UCD CASL

Complex & Adaptive Systems Laboratory

Dr. Michael O'Neill
Dr. Miguel Nicolau

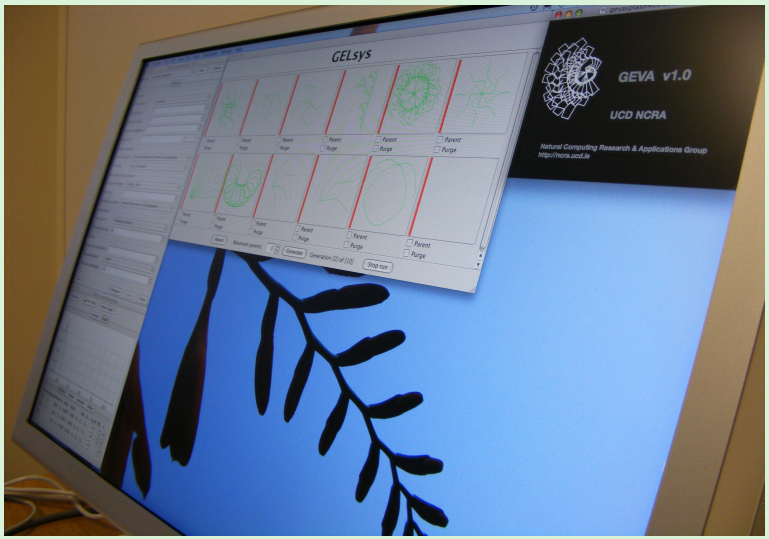
Grammatical Evolution

COMP30290 Natural Computing
COMP41190 Natural Computing and Applications

Grammatical Evolution

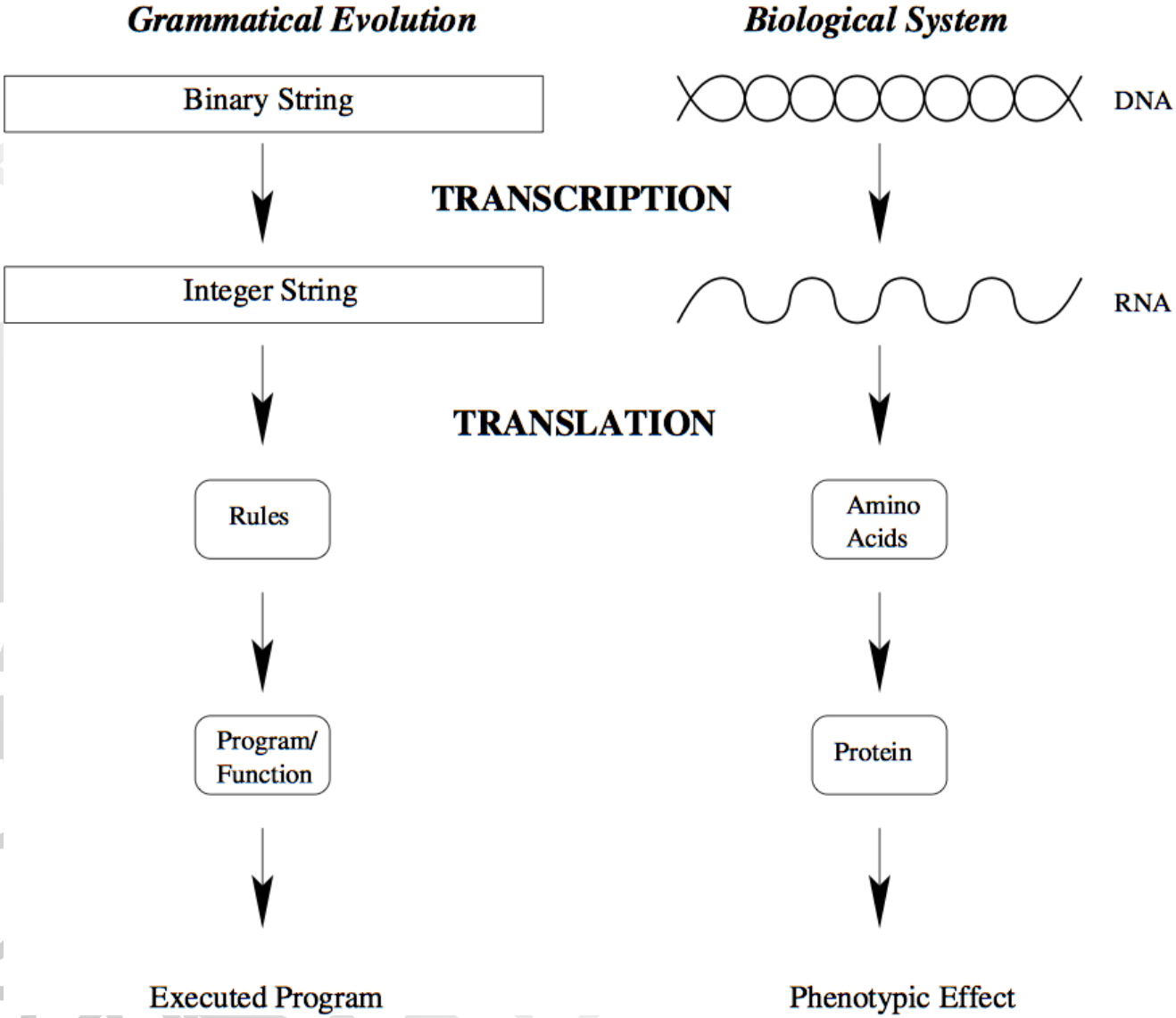
Grammatical Genetic Programming

- Chromosomes:
  - Linear;
  - Binary/Integer;
  - Variable-length.
- Genotype-Phenotype Map;
- Bio-inspired.



ONeill, Ryan. (2003). Grammatical Evolution. Kluwer Academic Press.
Brabazon, ONeill. (2006). Biologically Inspired Algorithms for Financial Modelling. Springer
Dempsey, ONeill, Brabazon. (2009). Foundations of GE in Dynamic Environments. Springer.

Genotype-Phenotype Map

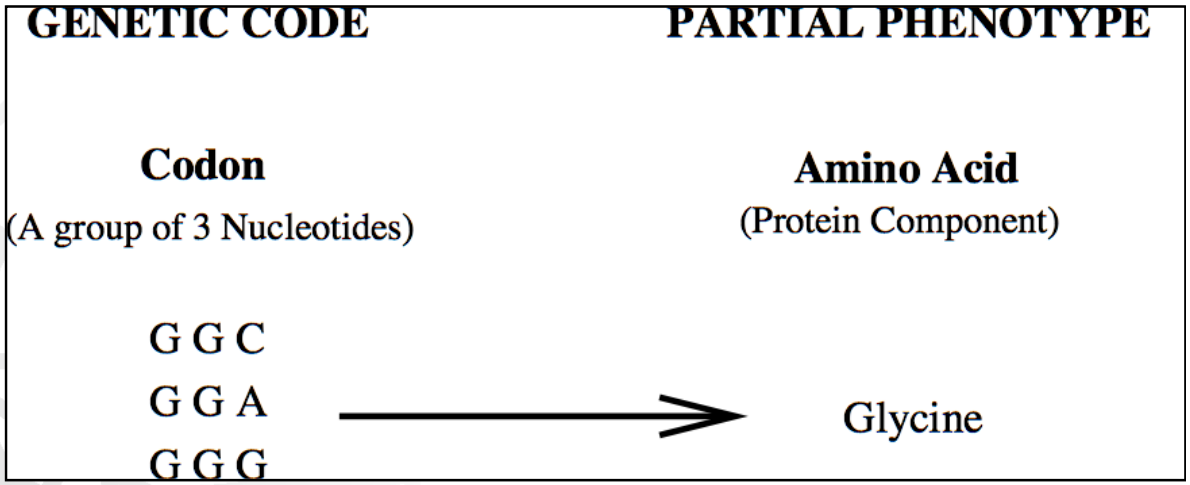


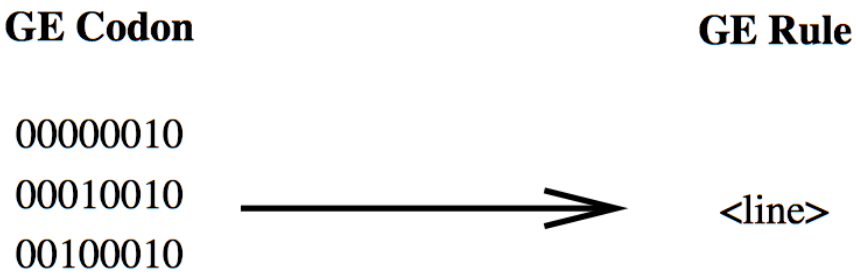
Genetic Code

	U	C	A	G	
U	UUU - Phe	UCU - Ser	UAU - Tyr	UGU - Cys	**U**
	UUC - Phe	UCC - Ser	UAC - Tyr	UGC - Cys	**C**
	UUA - Leu	UCA - Ser	UAA - Stop	UGA - Stop	**A**
	UUG - Leu	UCG - Ser	UAG - Stop	UGG - Trp	**G**
C	CUU - Leu	CCU - Pro	CAU - His	CGU - Arg	**U**
	CUC - Leu	CCC - Pro	CAC - His	CGC - Arg	**C**
	CUA - Leu	CCA - Pro	CAA - Gln	CGA - Arg	**A**
	CUG - Leu	CCG - Pro	CAG - Gln	CGG - Arg	**G**
A	AUU - Ile	ACU - Thr	AAU - Asn	AGU - Ser	**U**
	AUC - Ile	ACC - Thr	AAC - Asn	AGC - Ser	**C**
	AUA - Ile	ACA - Thr	AAA - Lys	AGA - Arg	**A**
	AUG - Met	ACG - Thr	AAG - Lys	AGG - Arg	**G**
G	GUU - Val	GCU - Ala	GAU - Asp	GGU - Gly	**U**
	GUC - Val	GCC - Ala	GAC - Asp	GGC - Gly	**C**
	GUA - Val	GCA - Ala	GAA - Glu	GGA - Gly	**A**
	GUG - Val	GCG - Ala	GAG - Glu	GGG - Gly	**G**

Code	Name	Code	Name
Phe	Phenylalanine	**Leu**	Leucine
Tyr	Tyrosine	**Cys**	Cysteine
Trp	Tryptophan	**Pro**	Proline
His	Histidine	**Gln**	Glutamine
Arg	Arginine	**Ile**	Isoleucine
Met	Methionine	**Thr**	Threonine
Asn	Asparagine	**Lys**	Lysine
Ser	Serine	**Val**	Valine
Ala	Alanine	**Asp**	Aspartic Acid
Glu	Glutamic Acid	**Gly**	Glycine

Genetic Code





For a rule with 2 choices, e.g.,

<code> :: = <line> (0)
| <code><line> (1)

i.e. (GE Codon Integer Value) MOD 2 = Rule Number

Example Mapping

Symbolic Regression Grammar

S = <expr>

P = <expr> ::= <expr> <op> <expr>
| (<expr> <op> <expr>)
| <pre-op> (<expr>)
| <var>

<op> ::= + | - | / | *

<pre-op> ::= sin | cos | tan | log

<var> ::= x

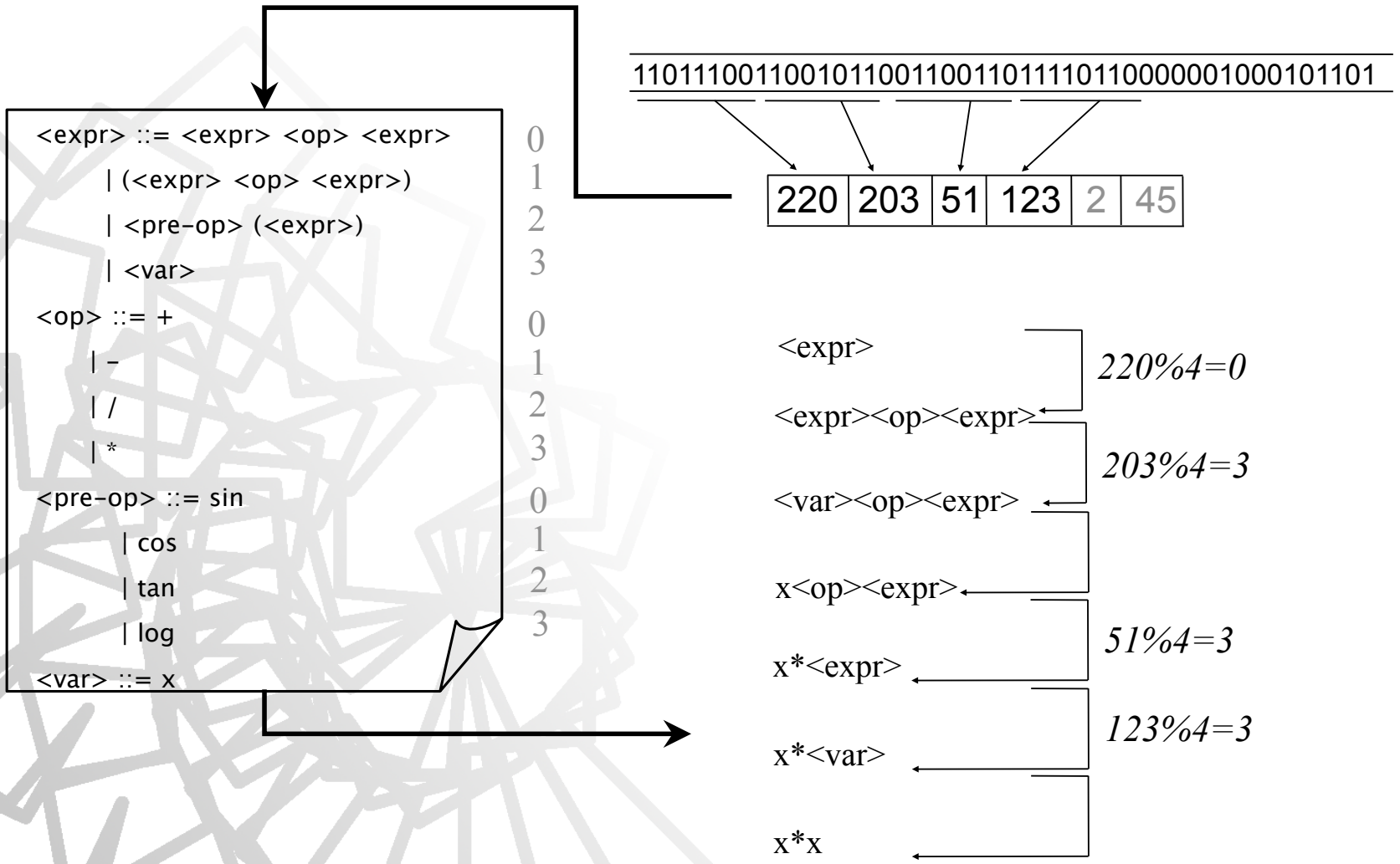
Wrapper

```
<func> ::= <header>
<header> ::= float symbreg(float x) { <body> }
<body> ::= <declarations><code><return>
<declarations> ::= float a;
<code> ::= a = <expr>;
<return> ::= return (a);
```

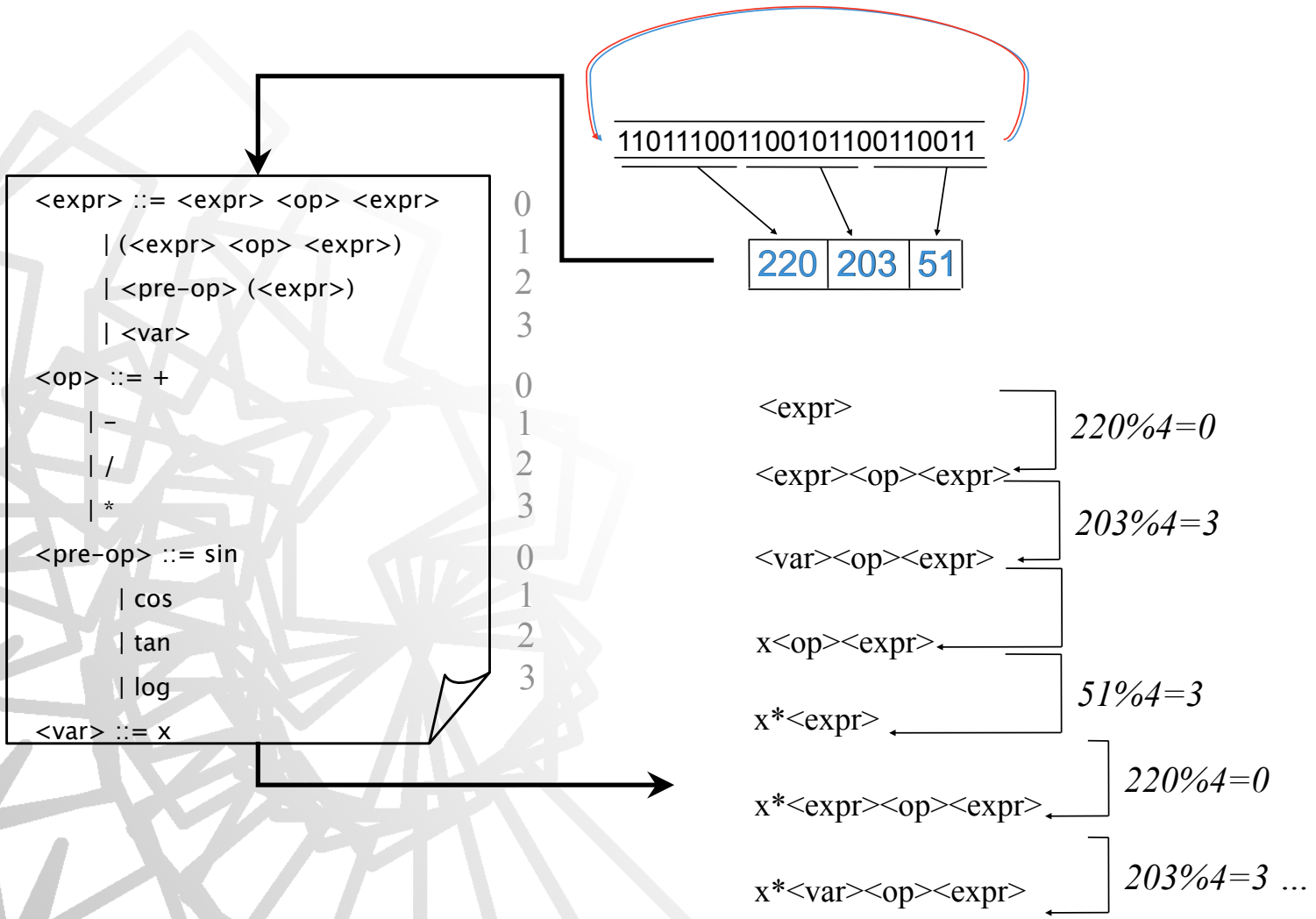
```
float symbreg(float x){
        float a;
        a= <expr>;
        return(a);
}
```

```
<code> ::= <line>;
         | <line>; <code>
<line> ::= ...
```

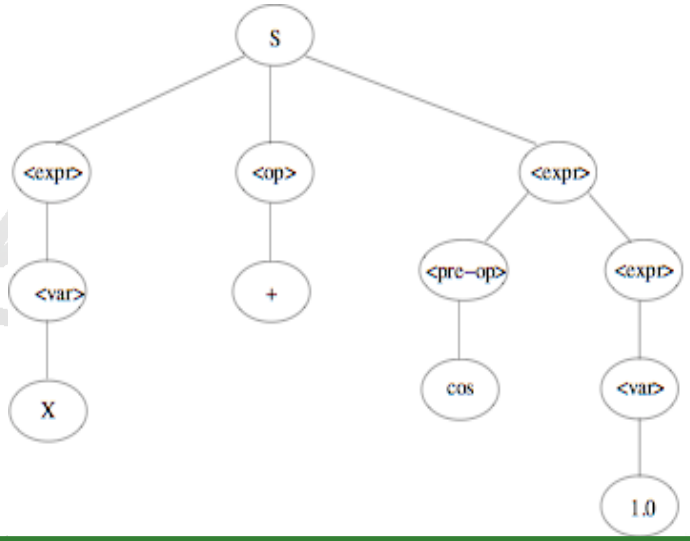
Example Mapping

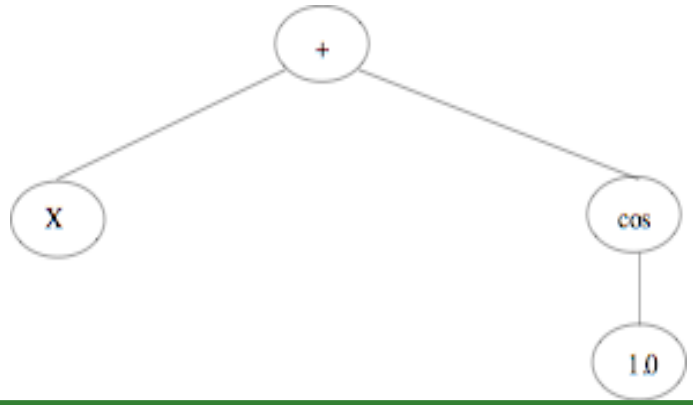


Example Mapping



Grammatical Evolution





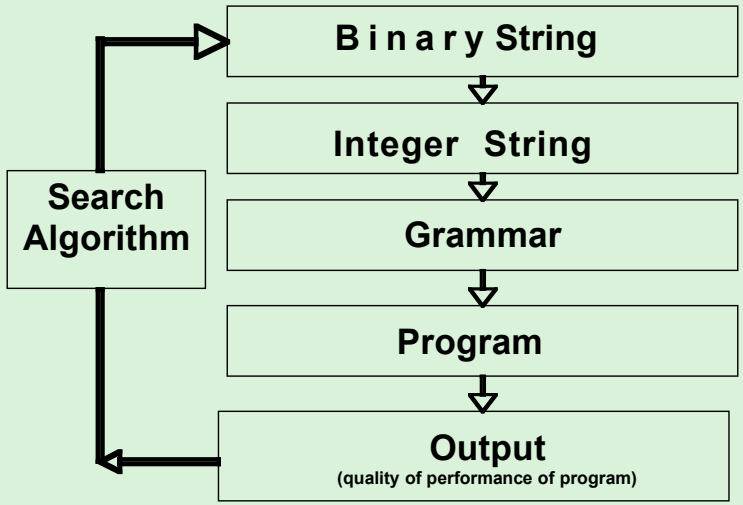
Genetic Operators

- Binary/Integer String (variable-length);
- Bit/Codon Mutation;
- 1pt Crossover;
- Duplication;
- Tree-based Operators.

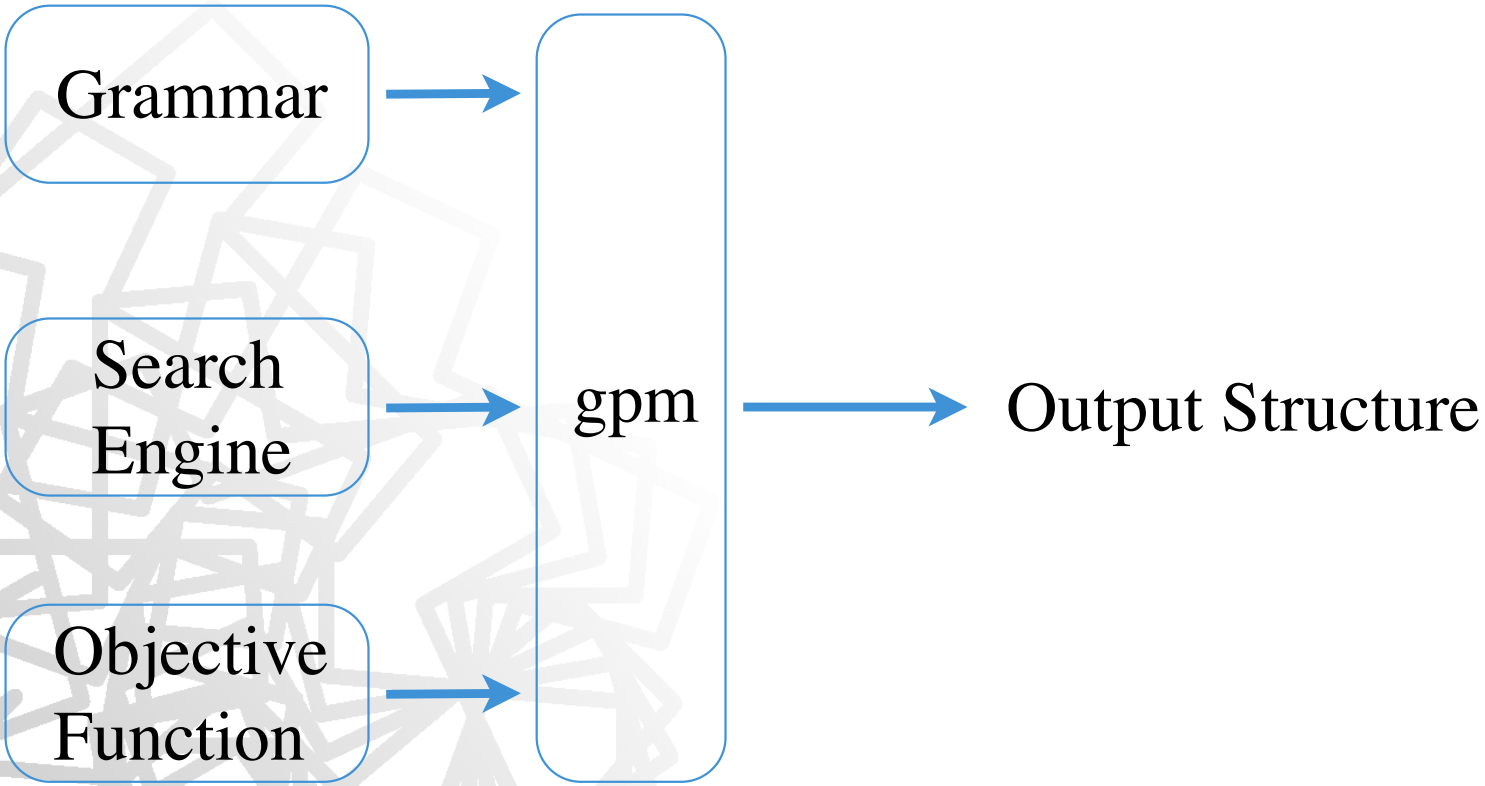
Grammatical Evolution

Mapping Analysis

- Genotype-Phenotype Map;
- Many-to-one;
- Diversity;
- Validity;
- Language Independence;
- Search & Solution Space.



GE Components



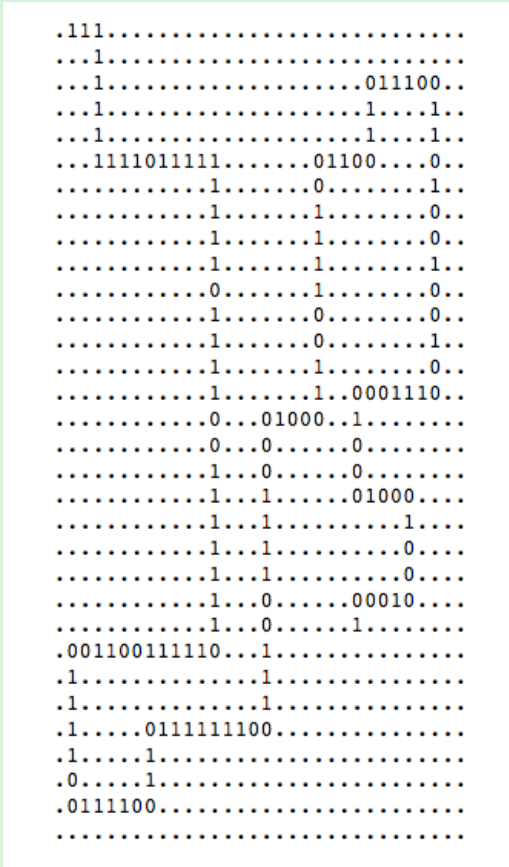
- Many important questions to address. . .

ONeill M., Vanneschi L., Gustafson S., Banzhaf W. (2010). Open Issues in Genetic Programming. Genetic Programming & Evolvable Machines. 11(3-4):339-363.

Santa Fe Ant Trail

Definition

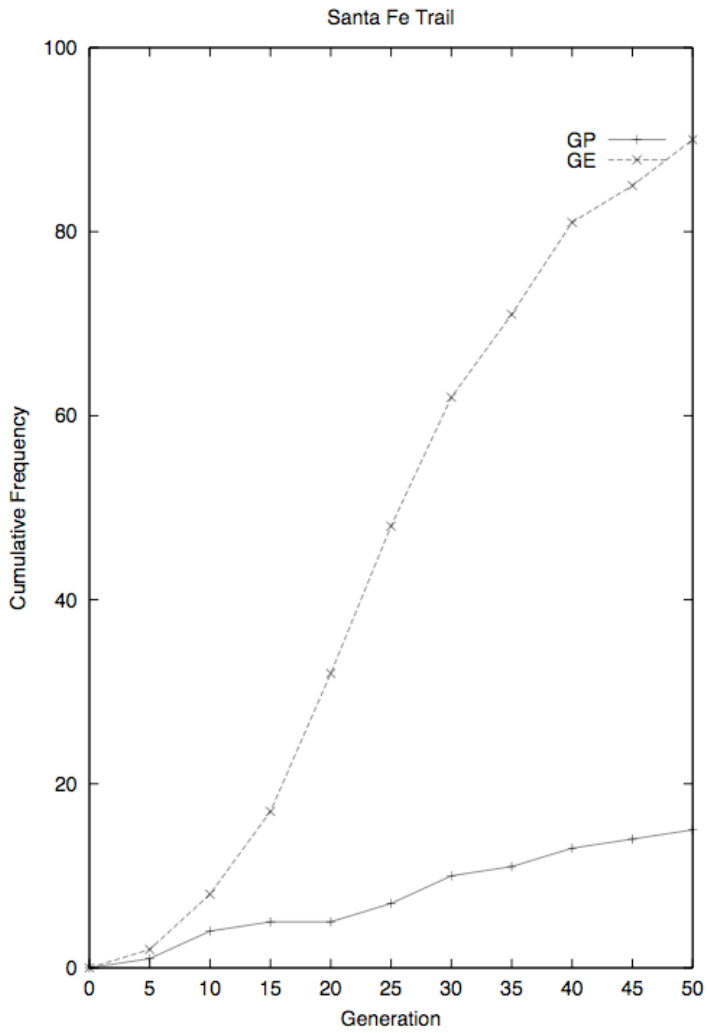
- Instructions:
  - move();
  - left();
  - right();
  - if_food_ahead();
- Fitness: pieces of food eaten within 600 time steps.



Santa Fe Ant Trail Grammar

```
<code> ::= <line>
        | <code><line>
<line> ::= <if-statement>
          |<op>
<if-statement> ::= if(food_ahead()){
                                  <line>
                          }
                          else{
                                  <line>
                          }
<op> ::= left();
       | right();
       | move();
```

Santa Fe Ant Trail Results



Symbolic Regression Grammar

<expr> ::= <expr> <op> <expr>
| (<expr> <op> <expr>)
| <pre-op> (<expr>)
| <var>

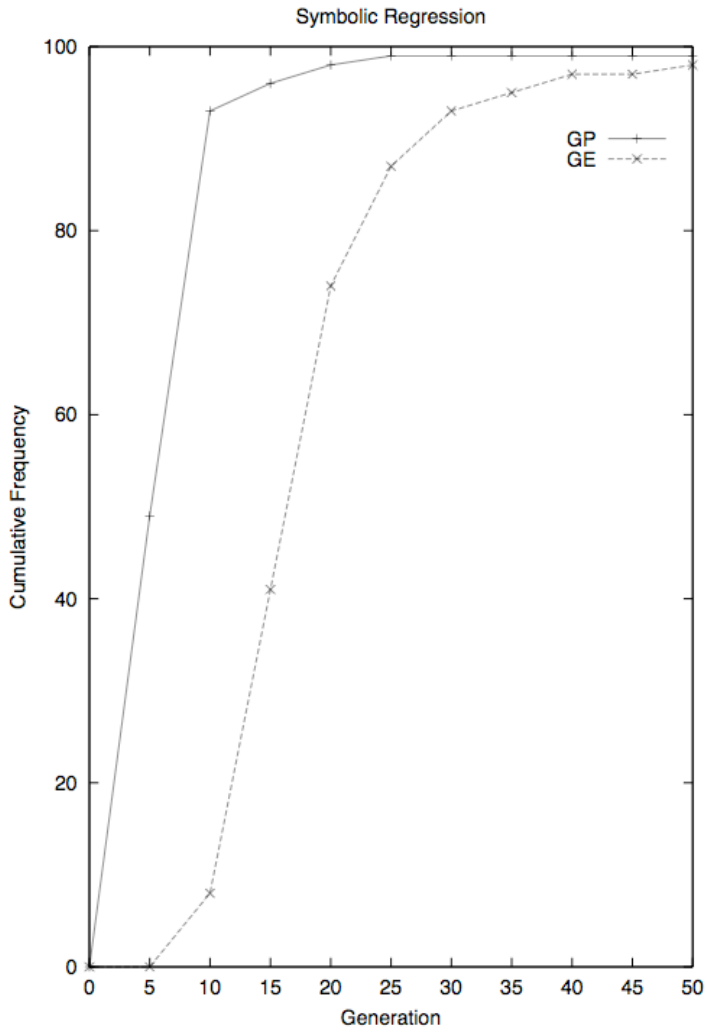
<op> ::= + | - | / | *

<pre-op> ::= sin | cos | exp | log

<var> ::= x | 1.0

$$F(x) = x + x^2 + x^3 + x^4$$

Symbolic Regression Results



Next Classes

- Tuesday 1st October:
  - Particle Swarm Optimisation (Mike).