

Automatic Lexicon Expansion for Sentiment Analysis using Evolutionary Algorithms

Airton Bordin Junior

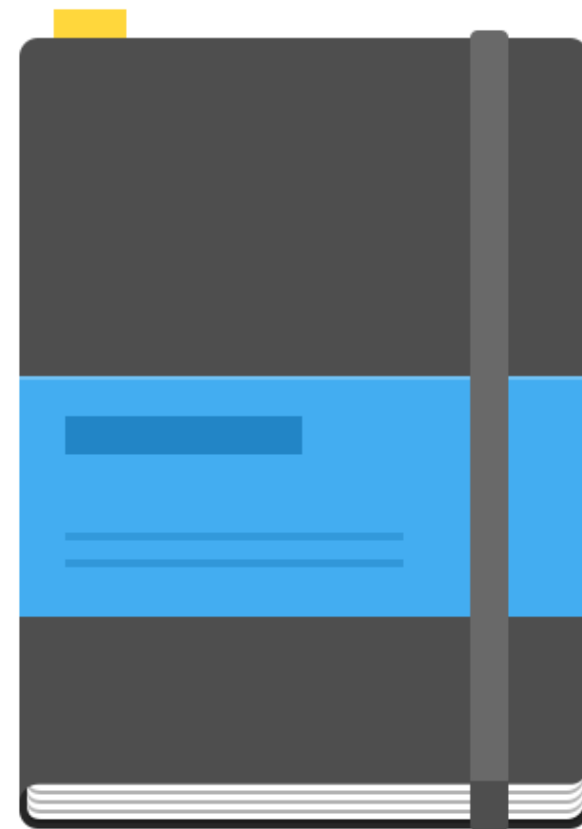
[airtonbjunior@gmail.com]

Mestrado em Ciência da Computação

Universidade Federal de Goiás (UFG) - Instituto de Informática – Junho/2017

Programação

- Introdução
- Heurísticas e Metaheurísticas
- Algoritmos evolucionários
- Análise de Sentimentos
- Programação Genética
- Referências





Introdução

- Problemas computacionais

Tratáveis

- Polinomiais
- Algoritmos determinísticos

Intratáveis

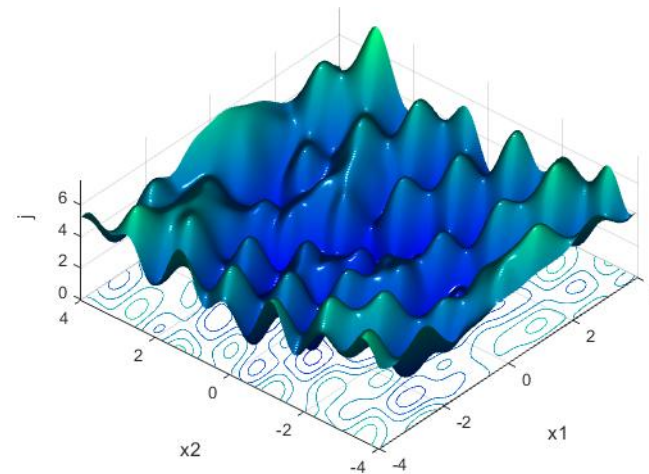
- Não polinomiais
- Algoritmos não determinísticos
- Solução determinística inviável
 - Sem solução em tempo hábil





Heurística

- Impraticabilidade de encontrar/calcular a melhor resposta para problemas não polinomiais;
- Desafio: produzir, em tempo reduzido, soluções tão próximas quanto possíveis da solução ótima.

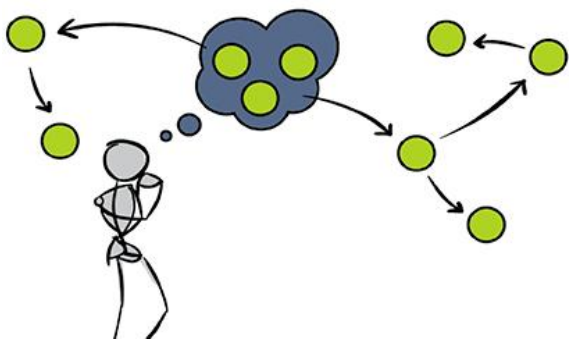




Metaheurística

Propriedades e características das metaheurísticas

[SALIBA, 2010]



Estratégias que guiam o processo de busca;

Exploração eficiente do espaço de busca - soluções ótimas ou quase ótimas;

De simples procedimentos de busca local a complexos processos de aprendizado;

Aproximados e usualmente não determinísticos;

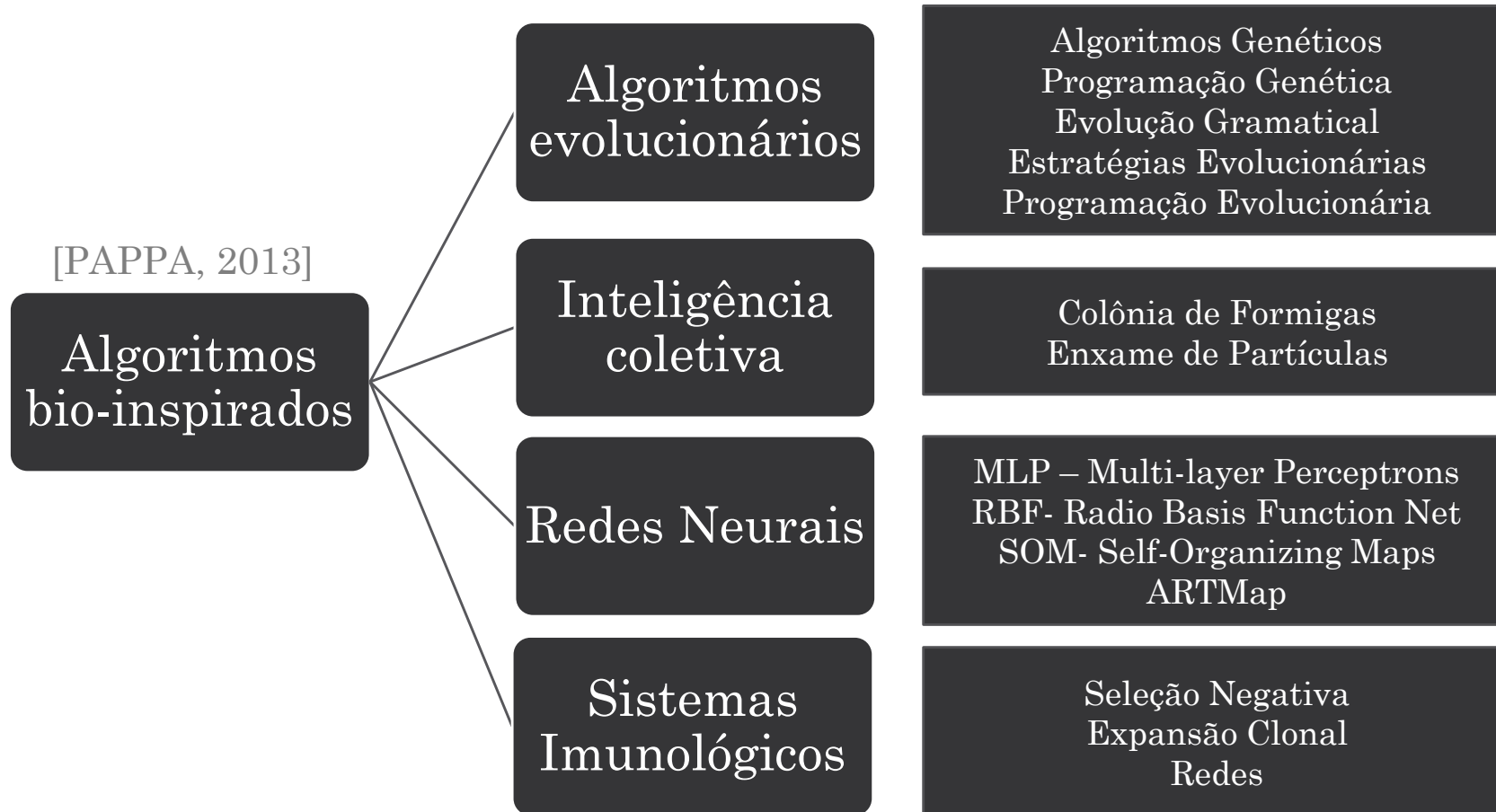
Podem incorporar mecanismos para evitar ficar presos em áreas confinadas do espaço de busca;

Não são específicas para um determinado problema;

Podem usar um conhecimento específico do problema na forma de heurísticas que são controladas por uma estratégia de nível superior.

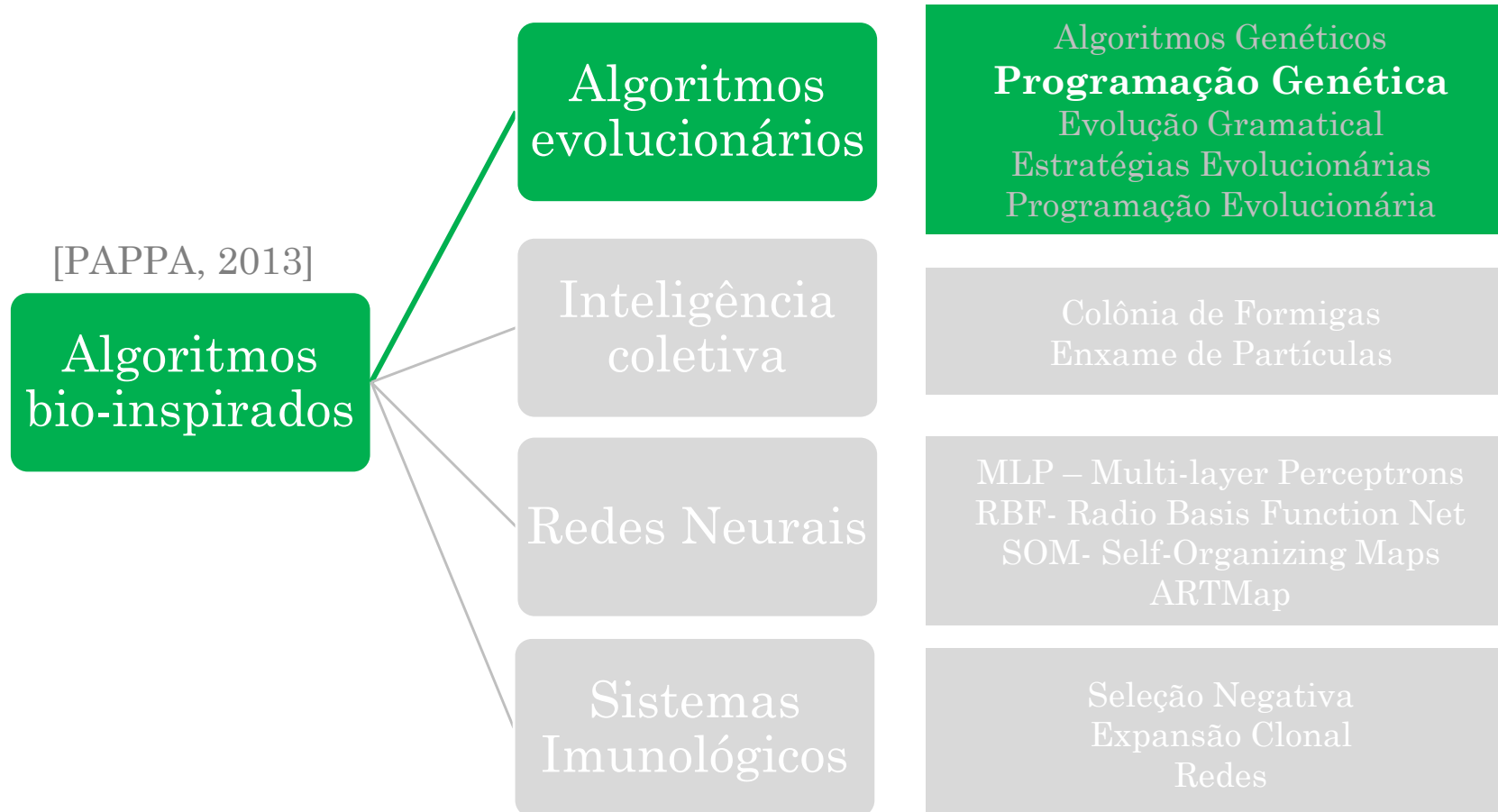


Algoritmos bio-inspirados





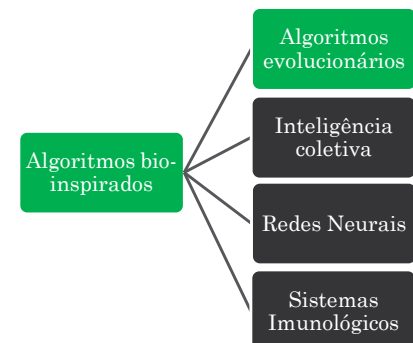
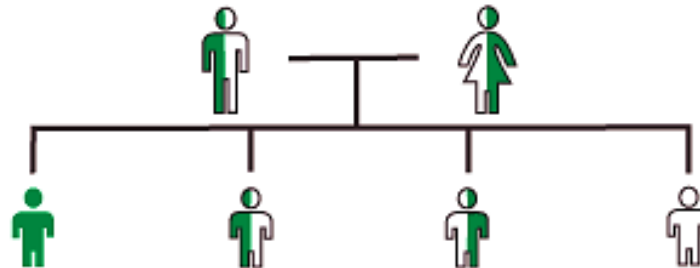
Algoritmos bio-inspirados





Algoritmos evolucionários

- Inspirados na teoria de evolução de Darwin;
- Evolução: mudança das características (genéticas) de uma população de uma geração para a próxima
 - Mutação dos genes;
 - Recombinação dos genes dos pais.

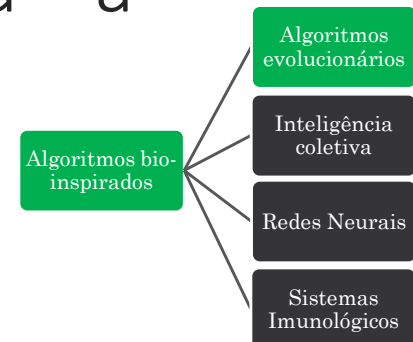




Algoritmos evolucionários

- Evolução é caracterizada basicamente por um processo constituído de 3 passos [VON ZUBEN, 2005]

1. Reprodução com herança genética;
2. Introdução de variação aleatória em uma população de indivíduos;
3. Aplicação da “seleção natural” para a produção da próxima geração.



Análise de Sentimentos



Análise de sentimentos

- Também chamado de Mineração de Opiniões;
- Estudo de opiniões que expressam/implicam um sentimento positivo/negativo;
- Opiniões e sentimentos subjetivos (não factuais)
- Nomenclaturas utilizadas
 - Orientação Semântica;
 - Polaridade.



Análise de sentimentos

- Motivação:
 - Aumento na quantidade de pessoas com acesso à Internet;
 - Consequente aumento de conteúdo gerado pelas pessoas;
 - Analisar/minerar os sentimentos/opiniões, identificar o sentimento das pessoas sobre determinado assunto/produto/contexto
 - Pode ser muito valioso para empresas, governos, etc.



Análise de sentimentos





Análise de sentimentos



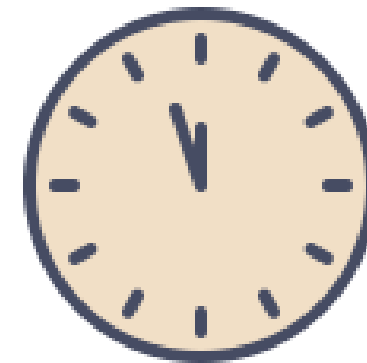
460 mil *tweets*



400 horas de vídeo



510 mil comentários



Por minuto



Análise de sentimentos

- Definição importante [LIU, 2012]

Opinião (quíntupla)

$(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$

(Entity, Aspect, Sentiment, Holder, Time)



Análise de sentimentos

- **Léxico de sentimentos**
 - Conjunto de palavras e frases com suas orientações semânticas.
- **Principais abordagens para criação/expansão do Léxico**
 - Manual;
 - Baseada em dicionário;
 - Baseada em Corpus.



Análise de sentimentos

- Abordagem manual
 - Por sua característica inerente, é limitada ao esforço de especialistas humanos;
 - Mais lenta que outras abordagens;
 - Raramente é utilizada como única forma de criação/expansão do Léxico.





Análise de sentimentos

- Baseada em dicionário
 - Usa um dicionário como base
 - *WordNet*, por exemplo.
 - Por meio de palavras-semente com polaridade conhecida, faz um processamento de forma a construir um dicionário léxico descobrindo a orientação semântica das palavras;
 - Diversas abordagens foram desenvolvidas
 - Sinônimos, antônimos, sufixos, prefixos;
 - Pointwise Mutual Information (PMI);
 - Distância em grafos;
 - *Label Propagation*, etc.



Análise de sentimentos

- Baseada em *Corpus*
 - Descobrir a orientação semântica das palavras no domínio do *Corpus*;
 - Adaptação de um Léxico de propósito geral para um domínio específico;
 - Palavras podem ter polaridade diferente em contextos distintos
 - Exemplo: câncer
 - Em um domínio técnico, a palavra pode não ter uma orientação semântica negativa e, sim, neutra.



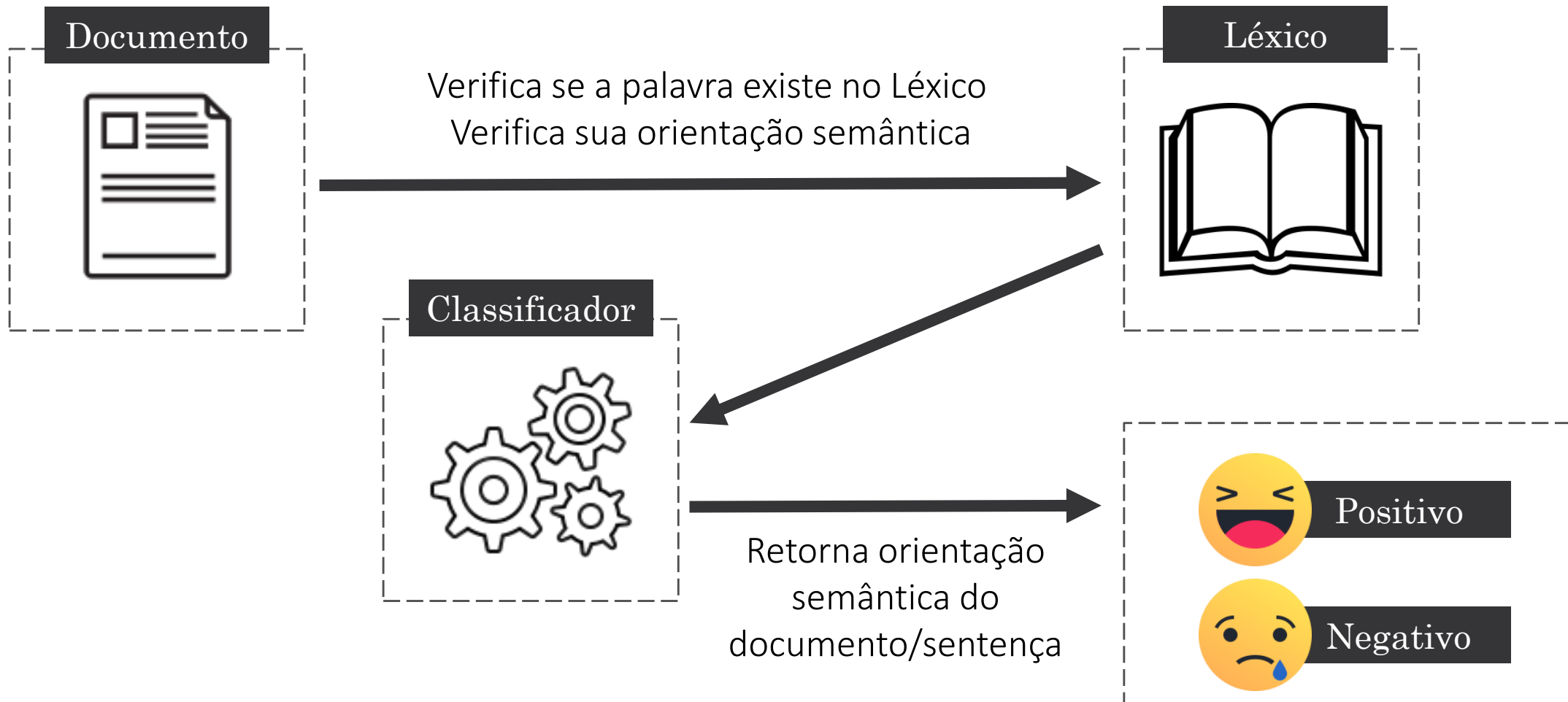
Análise de sentimentos

- Léxicos

- Como podemos observar, Léxicos são extremamente importantes para o correto funcionamento da Análise de Sentimentos;
- O classificador consulta o Léxico para processar o documento/sentença e retornar a orientação semântica do mesmo;
- Léxicos incorretos levam a resultados inconsistentes.



Análise de sentimentos



Programação Genética



Programação genética

Como computadores podem resolver problemas sem serem explicitamente programados para tal?



Programação genética

- *Como computadores podem resolver problemas sem serem explicitamente programados para tal?*
- Evolução de programas computacionais
 - Analogias com mecanismos utilizados da evolução biológica natural;
- Criação (automatizada) de um programa que resolve um determinado problema.



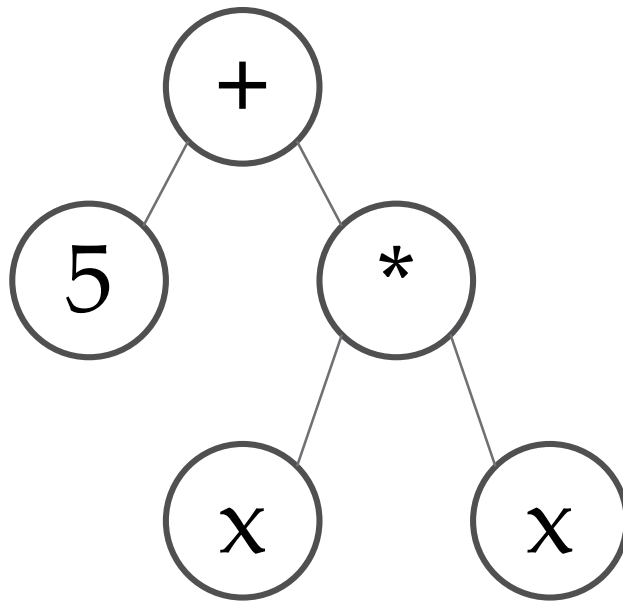
Programação genética

- *Como computadores podem resolver problemas sem serem explicitamente programados para tal?*
 - Pode ser vista como uma extensão dos AG's
 - Indivíduos são programas;
 - Espaço de busca são todos os possíveis programas.



Programação genética

- Programas?
 - Funções matemáticas, por exemplo;
 - Representação feita por meio de árvores.



Exemplo programa:
 $x^2 + 5$



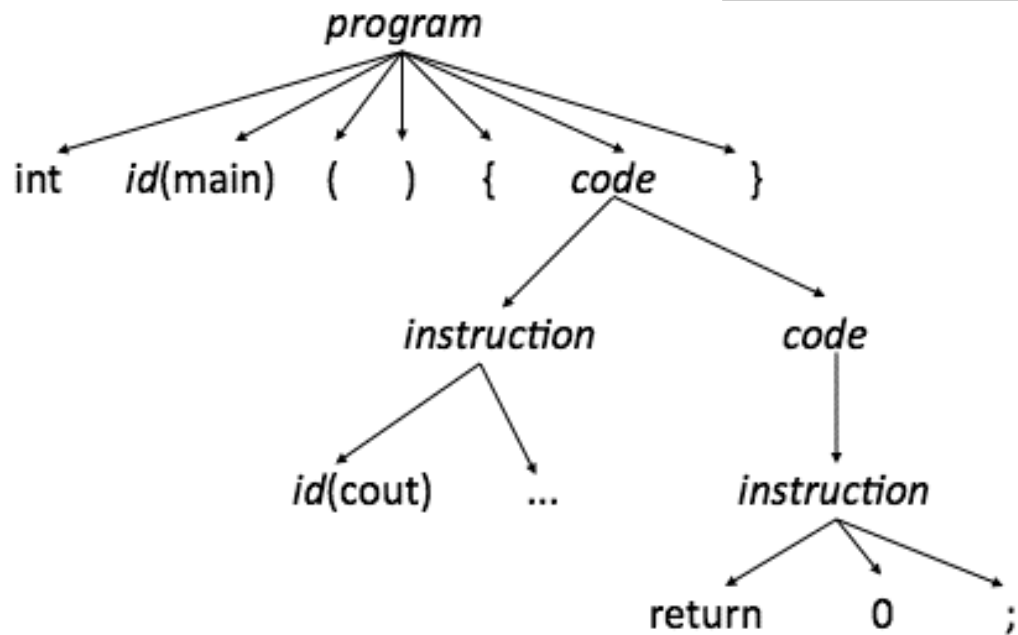
Programação genética

- Intimamente ligada à ideia de programação funcional (sequência de aplicação de funções a argumentos)
 - Independentemente da linguagem, todos os programas podem ser vistos como uma sequência de aplicações de funções a argumentos;
 - Compiladores usam esse fato para traduzir um programa em uma árvore sintática.

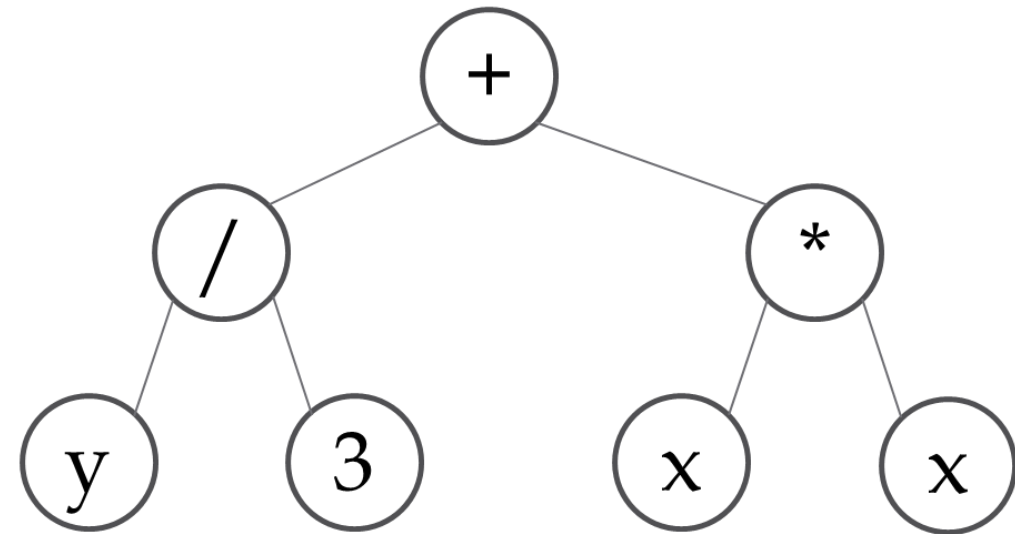


Programação genética

Árvore sintática



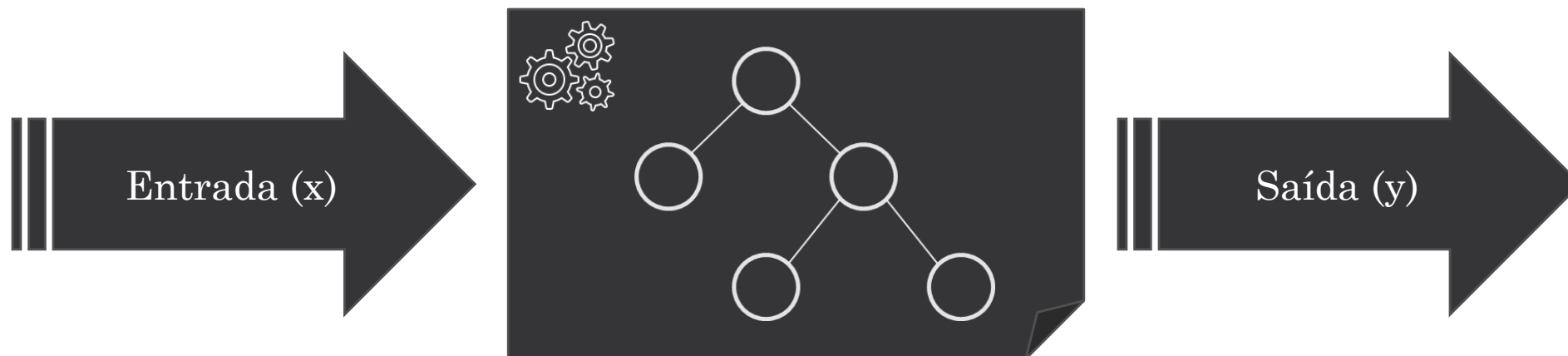
Expressão matemática





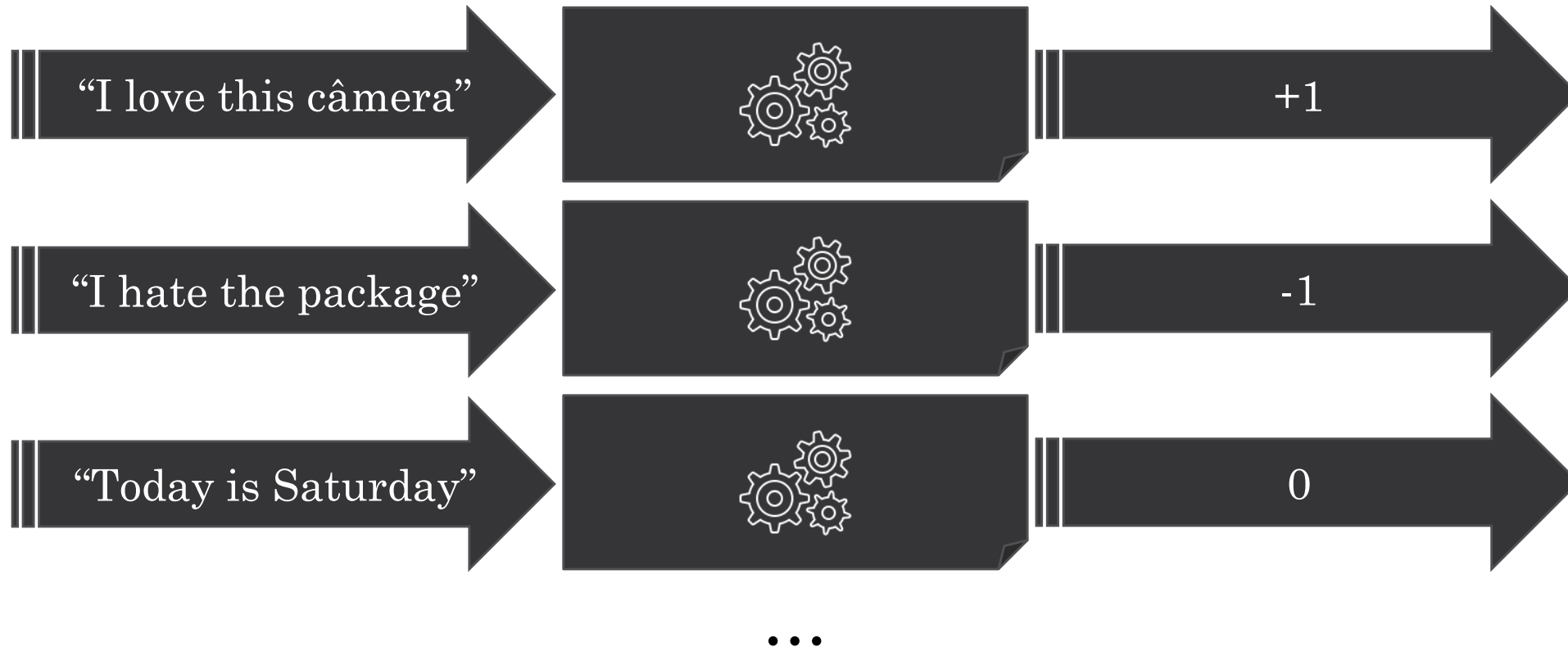
Programação genética

- Modelo M
 - Relaciona um vetor de entrada com um vetor de saída;
 - Assume-se que o modelo é desconhecido.





Programação genética





Programação genética

- Passos para o correto funcionamento [KOZA, 1992]
 1. Determinar conjunto de terminais;
 2. Determinar conjunto de funções;
 3. Determinar função *fitness*;
 4. Determinar parâmetros e variáveis para controle da execução;
 5. Determinar critério de parada.



Programação genética

- Library DEAP - Distributed Evolutionary Algorithms in Python;
- Computer Vision and Systems Laboratory (CVSL) at Université Laval, in Quebec city, Canada;



Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau and Christian Gagné, “DEAP: Evolutionary Algorithms Made Easy”, Journal of Machine Learning Research, pp. 2171-2175, no 13, jul 2012.



François-Michel De Rainville, Félix-Antoine Fortin, Marc-André Gardner, Marc Parizeau and Christian Gagné, “DEAP: A Python Framework for Evolutionary Algorithms”, Companion proc. of the Genetic and Evolutionary Computation Conference (GECCO 2012), July 2012.



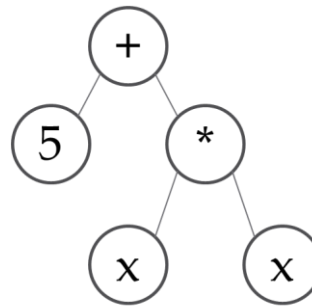
Programação genética

- Criação da população
 - Cria uma população de forma randômica;
 - Profundidade máxima definida por parâmetro;
- Principais métodos
 - Full;
 - Grow;
 - Ramped half-and-half.



Programação genética

- Criação da população
 - Método Grow
 - Respeita o critério de profundidade máxima da árvore;
 - Escolhe aleatoriamente entre funções e terminais em qualquer nível da árvore, podendo criar estruturas irregulares.

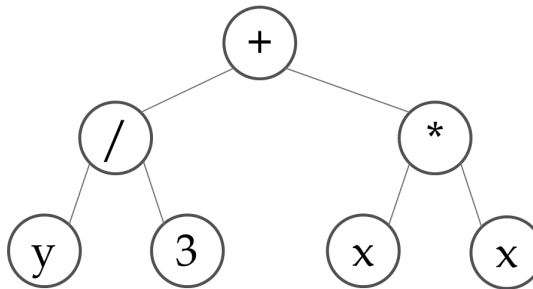


```
toolbox.register("expr", gp.genGrow, pset=pset, min_=1, max_=7)
```



Programação genética

- Criação da população
 - Método Full
 - Árvores com a profundidade máxima;
 - Escolhe aleatoriamente somente funções, até que um nó de profundidade máxima seja atingido, aí então escolhendo somente terminais.

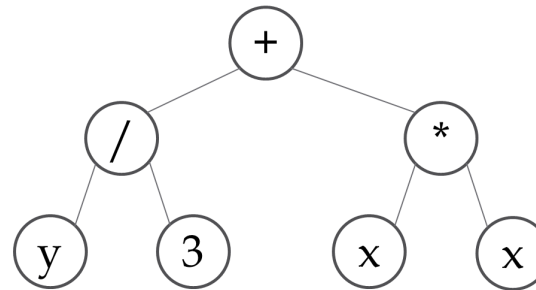
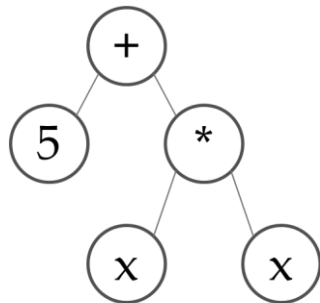


```
toolbox.register("expr", gp.genFull, pset=pset, min_=1, max_=7)
```



Programação genética

- Criação da população
 - Método Ramped half-and-half
 - Utiliza o método Grow e Full;
 - Gera um número igual de árvores para cada profundidade;
 - 50% utilizará o método full e 50% o método Grow.

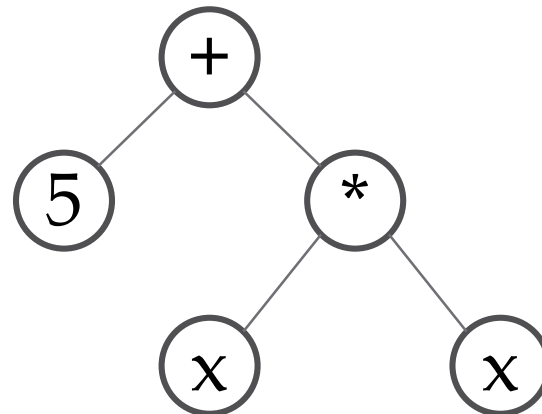


```
toolbox.register("expr", gp.genHalfAndHalf, pset=pset, min_=1, max_=7)
```



Programação genética

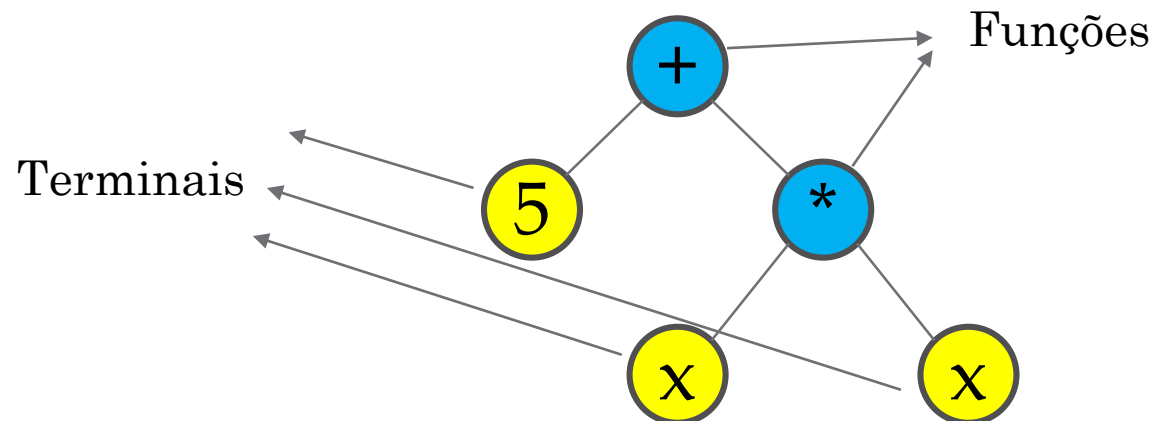
- Funções e terminais
 - **Funções:** funções aritméticas (+, -, /, *), funções booleanas, funções matemáticas, etc;
 - **Terminais:** constantes numéricas, dados externos, variáveis.





Programação genética

- Funções e terminais
 - **Funções:** funções aritméticas (+, -, /, *), funções booleanas, funções matemáticas, etc;
 - **Terminais:** constantes numéricas, dados externos, variáveis.





Programação genética

- Funções e terminais

```
pset.addPrimitive(operator.add, [float, float], float)
pset.addPrimitive(operator.sub, [float, float], float)
pset.addPrimitive(operator.mul, [float, float], float)
pset.addPrimitive(protectedDiv, [float, float], float)
pset.addPrimitive(math.cos, [float], float)
pset.addPrimitive(math.sin, [float], float)
```

```
pset.addPrimitive(protectedLog, [float], float)
pset.addPrimitive(invertSignal, [float], float)
```

```
pset.addPrimitive(positiveHashtags, [str], float)
pset.addPrimitive(negativeHashtags, [str], float)
pset.addPrimitive(polaritySum, [str], float)
```

```
pset.addEphemeralConstant("r", lambda: float(random.randint(-1,1)), float)
```





Programação genética

- Operadores genéticos

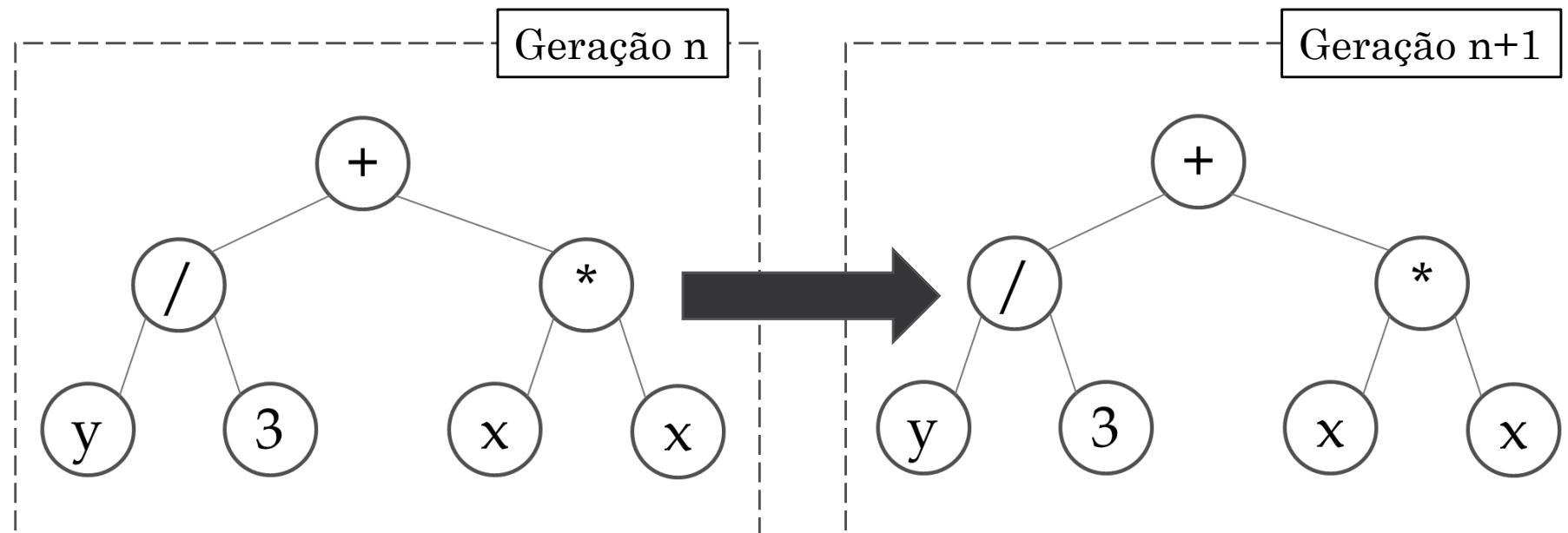
- Reprodução;
- Crossover;
- Mutação;
- Permutação;
- Edição;
- Encapsulamento;
- Destruição.



Operadores genéticos

- Reprodução

- Um indivíduo com uma bom valor após função de avaliação (*fitness*) é escolhido;
- É feita uma cópida idêntica do indivíduo para a próxima geração.

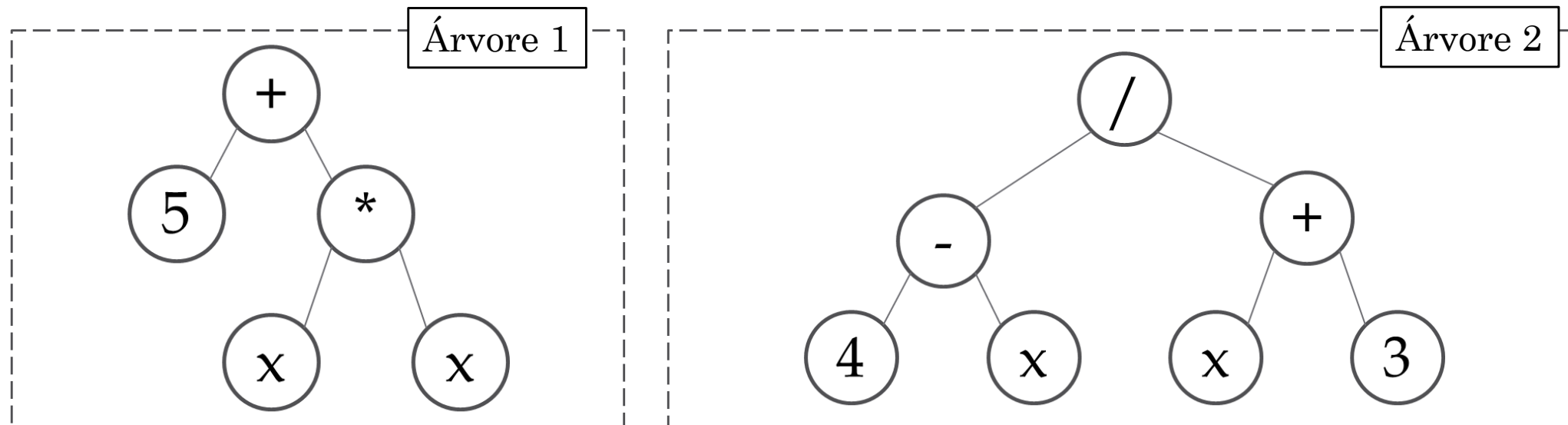




Operadores genéticos

- Crossover

- Troca entre partes dos indivíduos selecionados;
- Partes escolhidas de forma aleatória nas duas árvores.



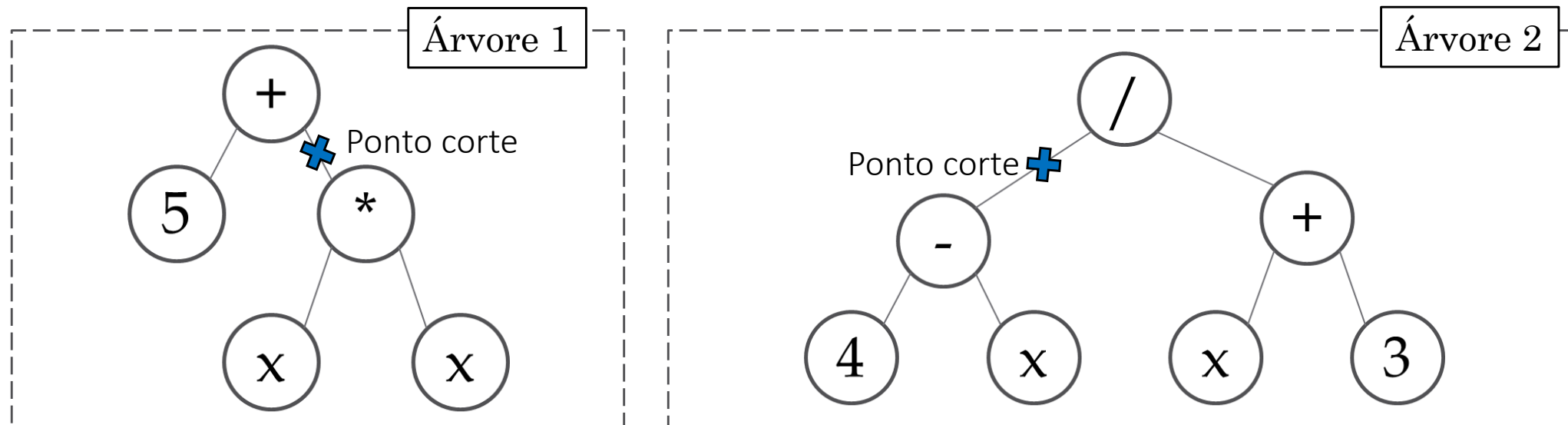
```
toolbox.register("mate", gp.cxOnePoint)
```



Operadores genéticos

- Crossover

- Troca entre partes dos indivíduos selecionados;
- Partes escolhidas de forma aleatória nas duas árvores.



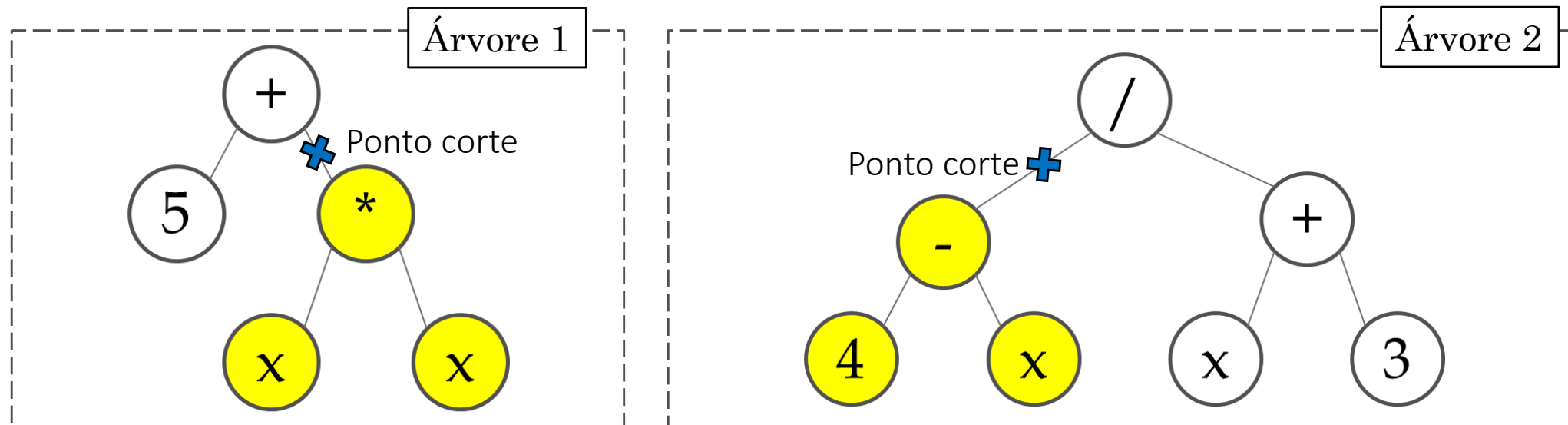
```
toolbox.register("mate", gp.cxOnePoint)
```



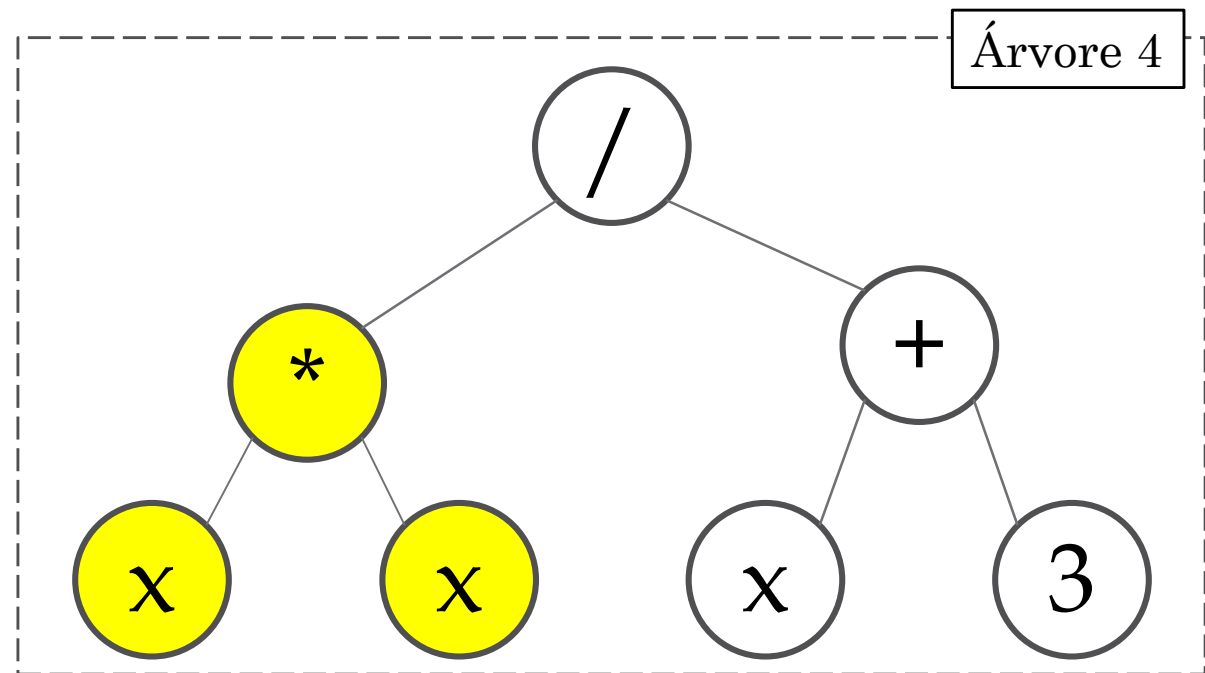
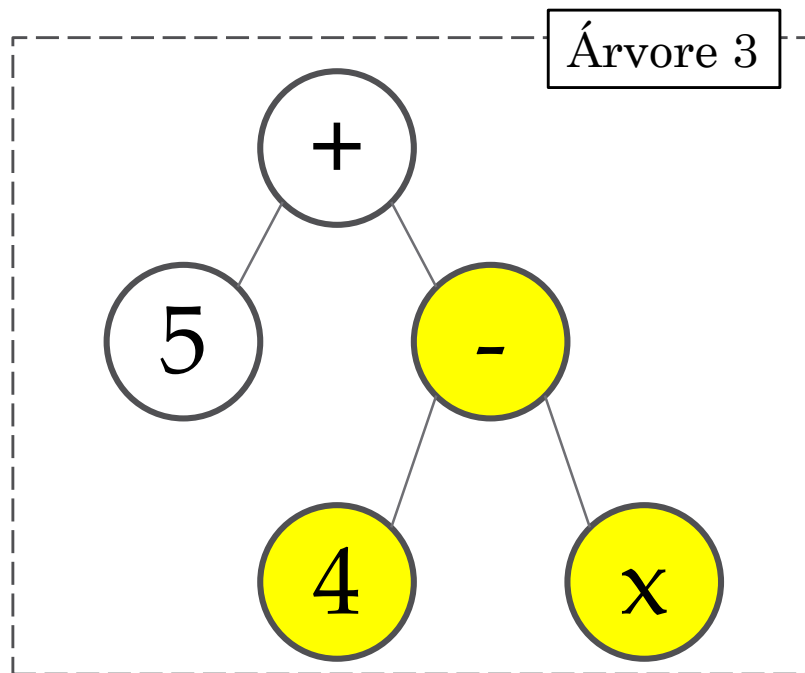
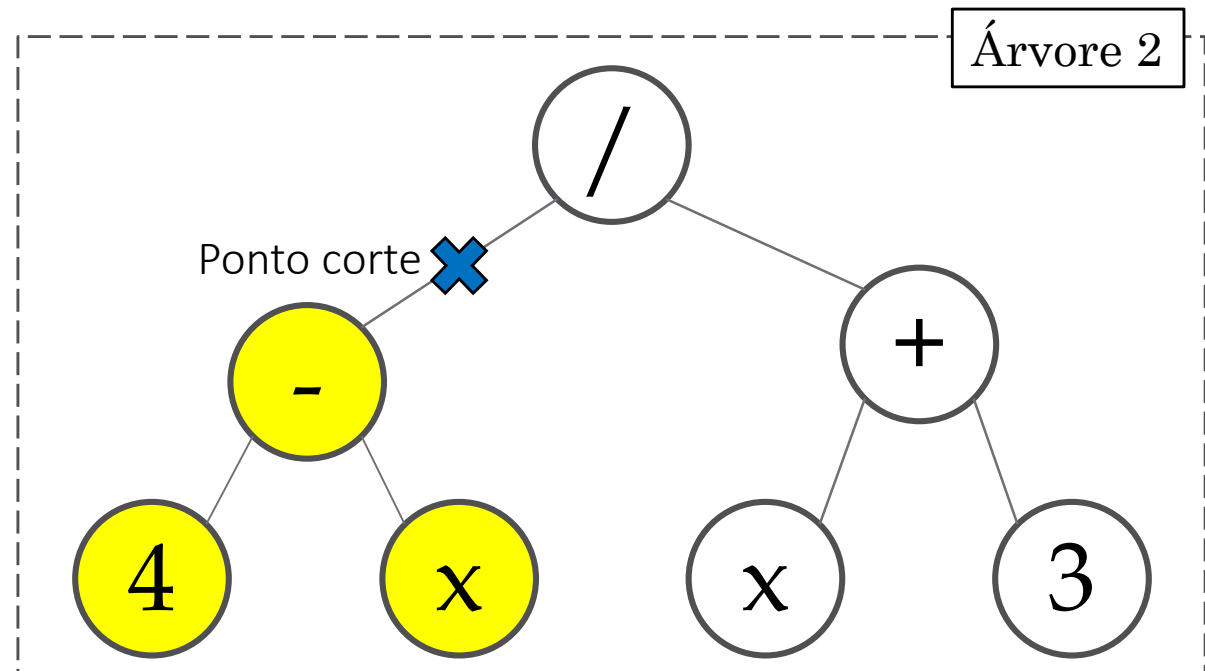
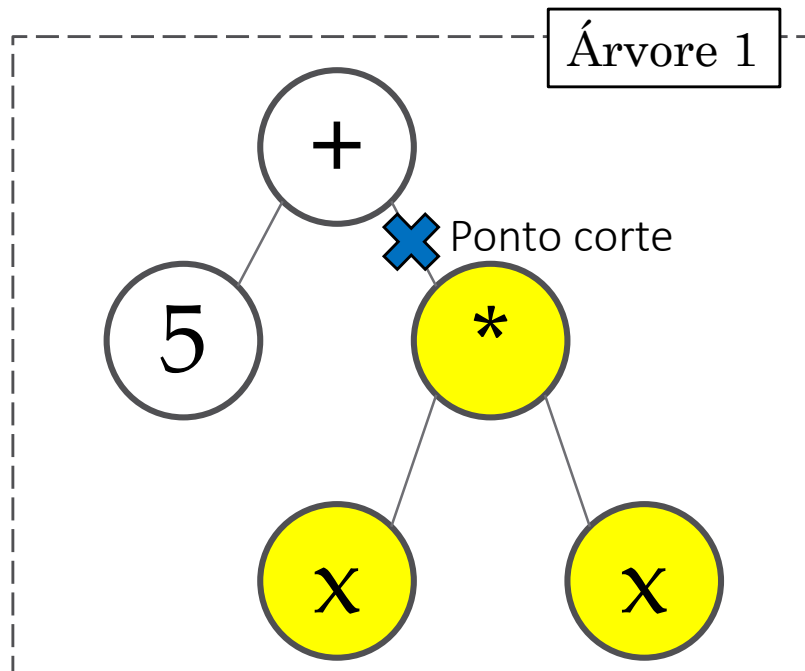
Operadores genéticos

- Crossover

- Troca entre partes dos indivíduos selecionados;
- Partes escolhidas de forma aleatória nas duas árvores.



```
toolbox.register("mate", gp.cxOnePoint)
```

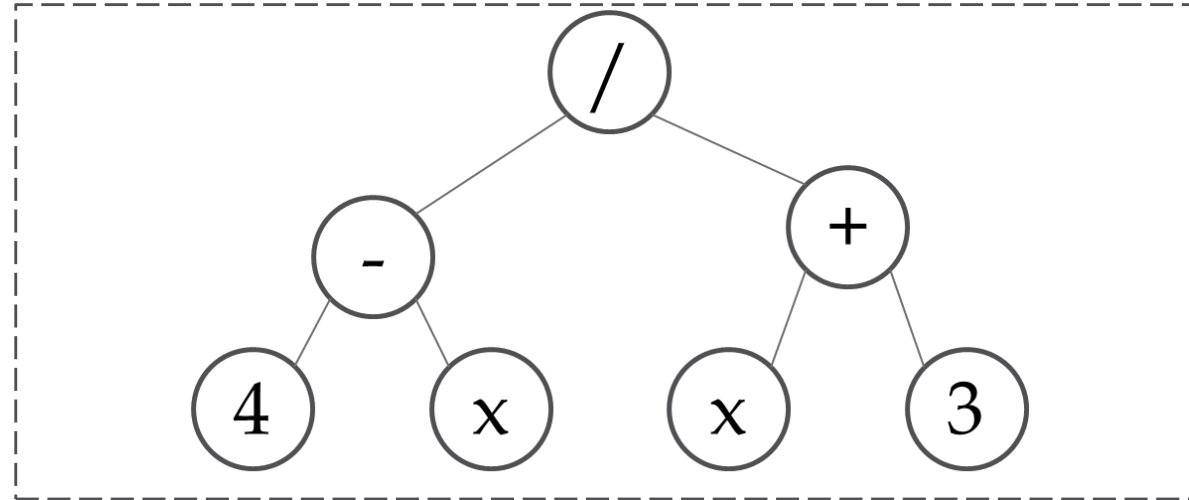




Operadores genéticos

- Mutaç o

- Mudan a aleat ria em um dos n os da  rvore;
- Adiciona diversidade na popula  o.



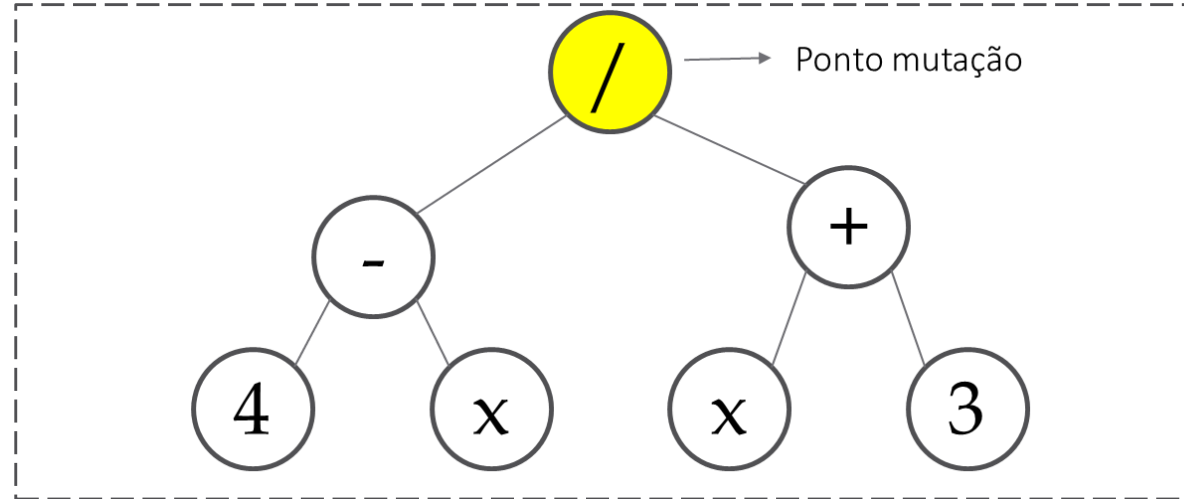
```
toolbox.register("mutate", gp.mutUniform, expr=toolbox.expr_mut, pset=p)
```



Operadores genéticos

- Mutaç o

- Mudan a aleat ria em um dos n os da  rvore;
- Adiciona diversidade na popula  o.



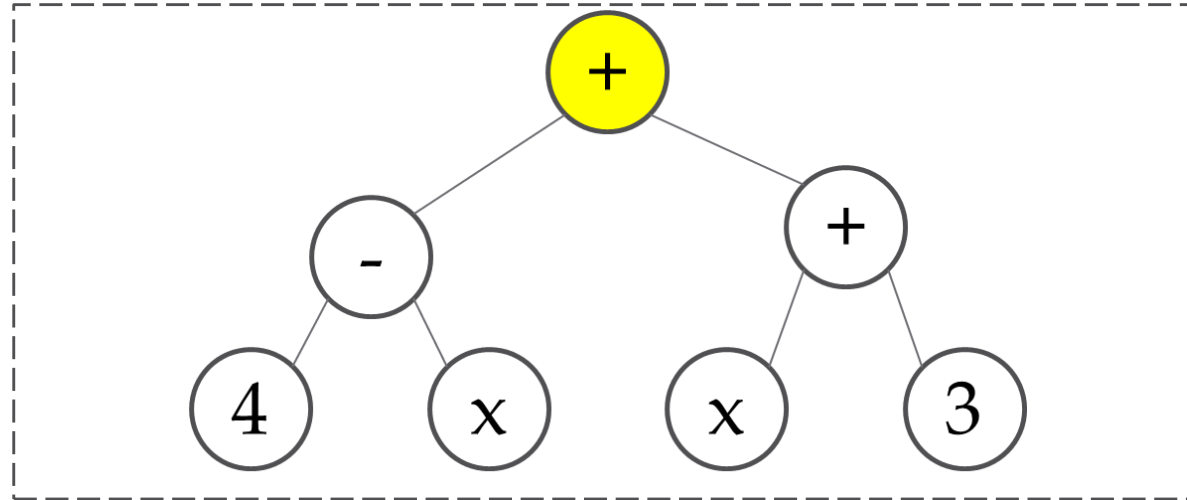
```
toolbox.register("mutate", gp.mutUniform, expr=toolbox.expr_mut, pset=p)
```




Operadores genéticos

- Muta  o

- Mudan  a aleat  ria em um dos n  s da   rvore;
- Adiciona diversidade na popula  o.



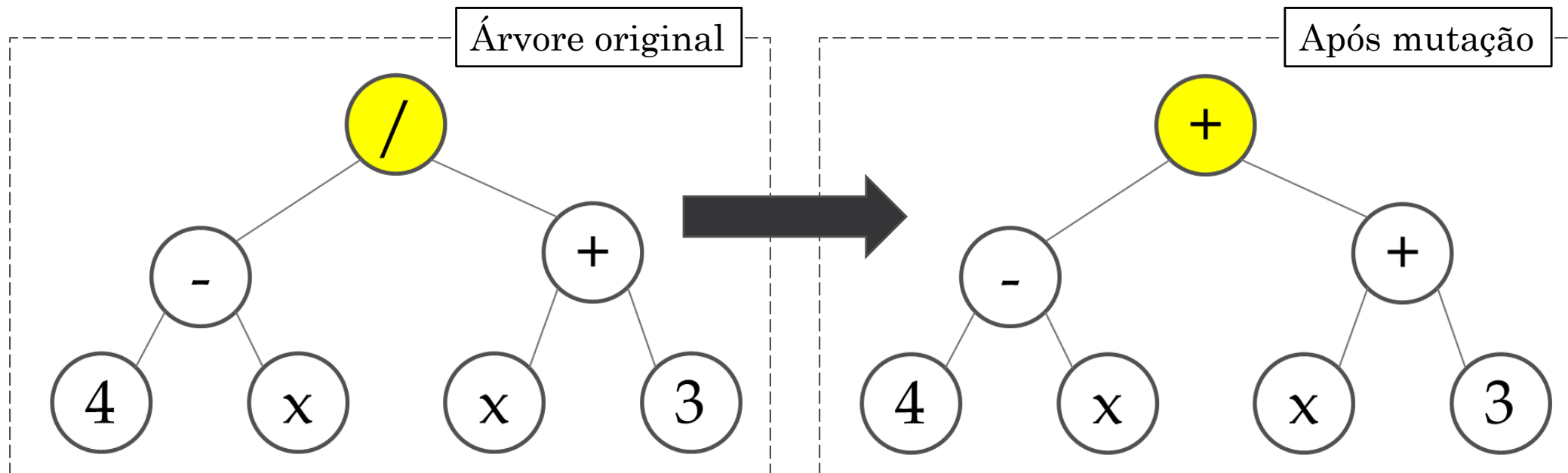
```
toolbox.register("mutate", gp.mutUniform, expr=toolbox.expr_mut, pset=p)
```



Operadores genéticos

- Mutaç o

- Mudan a aleat ria em um dos n os da  rvore;
- Adiciona diversidade na popula  o.





Operadores genéticos

- Mutaç o

- Mudan a aleat ria em um dos n os da  rvore;
- Adiciona diversidade na popula  o.



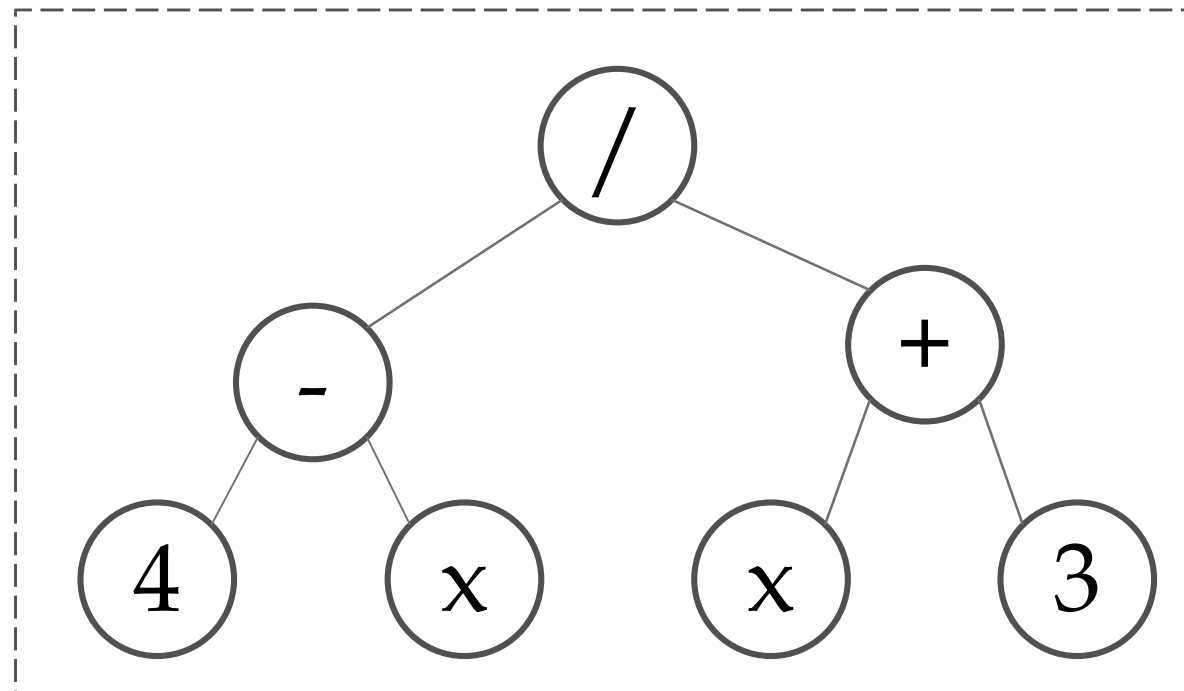
```
toolbox.register("mutate", gp.mutUniform,  
    expr=toolbox.expr_mut, pset=pset)  
toolbox.decorate("mutate",  
    gp.staticLimit(key=operator.attrgetter("height"),  
    max_value=17))
```



Operadores genéticos

- Permutação

- Escolhe um ponto aleatório e inverte os terminais e/ou funções.

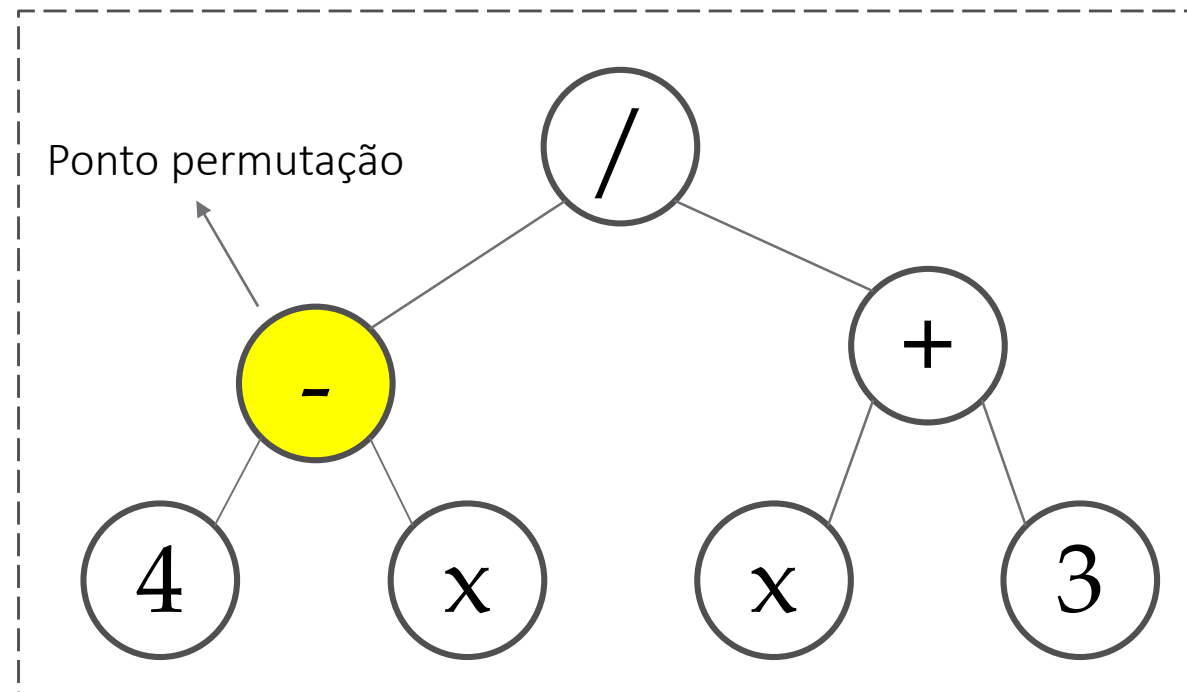




Operadores genéticos

- Permutação

- Escolhe um ponto aleatório e inverte os terminais e/ou funções.

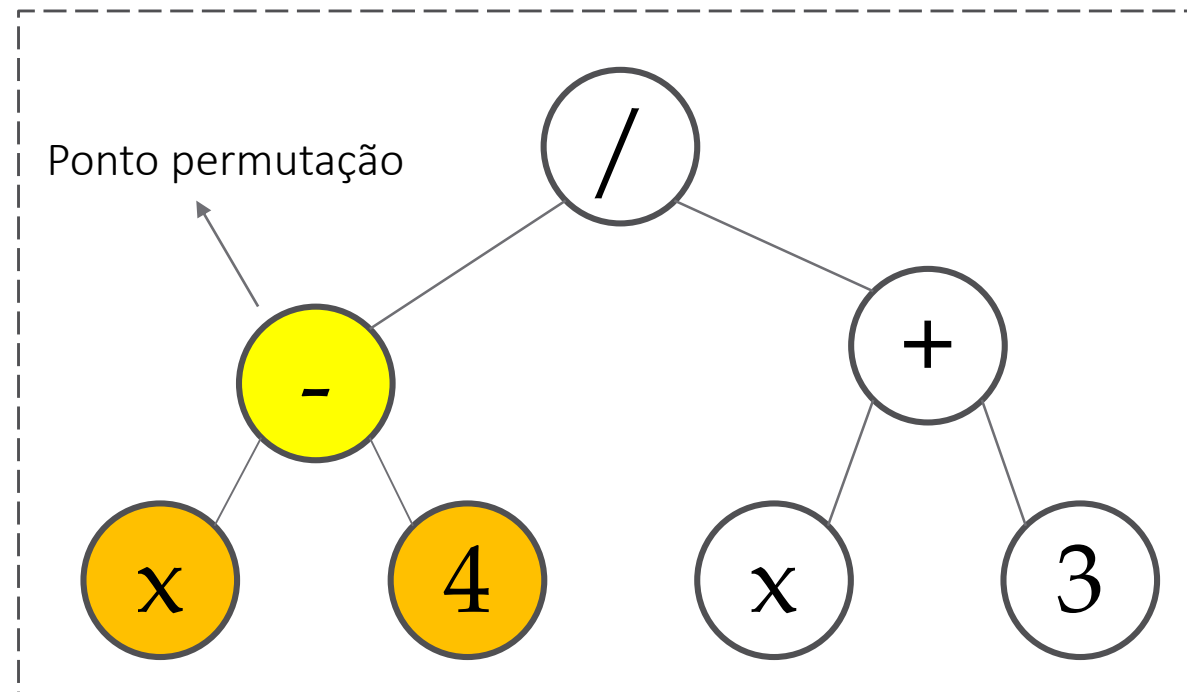




Operadores genéticos

- Permutação

- Escolhe um ponto aleatório e inverte os terminais e/ou funções.

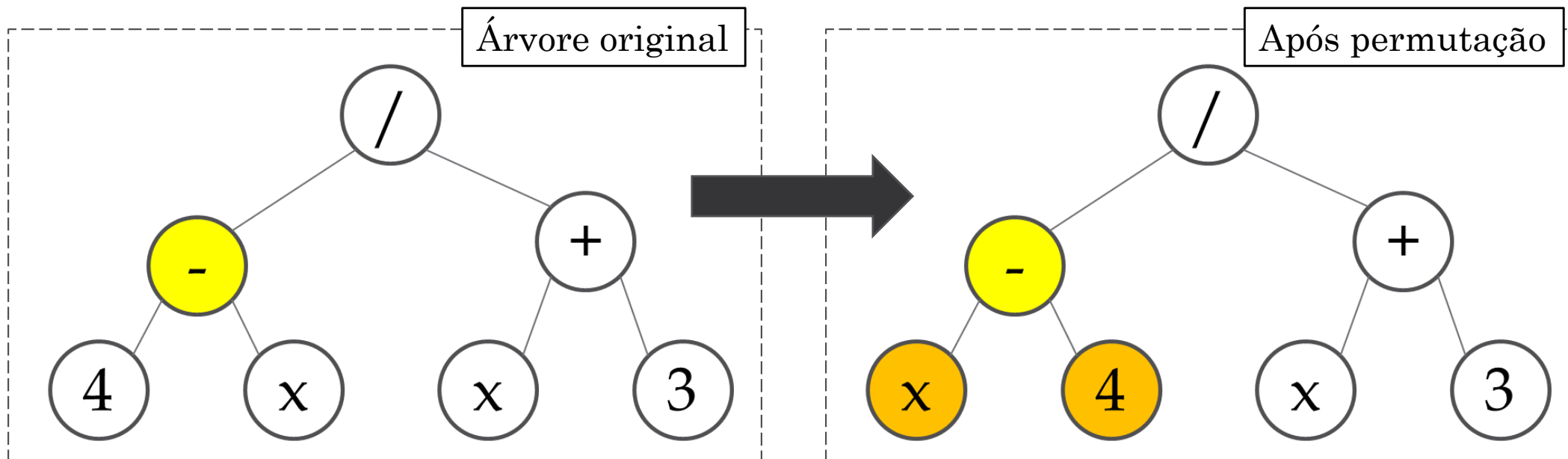




Operadores genéticos

- Permutação

- Escolhe um ponto aleatório e inverte os terminais e/ou funções.

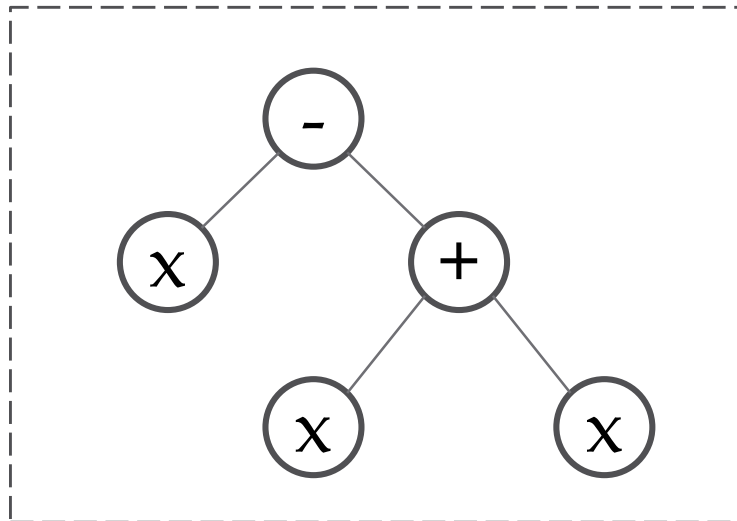




Operadores genéticos

- Edição

- Forma de simplificação e edição de expressões;
- Muito custosa – Consumo considerável de tempo;
- Torna a expressão menos vulnerável ao crossover.



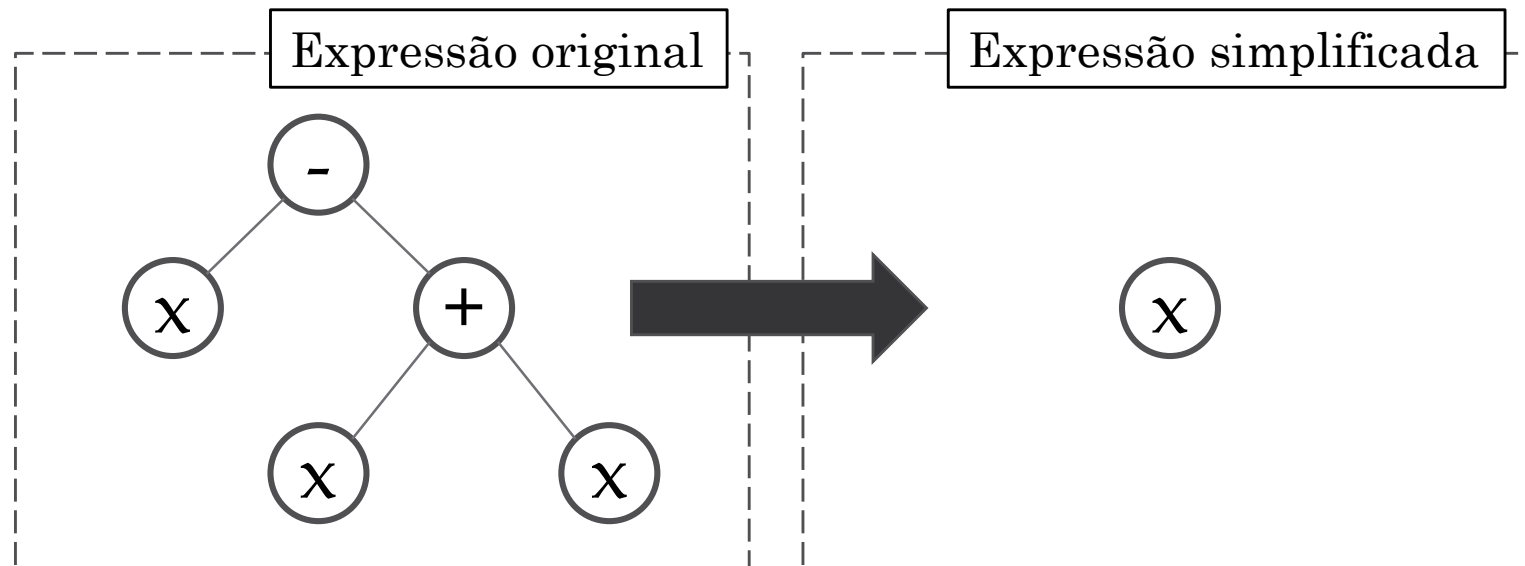
Expressão:
 $X+X-X$



Operadores genéticos

- Edição

- Forma de simplificação e edição de expressões;
- Muito custosa – Consumo considerável de tempo;
- Torna a expressão menos vulnerável ao crossover.

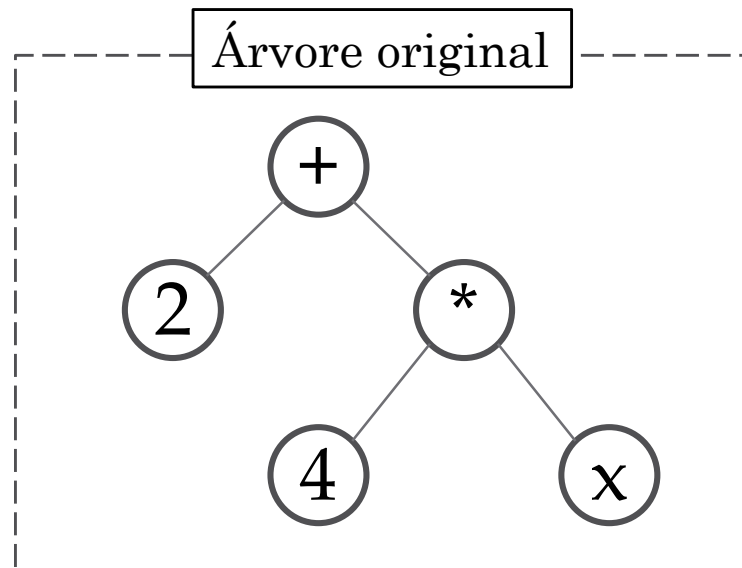




Operadores genéticos

- Encapsulamento

- Identifica subárvores potencialmente útil;
- Dá um nome para que possa ser referenciada futuramente.

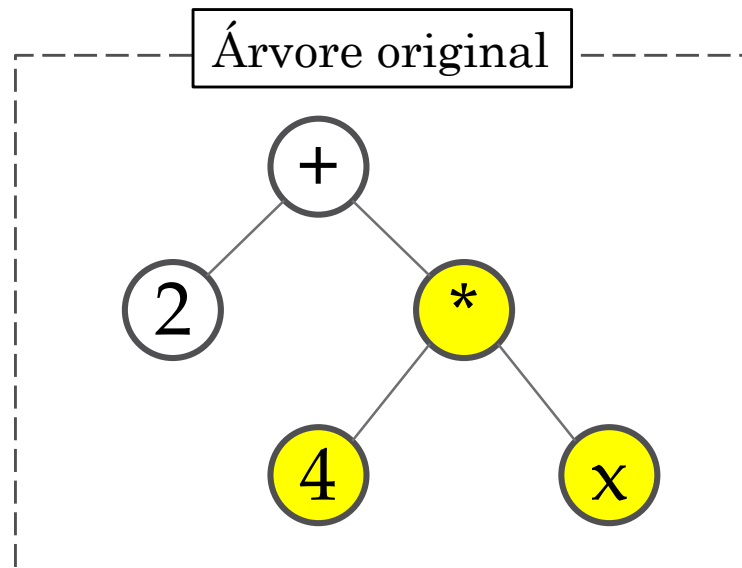




Operadores genéticos

- Encapsulamento

- Identifica subárvores potencialmente útil;
- Dá um nome para que possa ser referenciada futuramente.

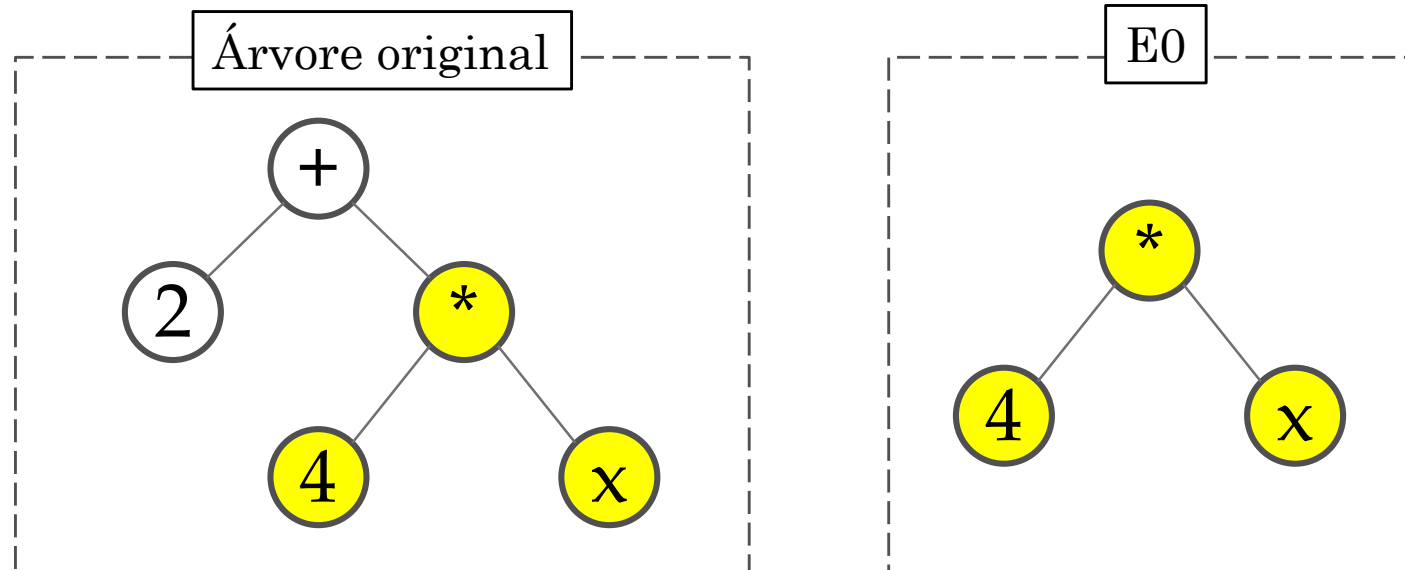




Operadores genéticos

- Encapsulamento

- Identifica subárvores potencialmente útil;
- Dá um nome para que possa ser referenciada futuramente.

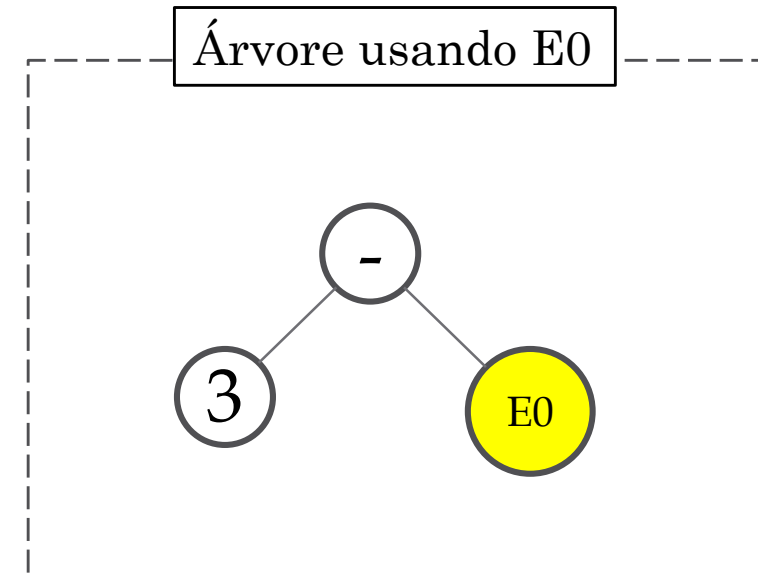
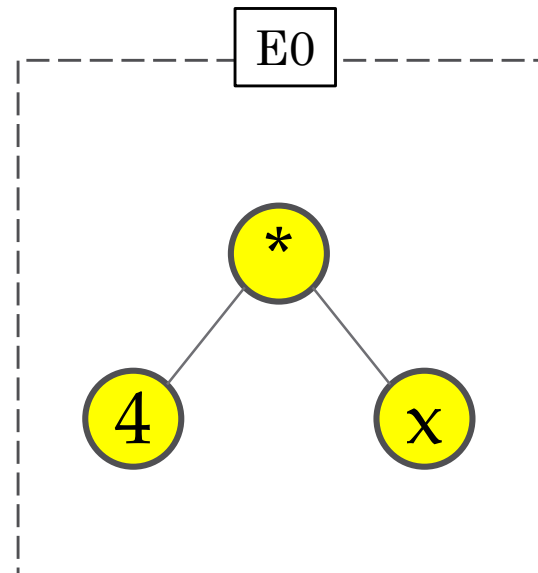
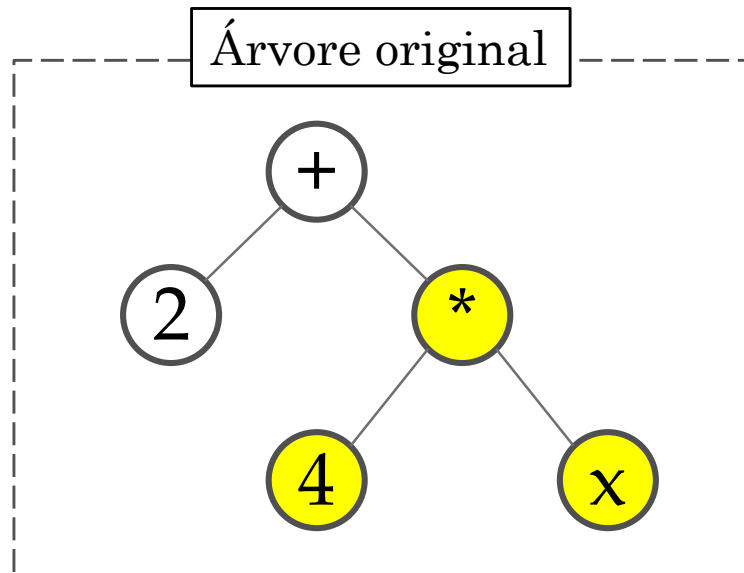




Operadores genéticos

- Encapsulamento

- Identifica subárvores potencialmente útil;
- Dá um nome para que possa ser referenciada futuramente.





Operadores genéticos

- Destruição

- Casos complexos, grande parte da população pode ter um *fitness* muito ruim, causando uma perda de diversidade rápida e um custo computacional muito grande;
- Forma de destruir indivíduos medíocres nas gerações iniciais;
- Parâmetros
 - Quantidade de indivíduos mantidos;
 - Condição em que o operador será invocado;
- Indivíduos sobreviventes são escolhidos com base no *fitness*.

Estratégia de solução



Estratégia de solução

- Ideia original
 - Atribuição da orientação semântica das palavras de um Léxico utilizando Algoritmos Genéticos;
 - Abordar a criação/expansão do dicionário Léxico como um problema de otimização.

I	love	this	camera
0	+1	0	-2
+3	+2	-4	-1
-2	-1	+1	-2
...			




Estratégia de solução


- Ideia original



Keshavarz, Hamidreza, and Mohammad Saniee Abadeh. "ALGA: Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs." Knowledge-Based Systems 122 (2017): 1-16.

**ELSEVIER**

Knowledge-Based Systems
Volume 122, 15 April 2017, Pages 1–16




ALGA: Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs

ACM **DL** DIGITAL LIBRARY

ALGA

Authors: [Hamidreza Keshavarz](#) [Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran](#)
[Mohammad Saniee Abadeh](#) [Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran](#)

Published in:
• Journal
Knowledge-Based Systems [archive](#)

 2017 Article

	good	was	loveable	water	...
Chromosome 1	+8	-6	+7	-2	...
Chromosome 2	+1	0	+10	-1	...
Chromosome 3	-9	-1	+1	+8	...

$$P(T_i, k) = \sum_{w_j \in T_i} v_k(w_j)$$

<http://www.sciencedirect.com/science/article/pii/S0950705117300382>



Estratégia de solução

- Ideia original
 - Apesar de utilizar Algoritmos Genéticos para a atribuição de orientação semântica para as palavras do dicionário, o trabalho faz a classificação com a soma das polaridades

$$P(T_i, k) = \sum_{w_j \in T_i} v_k(w_j)$$

<http://www.sciencedirect.com/science/article/pii/S0950705117300382>



Estratégia de solução

- Ideia atual
 - Usar Algoritmos Evolutivos para encontrar um modelo eficiente para a classificação de sentimentos
 - Tentar encontrar uma solução

$$P(T_i, k) = \textcircled{?}$$



Estratégia de solução

- Ferramentas e materiais
 - DEAP *library* (<https://github.com/deap/deap>);
 - Textos anotados
 - Amazon *Reviews* (LIU);
 - Tweets.
 - Dicionários Léxicos
 - Positive/Negative *words* (LIU);
 - Positive/Negative *emoticons* (SentiHealth);
 - SentiWordNet (em implementação).



Estratégia de solução



- Avaliações (Amazon) anotadas manualmente; [LIU, 2004]
- Conjunto de frases e suas polaridades $[-n, +n]$.

```
[t]excellent for the semi-serious amateur  
camera[-3]##i found that low light  
situations combined with any sort of action  
left this camera in the dust.
```



Estratégia de solução

- Avaliações (Amazon) anotadas manualmente; [LIU, 2004]
- Conjunto de frases e suas polaridades $[-n, +n]$.



[t]excellent for the semi-serious amateur
camera [-3] ## i found that low light
situations combined with any sort of action
left this camera in the dust.





Estratégia de solução



- Conjunto de *tweets* e suas orientações semânticas;
- 0: negativo; 1: positivo

```
18, 1, Sentiment140, Feeling strangely  
fine. Now I'm gonna go listen to some  
Semisonic to celebrate
```



Estratégia de solução



- Conjunto de *tweets* e suas orientações semânticas;
- 0: negativo; 1: positivo

18, 1, Sentiment140, Feeling strangely
fine. Now I'm gonna go listen to some
Semisonic to celebrate

ID sentimento source

Início frase



Estratégia de solução



[+2]##I love this câmera so much!

[-2]##I hate this câmera so much!

[+1]## Great!

[-1]##I hate the package

[+1]##Another phrase!

...



Estratégia de solução

[+2]##I love this câmera so much!

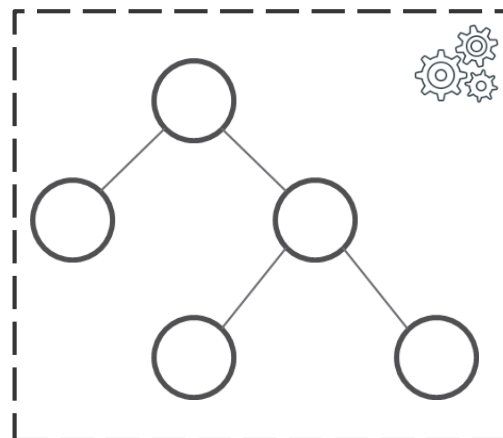
[-2]##I hate this câmera so much!

[+1]## Great!

[-1]##I hate the package

[+1]##Another phrase!

...



positive

negative

positive

negative

positive

...



Estratégia de solução

[+2]##I love this câmera so much!

[-2]##I hate this câmera so much!

[+1]## Great!

[-1]##I hate the package

[+1]##Another phrase!

...



positive

negative

positive

negative

positive

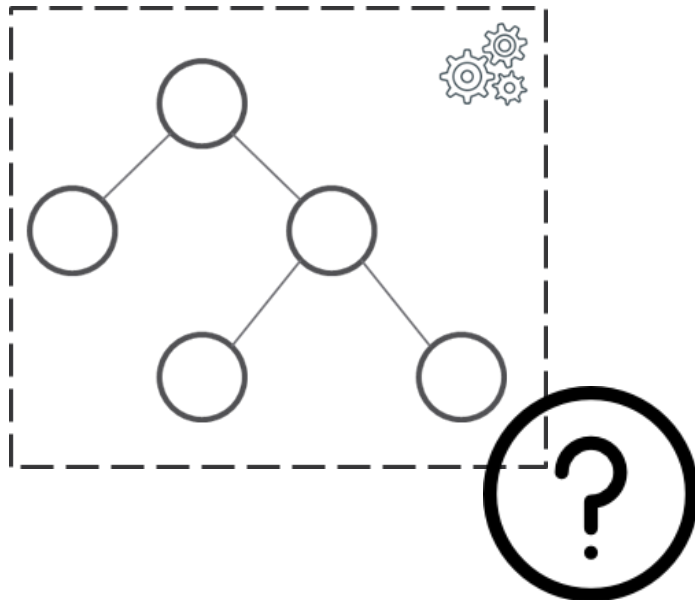
...



Estratégia de solução

- Ideia principal

- Encontrar um modelo para a classificação das opiniões que avalie corretamente a polaridade dos documentos/sentenças.



- **Programação Genética** para a criação dos modelos;
- Representação da solução como uma árvore;
- Funções matemáticas e de manipulação do texto para tentar chegar em um modelo ótimo;

Estratégia de solução



tweets.txt



reviews.txt



positive-words.txt



negative-words.txt



positive-emoticons.txt



negative-emoticons.txt



Estratégia de solução

```
addPrimitive(operator.add, [float,float], float)
addPrimitive(operator.sub, [float,float], float)
addPrimitive(operator.mul, [float,float], float)
addPrimitive(protectedDiv, [float,float], float)
addPrimitive(math.exp, [float], float)
addPrimitive(math.cos, [float], float)
addPrimitive(math.sin, [float], float)
addPrimitive(protectedSqrt, [float], float)
addPrimitive(protectedLog, [float], float)
addPrimitive(invertSignal, [float], float)

addPrimitive(positiveHashtags, [str], float)
addPrimitive(negativeHashtags, [str], float)
addPrimitive(polaritySum, [str], float)
addPrimitive(positiveWordsQuantity, [str], float)
addPrimitive(negativeWordsQuantity, [str], float)
pset.addPrimitive(positiveHashtags, [str], float)
pset.addPrimitive(negativeHashtags, [str], float)
pset.addPrimitive(positiveEmoticons, [str], float)
pset.addPrimitive(negativeEmoticons, [str], float)
```





Estratégia de solução

- *Fitness*

- Quantidade de frases com polaridades calculadas corretamente;
- Melhor caso: 100% das palavras avaliadas preditas corretamente.

```
def evalSymbReg(individual):  
    for item in enumerate(reviews):  
  
        if correctValue(individual, item):  
            fitnessReturn += 1  
  
    return fitnessReturn,
```

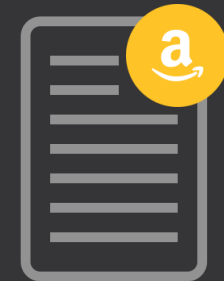


Resultados



Resultados parciais

Population: 20
Generations: 50
Mating probability: 1.5
Mutation probability: 0.5
Variation: varAnd (crossover and mutation)
Selection type: tournament - size 3
Creation: Half and Half - size [1, 7]



reviews.txt

```
[42 phrases] [39 matches (fitness)] [257 seconds]
    sub(polaritySum(x), invertSignal(sub(cos(1), sin(add(0, 1)))))

[239 phrases] [184 matches (fitness)] [806 seconds]
    exp(sin(add(invertSignal(protectedSqrt(-1.1160456169186785)),
    protectedDiv(invertSignal(-0.43733403591853515),
    positiveWordsQuantity(x)))))

[239 phrases] [188 matches (fitness)] [751 seconds]
    add(add(polaritySum(x), log(log(-2.0))), 1.0)

[239 phrases] [202 matches (fitness)] [406 seconds]
    protectedDiv(1.5159468201145594, polaritySum(x))
```



Resultados parciais

```
Population: 20
Generations: 50
Mating probability: 1.5
Mutation probability: 0.5
Variation: varAnd (crossover and mutation)
Selection type: tournament - size 3
Creation: Half and Half - size [1, 7]
```



tweets.txt

```
[207 phrases] [138 matches (fitness)] [458 seconds]
      add(add(add(-1.3538745456927384, -1.0854354436218143),
      negativeHashtags(x)), mul(negativeWordsQuantity(x), -1.3287832809148568))
```



Referências

- ZUBEN, F. V. Representação e Operadores Evolutivos
- ZUBBEN, F. B. Programação Genética
- KOZA, J.R. Genetic Programming: On the Programming of Computers by means of Natural Selection
- NETO, A. G. Programação Genética
- CRUZ, A. J. O. Algoritmos Genéticos
- MEDEIROS, D. Programação Genética
- FORTIN, F, RAINVILLE, F, Marc-André GARDNER, M, PARIZEAU, M, GAGNÉ, C. DEAP: Evolutionary Algorithms Made Easy
- FORTIN, F, RAINVILLE, F, Marc-André GARDNER, M, PARIZEAU, M, GAGNÉ, C. DEAP: A Python Framework for Evolutionary Algorithms