to produce results that are competitive with human-produced results

# Genetic Improvement - The Idea

- Genetic Improvement (GI) is related to the original goal of GP (i.e., automatic programming), rather than symbolic regression.
- GP generally cannot start **from scratch** to create a real-world program
- GI offers a different slant: why not take human-written program as a starting point and **improve** it