

Programação Genética

Parte 1/2

Gisele L. Pappa

Programação Genética

- Criada com o objetivo de evoluir programas
- Lista de 36 (re-)invenções que são competitivas com as soluções propostas por humanos
 - 2 geraram patentes

<http://www.human-competitive.org/>



Programação Genética

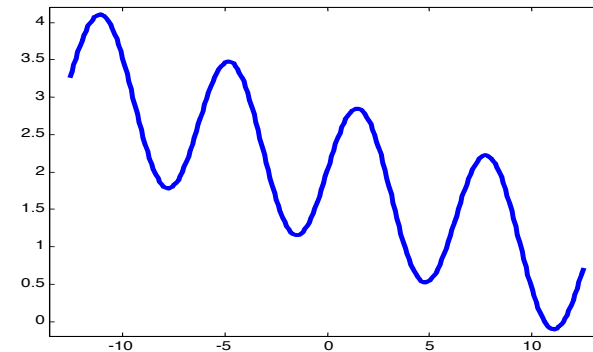
- Principais características
 - Um indivíduo é uma solução candidata contendo funções e terminais, e não apenas “dados” (variáveis /constantes)
 - Normalmente indivíduos tem tamanhos e formas variadas
 - Na teoria, um indivíduo é uma “receita” para resolver um dado problema, ao invés de uma solução para uma dada instância de um problema

GA vs. GP

- Ex: **otimização** de funções vs. **aproximação** de funções
 - Dada uma função complexa, por exemplo $\sin(x) - 0.1x + 2$, podemos usar um GA para encontrar o valor ótimo da função
 - Dado um conjunto de dados, contendo pares $\langle x, f(x) \rangle$, podemos utilizar GP para encontrar uma função $g(x)$ que se aproxime da função desconhecida $f(x)$ (regressão simbólica)

Otimização versus Aproximação de funções

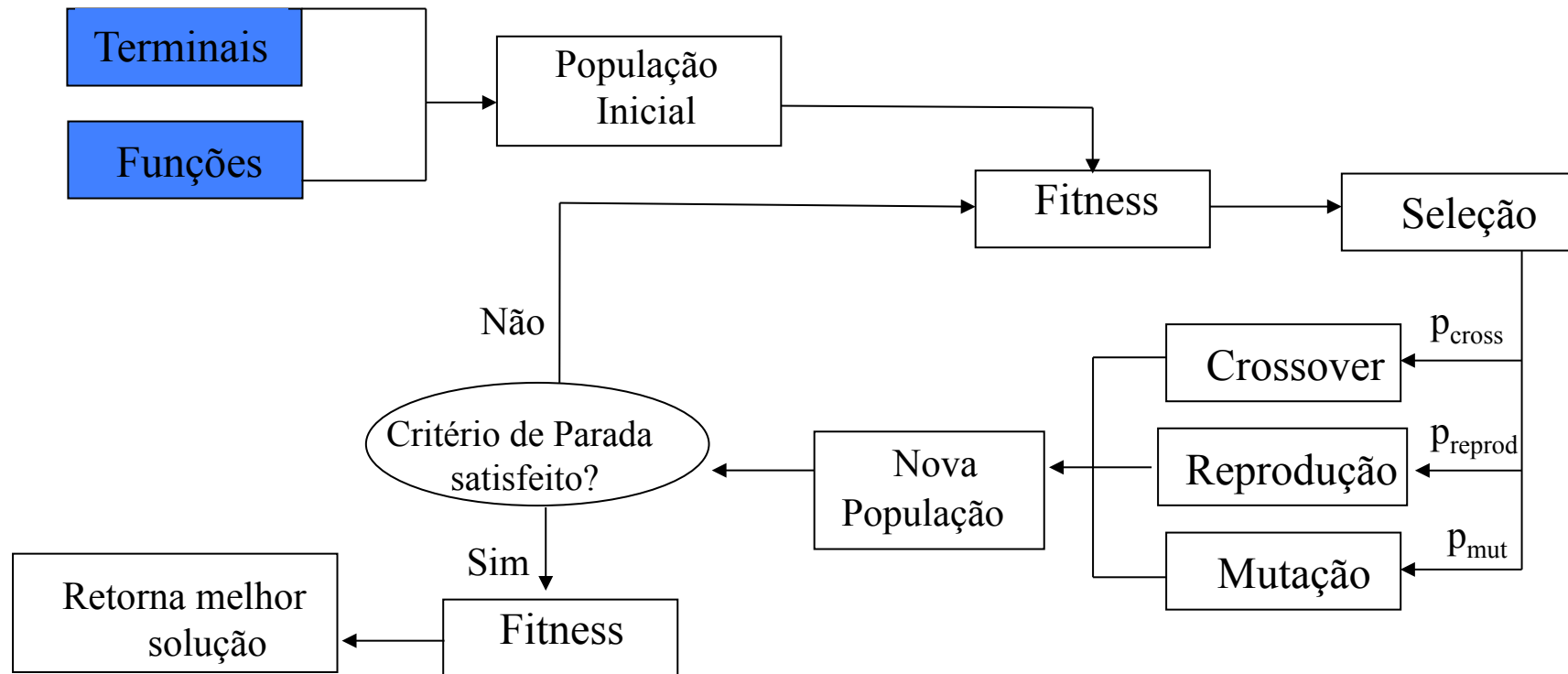
Exemplo de aplicação de um GA:
Encontrar os máximos ou mínimos
da função $f(x) = \sin(x) - 0.1x + 2$



Exemplo de aplicação de um GP:
Encontrar a função (programa)
que produz os pontos dados na
tabela ao lado

x	saída
-10	3.6
-8	2.1
-6	2.7
-4	2.6
.	.
.	.

Programação Genética

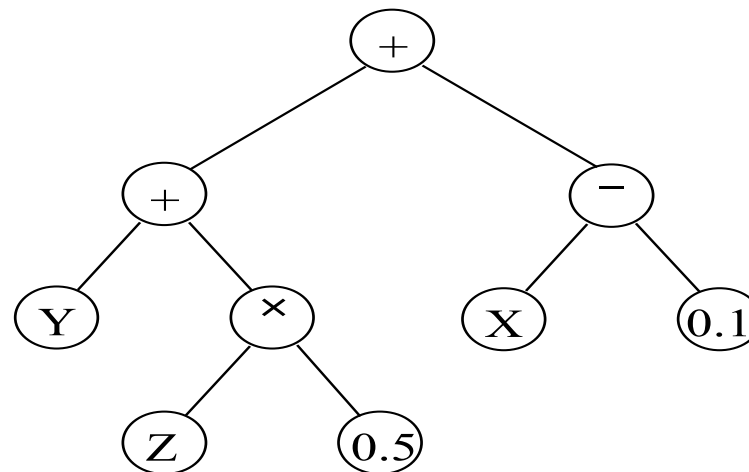


Programação Genética (GP)

- 2 componentes básicos:
 - Conjunto de terminais – variáveis e constantes
 - Conjunto de funções – funções apropriadas para resolver o problema em questão
- Tipos de representação:
 - Representação linear
 - Representação por árvores (mais comum)
 - Representação por grafos

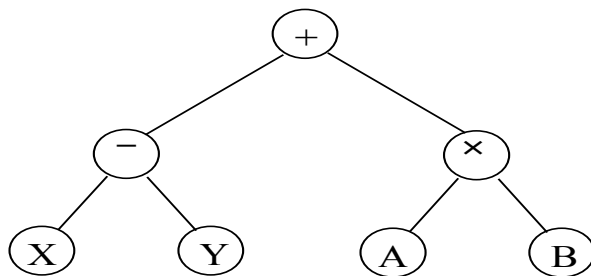
Componentes básicos de um GP

- Focaremos na representação por árvore
 - Nós internos: funções ou operadores
 - Nós folhas: variáveis ou constantes

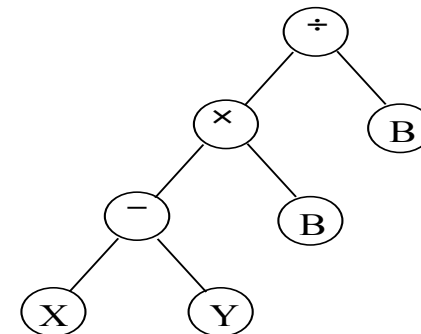


Exemplos de Indivíduos

- Conjunto de funções: $F = \{+, -, \times, \div\}$
- Conjunto de terminais: $T = \{A, B, X, Y\}$



Indivíduo 1: $(+ (- X Y) (\times A B))$



Indivíduo 2: $(\div (\times (- X Y) B) B)$

Inicialização da População

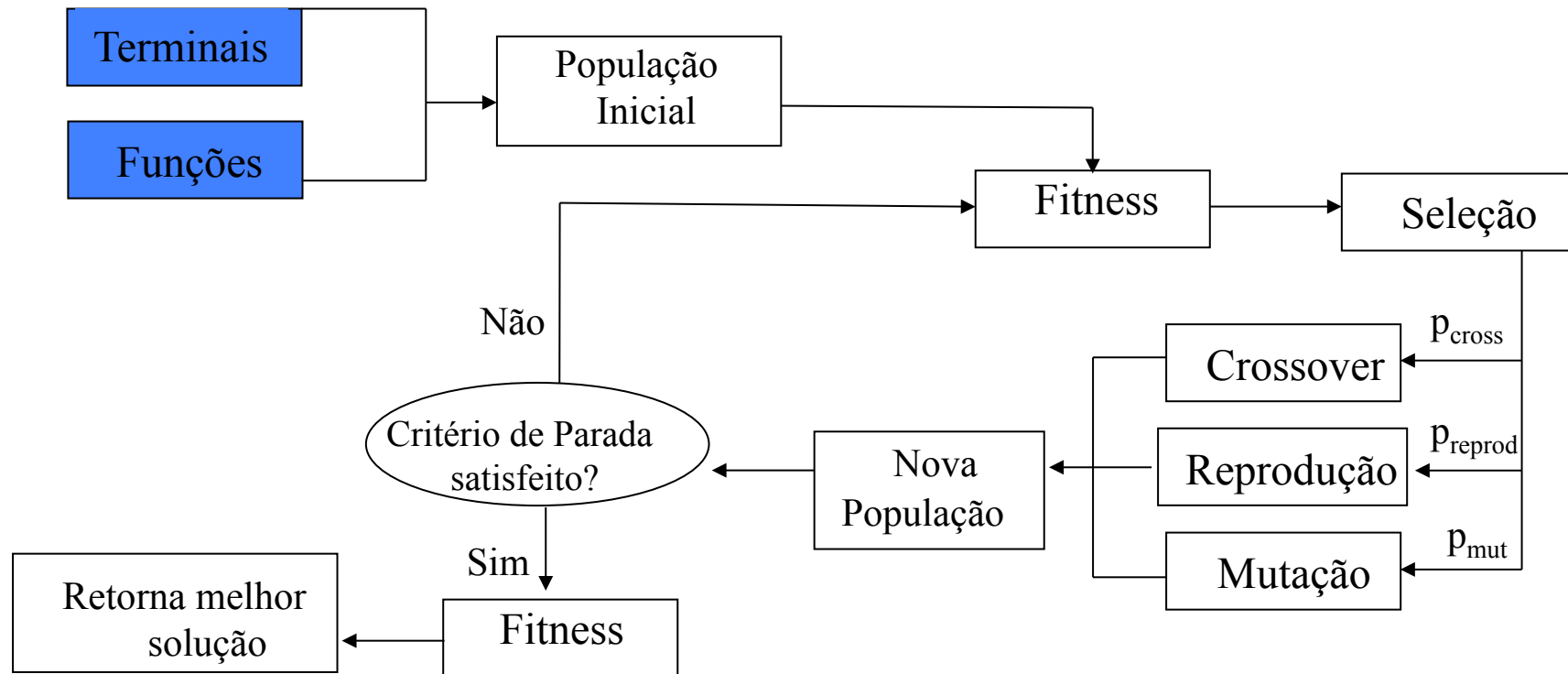
- 3 métodos principais
 - *Grow*: O nó de uma árvore é escolhido considerando elementos em **ambos** os conjuntos de terminais e funções, considerando uma altura máxima
 - Produz árvores com formas irregulares
 - *Full*: O nó de uma árvore é escolhido considerando elementos **apenas** do conjuntos de funções, até que a profundidade máxima seja alcançada. Nesse momento, nós passam a ser escolhidos do conjunto de terminais
 - Produz árvores balanceadas

Inicialização da População

– *Ramped half-and-half*

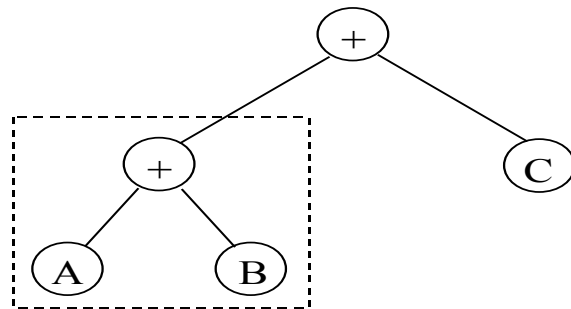
- Combina os métodos *full* e *grow* para aumentar diversidade
- Divide a população em subconjuntos com o mesmo número de indivíduos, e inicializa metade dos indivíduos de cada conjunto com o método *grow* e metade com o método *full*
- Se a profundidade máxima da árvore é 6, e o tamanho da população 50, serão criados um mesmo número de indivíduos com profundidades 2, 3, 4, 5 e 6 (nesse caso, 10 indivíduos). 5 deles serão inicializados utilizando *grow* e 5 *full*.

Programação Genética

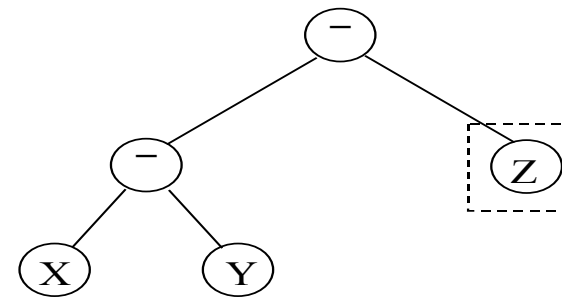


Operadores Genéticos

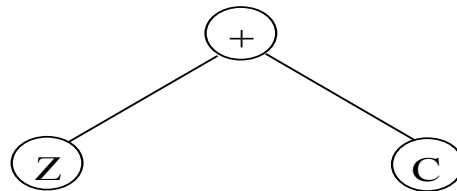
- Crossover: troca sub-árvores



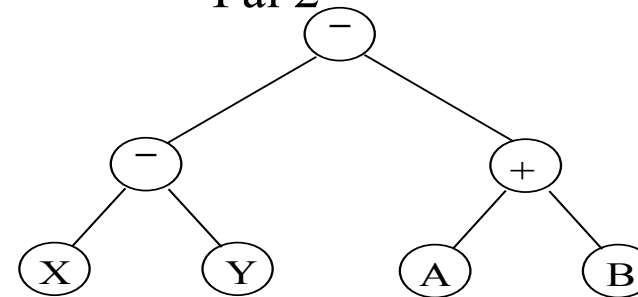
Pai 1



Pai 2



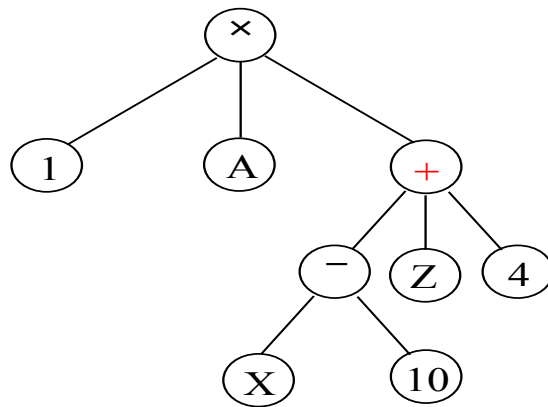
Filho 1



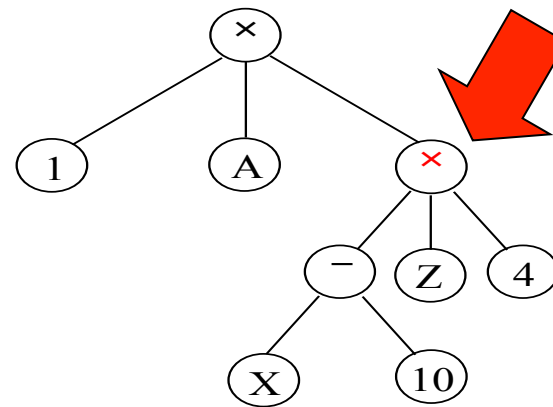
Filho 2

Operadores Genéticos

- Mutação de um ponto



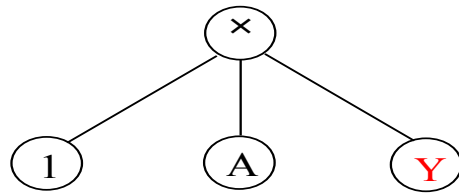
Pai



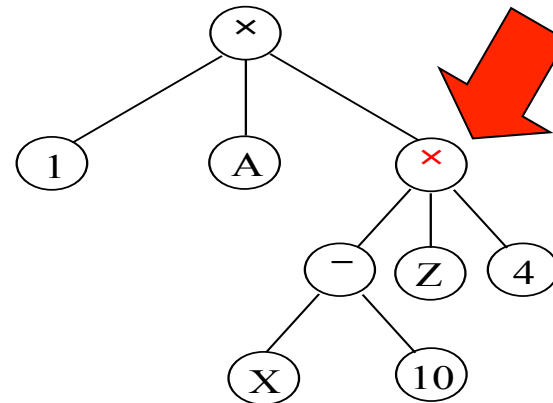
Filho

Operadores Genéticos

- Mutação de expansão



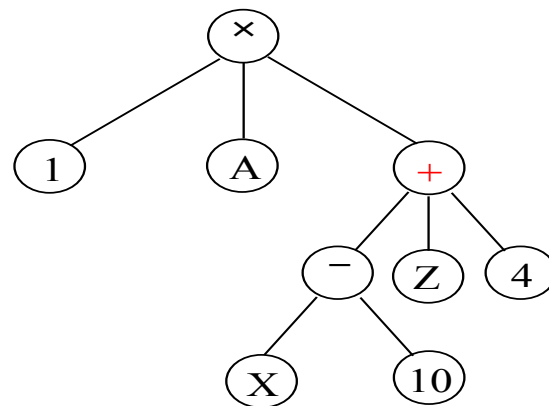
Pai



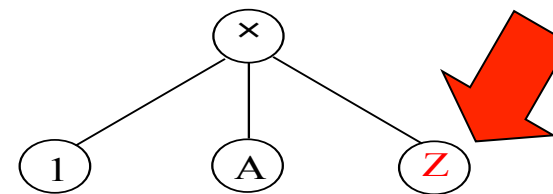
Filho

Operadores Genéticos

- Mutação de redução

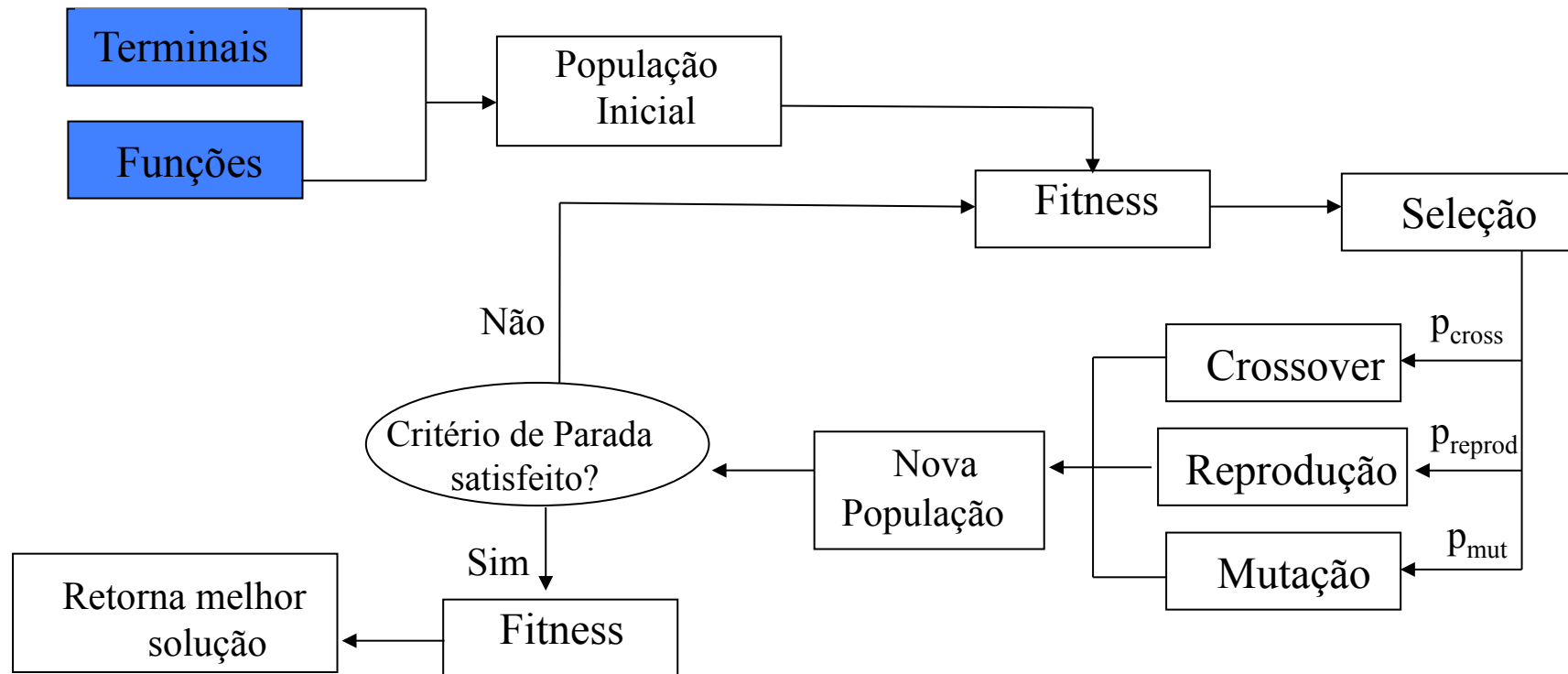


Pai



Filho

Programação Genética



Conjunto de Funções

- Propriedades desejadas
 - *Suficiência*: o poder de expressão é suficiente para representar uma solução candidata para o problema em questão
 - *Fechamento (Closure)*: uma função deve aceitar como entrada qualquer saída produzida por qualquer outro elemento do conjunto de funções ou do conjunto de terminais
 - *Parcimônia*: idealmente, conter apenas funções necessárias para resolver o problema em questão (propriedade não necessária, mas desejada)

Conjunto de Funções

- Como encontrar um equilíbrio entre poder de expressão e parcimônia?
 - Sugestão [Banzhaf et al. 1998, p. 111]:
 $FS = \{+, -, \times, /, OR, AND, XOR\}$

Conjunto de Funções

- Problemas com a propriedade de fechamento
 - Exige a modificação de certas funções/operadores
 - Ex. divisão por 0 é indefinida
 - Solução: divisão protegida – se o denominador for zero, retornar um valor padrão
 - Variáveis com diferentes tipos de dados
 - Ex., *Sexo* (binário: masculino, feminino) e *Idade* (inteiro)
 - Solução: GP restrito a sintaxe e/ou GP baseado em gramática

Leitura Recomendada

- A Field Guide to Genetic Programming, Livro online, <http://www.gp-field-guide.org.uk/>

Curiosidades

- <http://rogeralsing.com/2008/12/07/genetic-programming-evolution-of-mona-lisa/>

Agradecimentos

- A maioria dos slides dessa aula foram retiradas das aulas de Computação Natural de Alex A. Freitas

Programação Genética

Parte 1/2

Gisele L. Pappa