

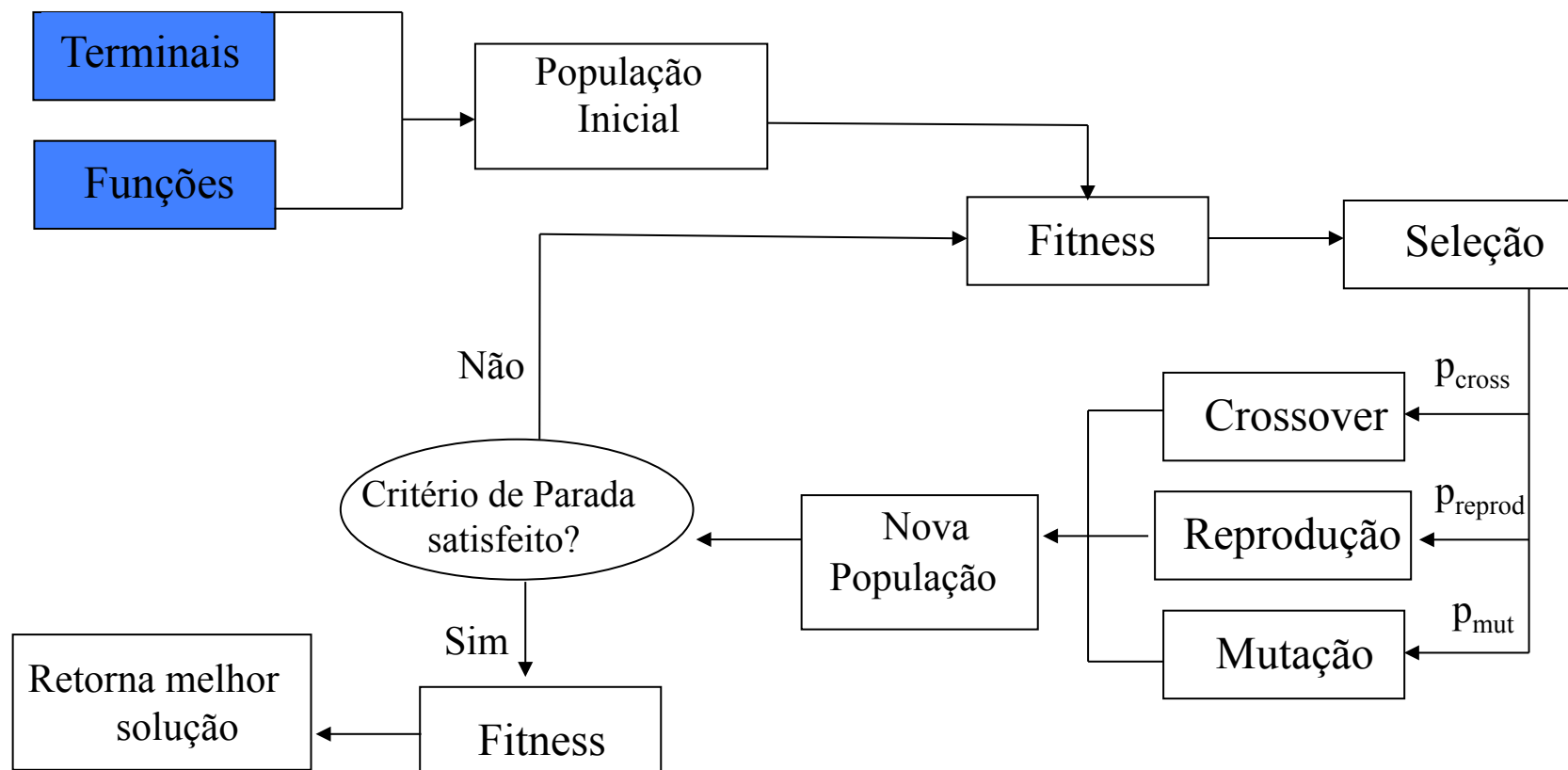
Programação Genética baseada em Gramáticas Parte 1/2

Gisele L. Pappa

Introdução

- Em GP, existem 3 propriedades que devem ser respeitadas ao criar um conjunto de funções de um PG, incluindo *fechamento*
- As dificuldades impostas pelo fechamento levaram a criação de novas vertentes dentro da GP:
 - GP restrito a sintaxe
 - GPs baseados em gramáticas

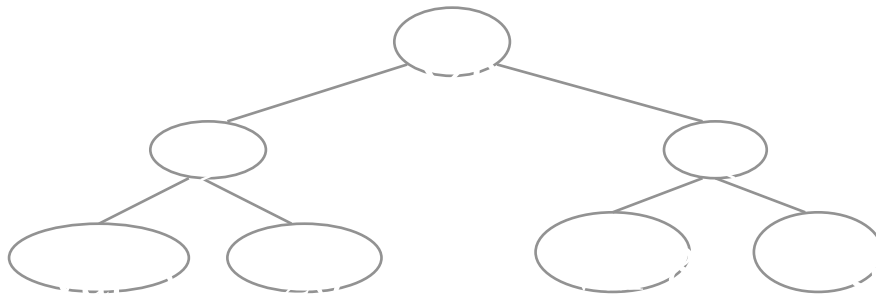
Programação Genética



GP Restrito a Sintaxe

- Para cada função do conjunto de funções, especificar o tipo de dados de seus argumentos e o tipo de dados retornado
- Cada terminal é também associado a um tipo de dados
- Cruzamento e mutação são modificados com respeito a restrições nos tipos de dados

GP Restrito a Sintaxe



Função	Tipo de dados dos argumentos	Tipo de dado retornado
+, -, *, /	(real, real)	real
>, <	(real, real)	boolean
AND, OR	(boolean, boolean)	boolean

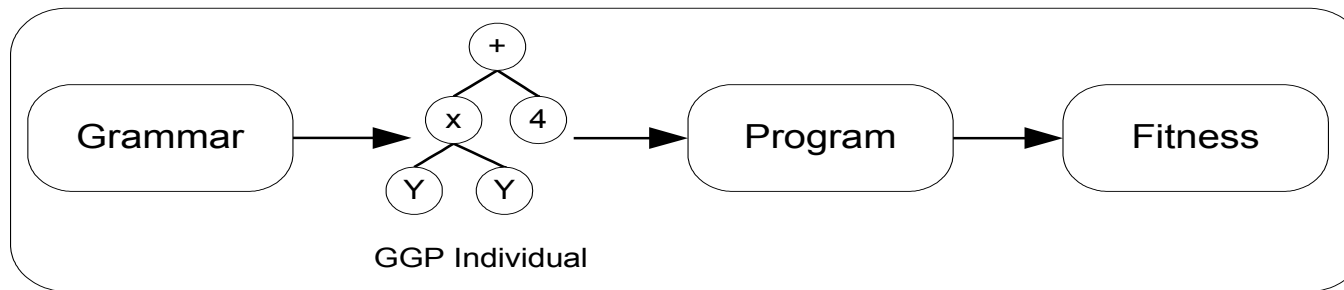
GP baseada em gramática

- Além de garantir a propriedade de fechamento, permite **incorporar ao espaço de busca domínio sobre o problema**
- GP baseada em gramática podem ser divididas em 2 grandes classes de acordo com:
 - Tipo de representação utilizado
 - Tipo da gramática utilizada
 - Livre de contexto, lógica, etc

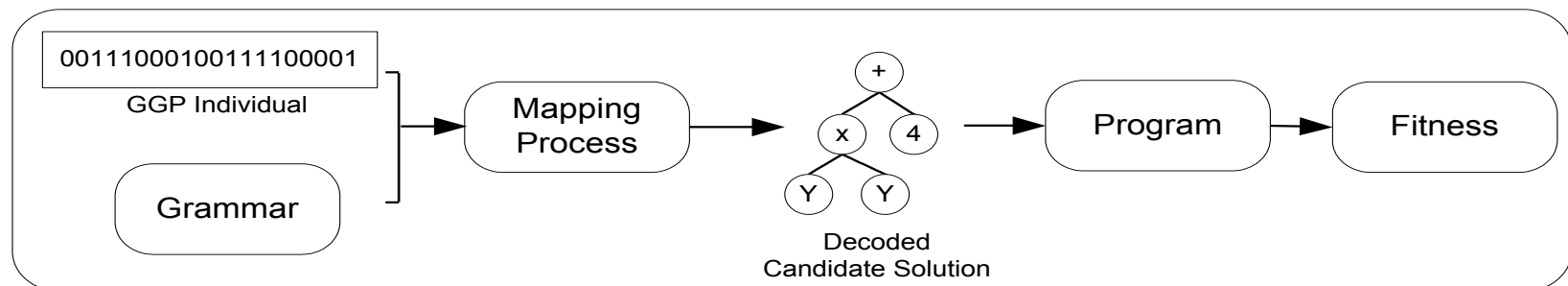
PG baseada em gramática

- Classificação de acordo com a representação

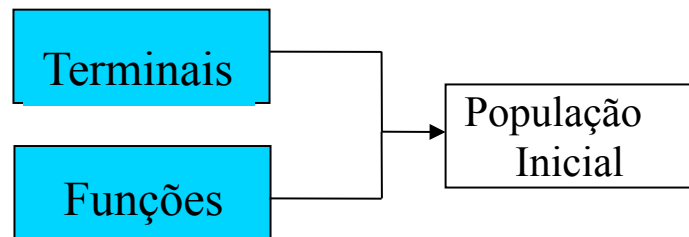
Tipo 1



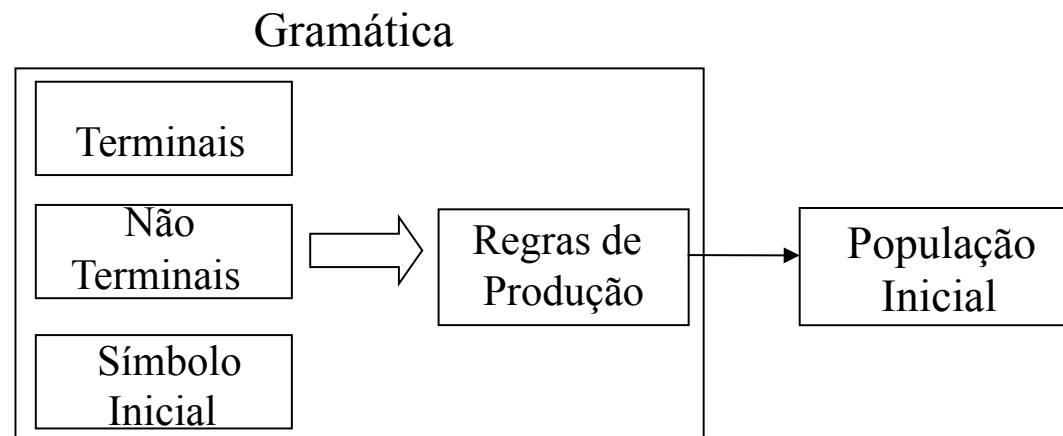
Tipo 2



PG baseada em Gramáticas (Tipo 1)



PG Tradicional



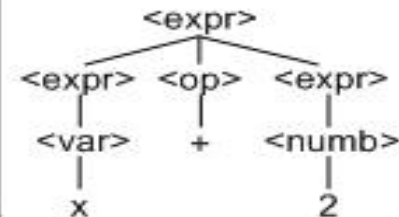
PG baseada em Gramáticas

Exemplo de Gramática

CFG Grammar

$\langle \text{expr} \rangle ::= \langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle \mid$ (1)
 $\langle \text{numb} \rangle \mid$ (2)
 $\langle \text{var} \rangle$ (3)
 $\langle \text{op} \rangle ::= + \mid$ (4)
 $-$ (5)
 $\langle \text{var} \rangle ::= x \mid$ (6)
 y (7)
 $\langle \text{numb} \rangle ::= 2 \mid$ (8)
 4 (9)

Derivation Tree for expression x+2



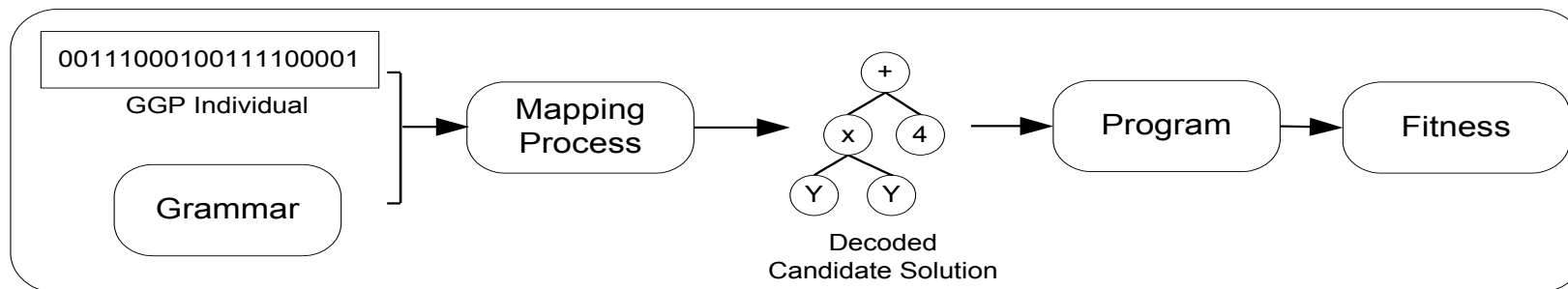
Derivation Steps followed to produce x+2

$\langle \text{expr} \rangle \xrightarrow{1} \langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle \xrightarrow{3} \langle \text{var} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle \xrightarrow{6} x \langle \text{op} \rangle \langle \text{expr} \rangle \xrightarrow{4}$
 $x + \langle \text{expr} \rangle \xrightarrow{2} x + \langle \text{numb} \rangle \xrightarrow{8} x + 2$

Diferenças em relação a PG convencional

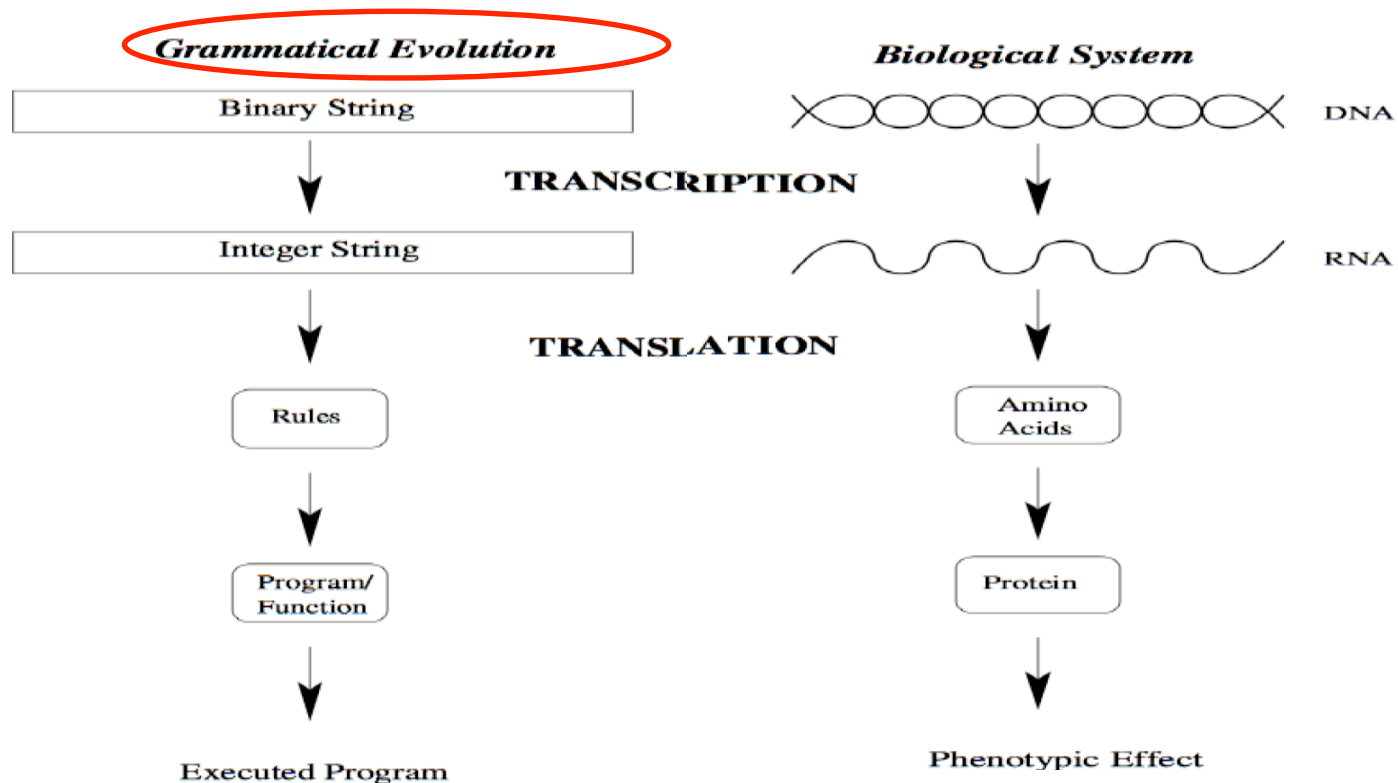
- Indivíduos criados através de mutação e crossover devem respeitar as regras de produção da gramática
- Crossover não tem um poder destrutivo

PG baseada em Gramáticas (Tipo 2)



- Indivíduos são normalmente representados por um vetor binário
- Existe um mapeamento do genótipo para o fenótipo baseado em processos biológicos

Mapeamento inspirado na biologia



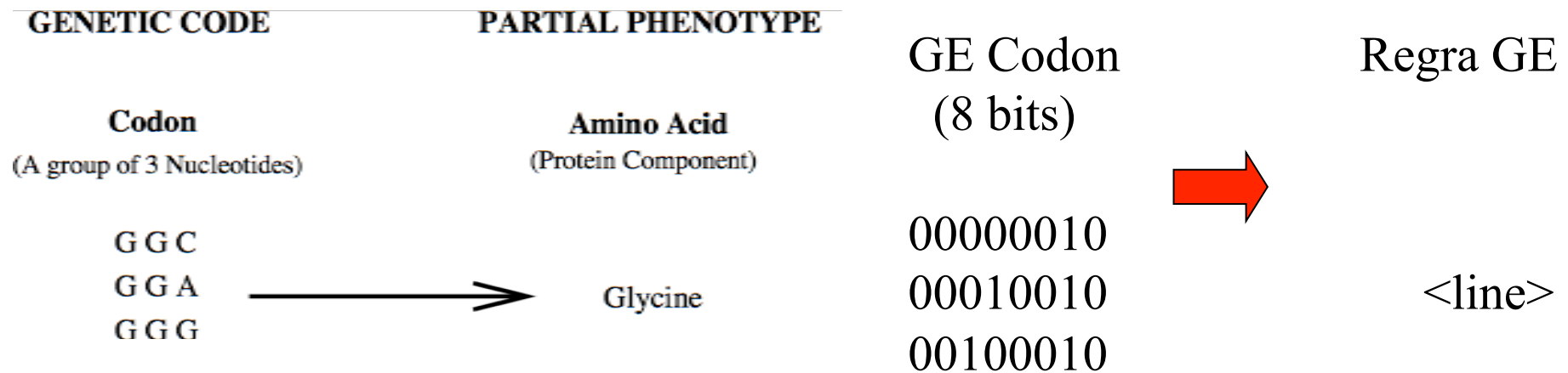
Código Genético

Códon	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

- Degeneração de código genético (diferentes códons mapeiam o mesmo aminoácido)

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

Código Genético



- Para uma regra com duas escolhas
 $\langle \text{code} \rangle ::= \langle \text{line} \rangle \mid (0)$
 $\langle \text{code} \rangle \langle \text{line} \rangle (1)$

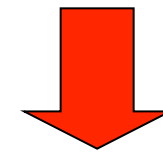
- Valor do códon de um GE mod número de regras determina o número da regra

Exemplo de Mapeamento

00101011011101101000101100010001

Indivíduo

43	118	144	17
----	-----	-----	----



Gramática

A $\langle \text{seq} \rangle ::= \langle \text{vowel} \rangle \quad (0)$
 $|\langle \text{seq} \rangle \langle \text{vowel} \rangle \quad (1)$

B $\langle \text{vowel} \rangle ::= a \quad (0)$
 $|\ e \quad (1)$
 $|\ i \quad (2)$
 $|\ o \quad (3)$
 $|\ u \quad (4)$

Processo de Decodificação

$\langle \text{seq} \rangle$	$43 \% 2 = 1$
$\langle \text{seq} \rangle \langle \text{vowel} \rangle$	$118 \% 2 = 0$
$\langle \text{vowel} \rangle \langle \text{vowel} \rangle$	$144 \% 5 = 4$
$u \langle \text{vowel} \rangle$	$17 \% 5 = 2$
ui	

Mapeamento

- O que acontece se eu termino de ler o genótipo e meu indivíduo ainda apresenta não-terminais?
 - Uso o conceito de “**wrapping**”
 - Reaproveitamento de material genético (inspirado na sobreposição de genes, comum em bactérias)

Implicações da Degeneração de Código Genético

- Aparecimento de mutações neutras
- Variações no genótipo não tem efeito no fenótipo

	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

Representação

- Trabalha com vetores de bits de tamanho variável
- Ao gerar a população inicial, determina um número máximo de códon que cada indivíduo pode ter

Operadores Genéticos

- Mutação de um ponto
- Crossover de um ponto
- Duplicação de códon
 - Selecionados aleatoriamente e inseridos antes do último códon

Principais características de Evolução Gramatical

- Separa o genótipo do fenótipo
- Degeneração de código genético
 - Ajuda a manter a diversidade da população
 - Ajuda a preservar a funcionalidade dos programas através de mutações neutras
- Operador *wrapping*
 - Reusar código genético

Críticas (Problemas)

- Como no GP, não existe semântica
- Cruzamento não faz muito sentido
 - Estaremos trocando bits que não fazem referência alguma a gramática
 - Operador *wrap* também faz com que o efeito do cruzamento seja amplificado
- Não existe localidade nos operadores, característica importante em EAs

Leitura Recomendada

- O'Neil M., Ryan C. *Automatic Generation of Programs with Grammatical Evolution*. In Proceedings of AICS 1999, pages 72-78.
- P. A. Whigham, Grammatically-based Genetic Programming, Proc. of the Workshop on Genetic Programming: From Theory to Real-World Applications, 1995, pages 33-41.

Mais informações

- <http://www.grammatical-evolution.org/>

Programação Genética baseada em Gramáticas Parte 1/2

Gisele L. Pappa