

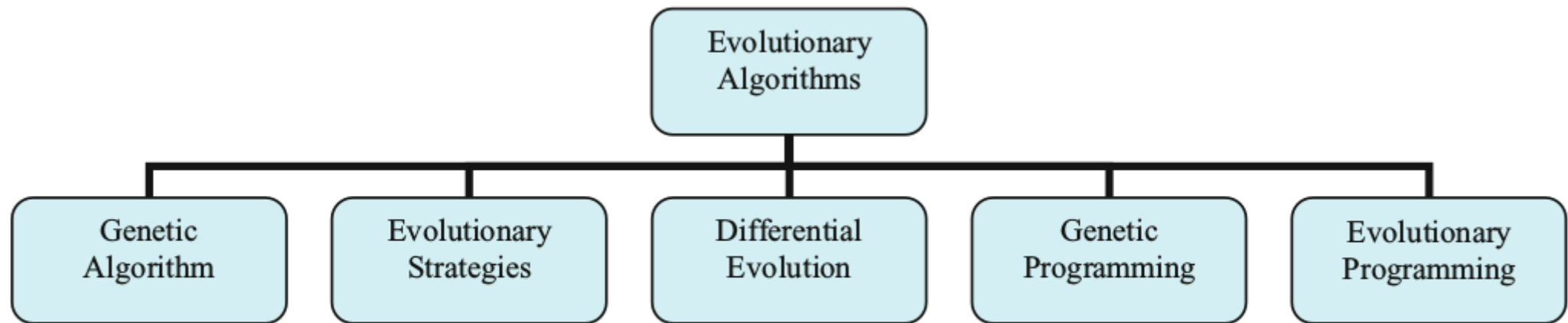
# Computación Bioinspirada

Dr. Edward Hinojosa Cárdenas  
[ehinojosa@unsa.edu.pe](mailto:ehinojosa@unsa.edu.pe)

# Primer Parcial

- Evaluación: Primer Parcial (Semana del 06/05 al 10/05).
- Evaluación Escrita. Traer pliegos de papel A4, todos los útiles de escritorio, calculadora (no celulares).
- Todos los temas vistos en el primer parcial (teoría, práctica y laboratorio).

# Algoritmos Evolutivos



**Fig. 2.2.** Main branches of evolutionary computation

# Programación Genética

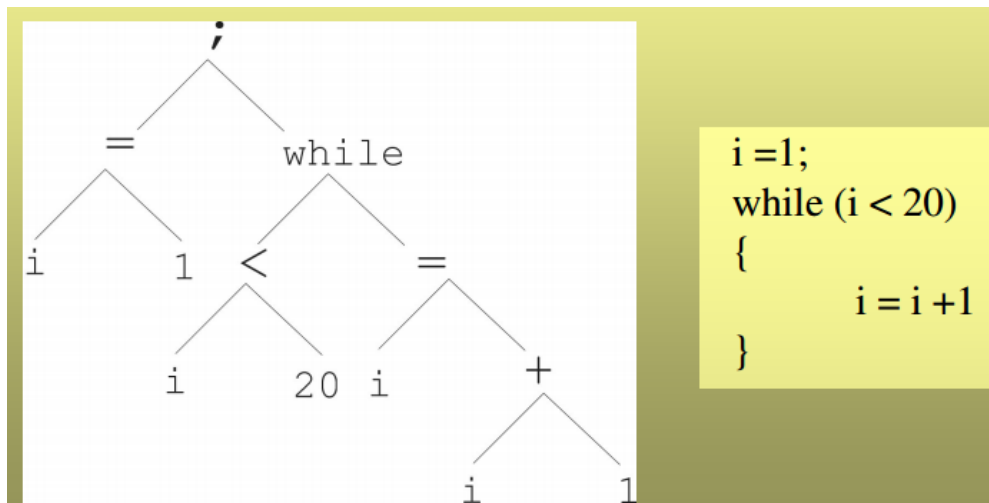
- La Programación Genética (PG) intenta responder una de las preguntas de la ciencia de la computación:
  - Como las computadoras pueden aprender a resolver problemas sin ser explícitamente programadas para tal?
  - Es decir, como las computadoras pueden hacer lo que debe ser hecho sin ser orientados exactamente para hacer ello.

# Programación Genética

- En 1975, John Holland (Ph. D. en Ciencias de la Computación – Universidad de Michigan, 1959) creó los Algoritmos Genéticos.
- En 1992, John Koza (Ph. D. en Ciencias de la Computación – Univesidad de Michigan, 1972) usó los algoritmos genéticos para desarrollar programas para realizar ciertas tareas. Llamó a su método de programación genética.

# Programación Genética

- Programación:
  - Podemos representar un programa en un árbol.



- Genética:
  - Utiliza principios de selección natural.

# Programación Genética

- La PG es una extensión de los AG en el dominio de los programas, donde:
  - El individuo es un programa de computador.
  - El espacio de búsqueda son todos los posibles programas de computador.
- En resumen, la PG es un método de búsqueda, dentro de un espacio significativamente polinomial/exponencial y restricto de programas de computador, de una solución exacta o por lo menos aproximada para resolver determinado problema.

# Programación Genética

- Un programa de computador es una extensión matemática compuesta de funciones y terminales.
- Las funciones pueden ser operaciones aritméticas (+, -, \*, ...) operaciones booleanas (and, or, not, ...), funciones matemáticas (sen, cos, ...), operadores condicionales (if, then, ...), funciones de iteración (while, ...), funciones que causan recursión, funciones específicas del problema.
- Los terminales pueden ser variables (representando, posiblemente, las entradas) o constantes (5).



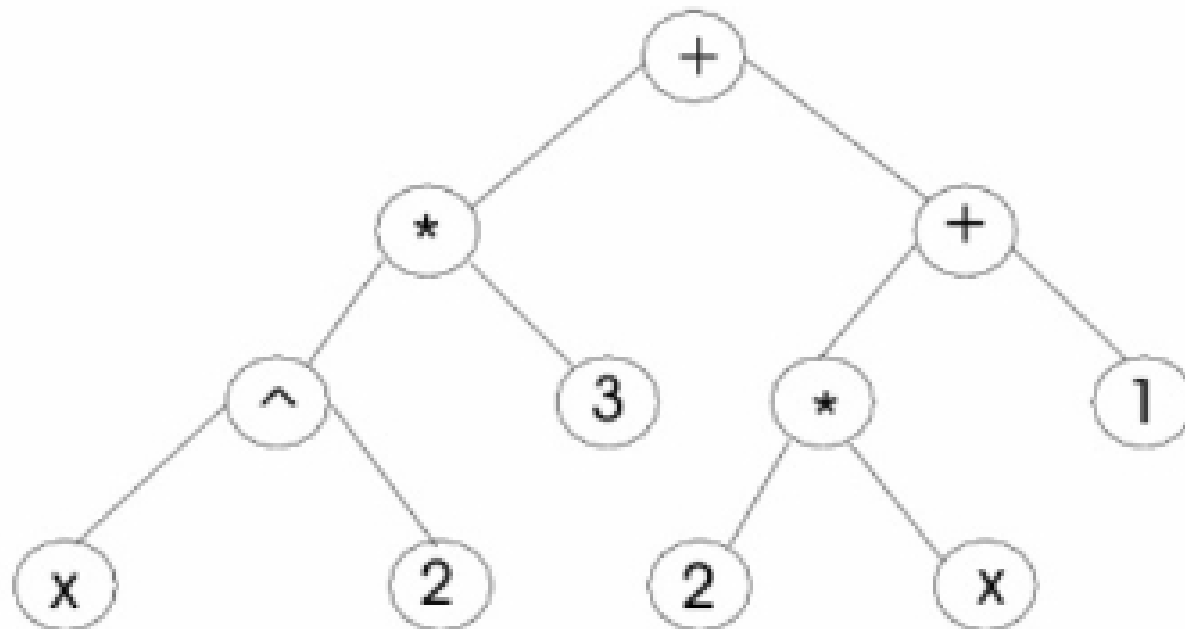
# Programación Genética

- Los programas de computador, en PG, son representado bajo una forma de árboles.
- Las funciones aparecen en los nodos internos del árbol.
- Los terminales aparecen en los nodos fuera de los árboles.

# Programación Genética

- Por ejemplo:

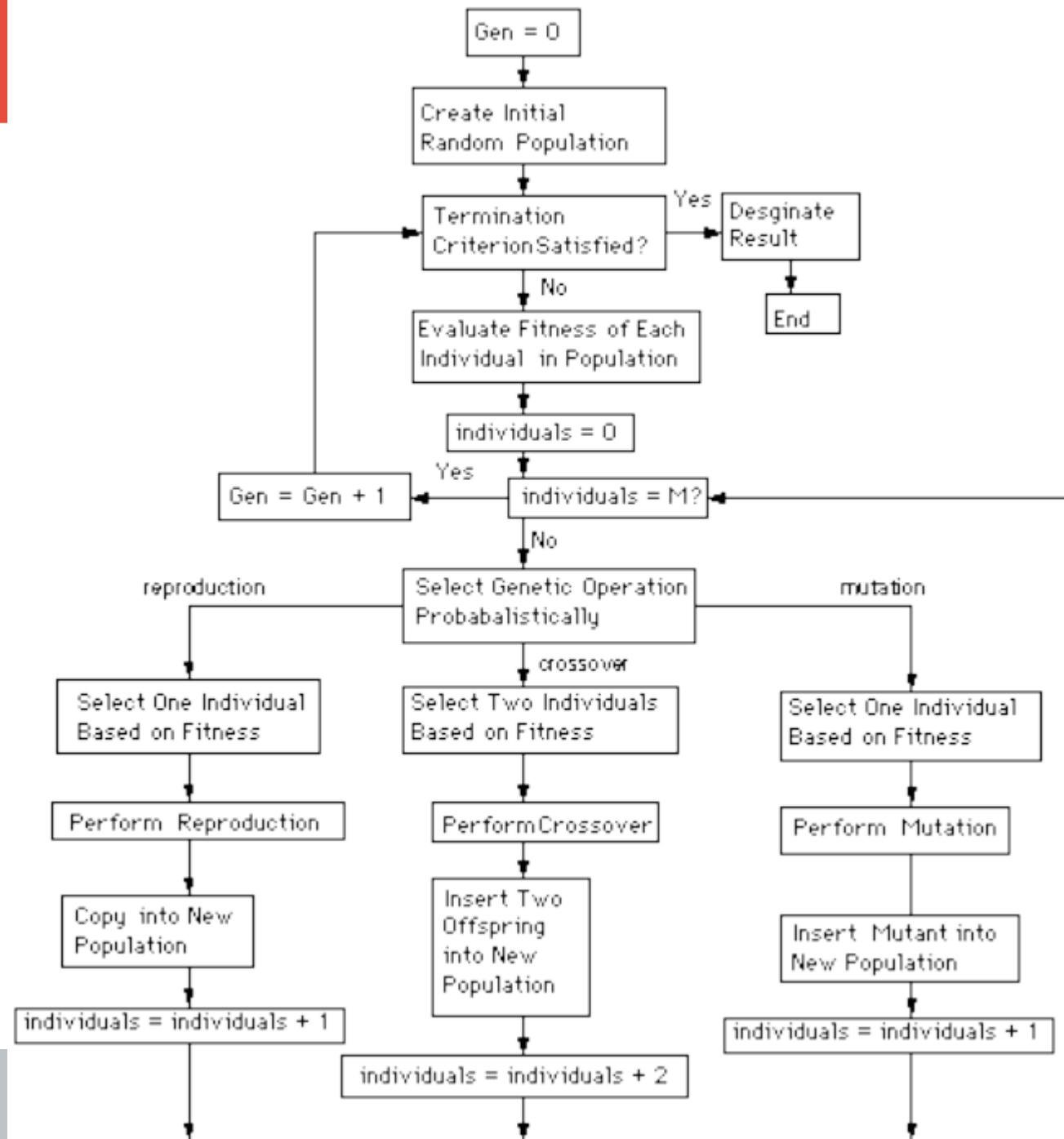
$$3x^2+2x+1$$



# Programación Genética

- La PG envuelve programas de computador a partir de los conjuntos de terminales y funciones.
- Un proceso de PG sigue le siguiente flujo. Considere probabilidades para los diferentes para:
  - Reproducción (10% a 20%)
  - Cruzamiento (70% a 80%)
  - Mutación (10% a 20%)

# Flowchart for Genetic Programming



# Programación Genética - Aptitud

- Cada individuo tiene asociado a si una medida numérica, que es el resultado de la interacción con el ambiente.
- Es decir, es una medida de grado de adaptación del individuo.
- Está relacionada al proceso evolutivo, haciendo que tenga mayor probabilidad de que sus características sean propagadas y permanecer en las siguientes poblaciones.

# Programación Genética - Aptitud

- Ejemplo:

Input	Output
10	29
42	125
3	8

- Para la tabla anterior, que muestra una entrada y una salida, se requiere encontrar una función aritmética que realice esta asignación.

# Programación Genética - Aptitud

- El valor de aptitud para un individuo es la suma de las diferencias entre las salidas esperadas (listadas en la tabla anterior) y las salidas reales cuando se evalúa el individuo con uno de los tres valores de entrada.
- Cuanto menor sea el valor de la aptitud, mejor será el candidato.

# Programación Genética - Aptitud

- Definimos el conjunto de funciones y terminales:
- Un conjunto de funciones que contiene los operadores aritméticos de suma (+), resta (-) y multiplicación (\*).
- Conjunto de terminales que contiene una sola variable denominada  $x$ . Y un conjunto constante que contiene los valores enteros 0, 1, 2, 3, 4 y 5.



# Programación Genética - Aptitud

- Considerando un individuo:

$(+ (+ 3 2) (* 1 1))$

Candidate 1:  $(+ (+ 3 2) (* 1 1))$

Input	Expected	Actual	Difference
10	29	6	23
42	125	6	119
3	8	6	2

Fitness =  $23 + 119 + 2 = 144$

# Programación Genética - Aptitud

- Considerando otro individuo:

(+ (- x x) (+ x x))

Candidate 2: (+ (- x x) (+ x x))

Input	Expected	Actual	Difference
10	29	20	9
42	125	84	41
3	8	6	2

Fitness = 9 + 41 + 2 = **52**

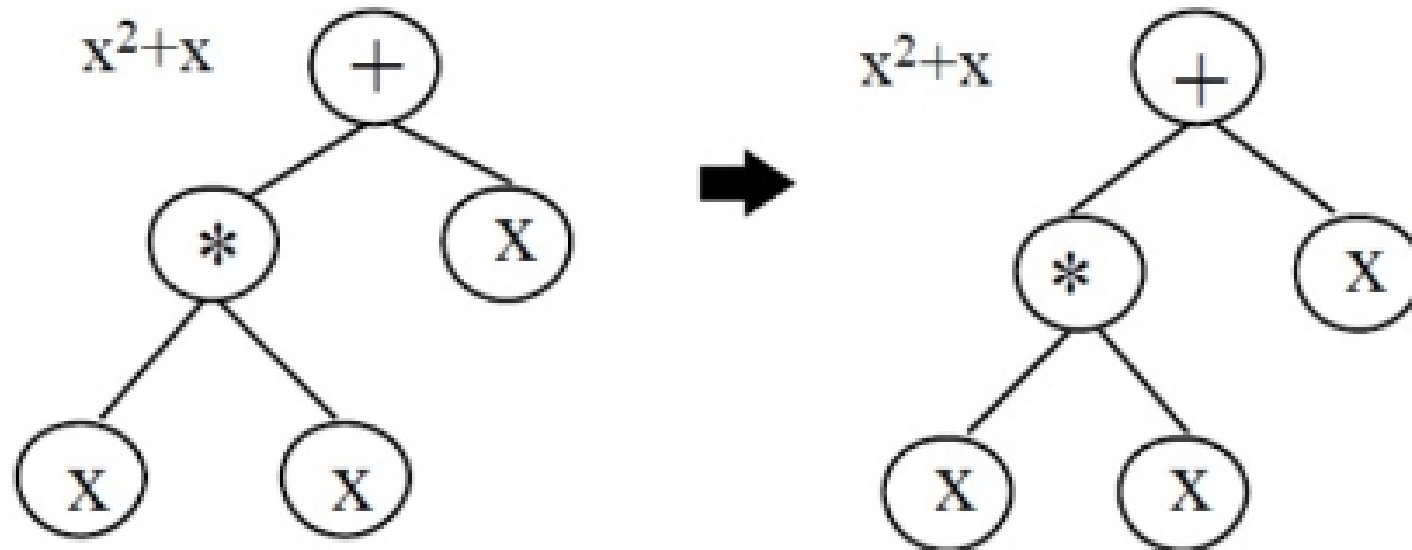
# Programación Genética - Aptitud

- Para los dos individuos:

Rank	Candidate	Fitness
1st	$(+ (- \times \times) (+ \times \times))$	52
2nd	$(+ (+ 3 2) (* 1 1))$	144

# Programación Genética - Reproducción

- Un individuo de la población es seleccionado de acuerdo con algún método basado en la aptitud.
- El individuo es copiado, sin ninguna alteración, para la próxima generación.



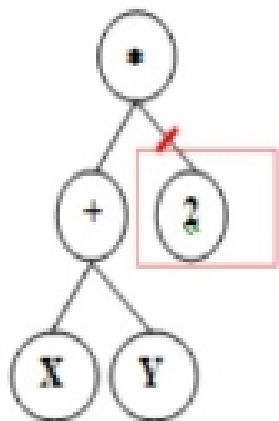
# Programación Genética - Cruzamiento

- Cambia el material genético entre dos individuos.
- Es seleccionado un punto de corte en los dos árboles y las ramas inferiores son intercambiadas.

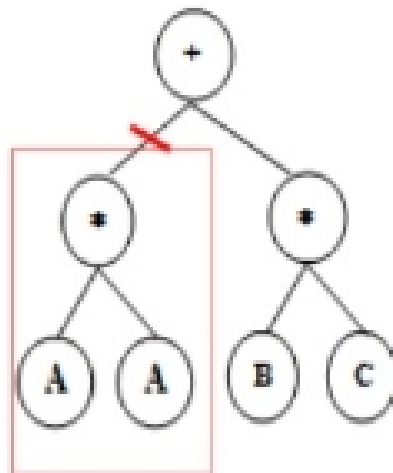
Padres

Hijos

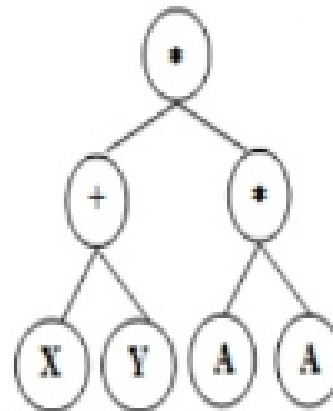
$(*(+ X Y) 2)$



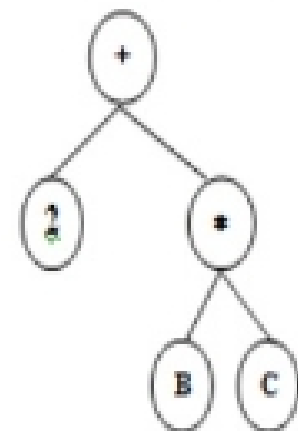
$(+(* A A) (* B C))$



$(*(+ X Y) (* A A))$



$(+(2) (* B C))$

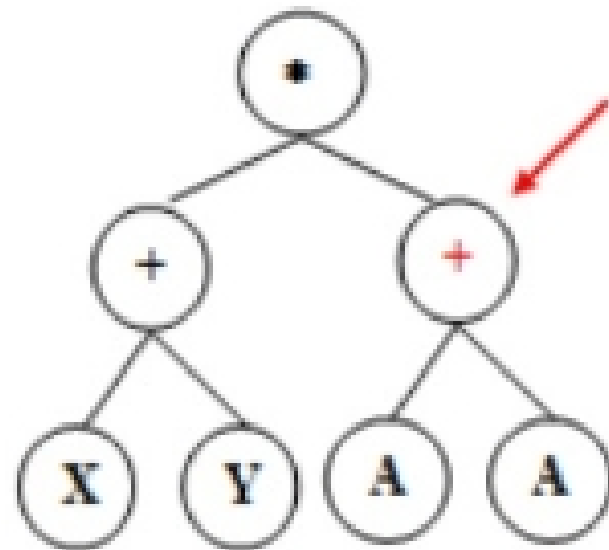
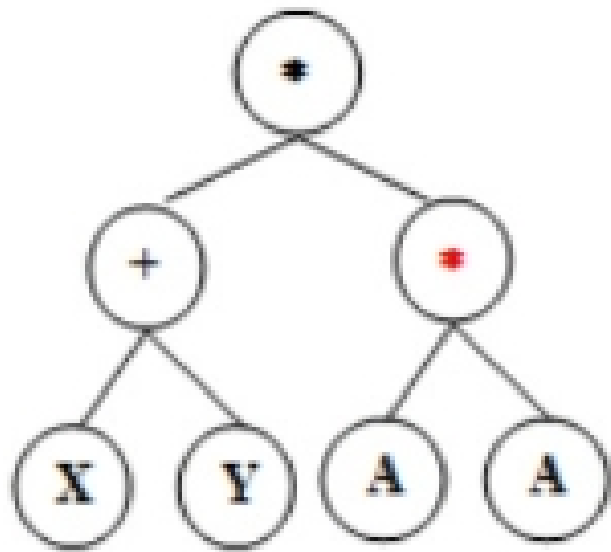


# Programación Genética - Cruzamiento

- Ejemplo (para el ejemplo anterior):
- Padre 1:  $(+ (- x x) (+ x x))$
- Padre 2:  $(+ (+ 3 2) (* 1 1))$
- Hijo 1:  $(+ (- x x) (* 1 1))$
- Hijo 2:  $(+ (+ 3 2) (+ x x))$

# Programación Genética - Mutación

- La mutación consiste en un cambio aleatorio de una función, una entrada (o una constante) en el árbol.



# Programación Genética - Mutación

- Ejemplo (para el ejemplo anterior):
- Individuo 1: (+ (- x x) (+ x x))
- Individuo Mutado: (+ (+ x x) (+ x x))



# GRACIAS

Dr. Edward Hinojosa Cárdenas  
[ehinojosa@unsa.edu.pe](mailto:ehinojosa@unsa.edu.pe)