

**ORGANISMOS DIGITALES, ROBOTS Y
MODELOS COMPUTACIONALES DE
ORGANISMOS OPERANTES.**

LECTURAS PARA UN SEMINARIO



AVIDA

Jaime Ernesto Vargas Mendoza



ASOCIACIÓN
OAXAQUEÑA DE
PSICOLOGÍA A.C.

Organismos digitales, robots y modelos computacionales de organismos operantes: Apuntes para un seminario

Vargas-Mendoza, Jaime Ernesto.

© 2006. Asociación Oaxaqueña de Psicología A.C.

Calzada Madero 1304, Centro, Oaxaca de Juárez, Oaxaca,

México. C.P. 68000

Tel. (951)5142063, (951) 5495923, Fax. (951) 5147646

www.conductitlan.net

E-mail: jorgeever@yahoo.com.mx, comentarios@conductitlan.net

Se promueve la reproducción parcial o total de este documento citando la fuente y sin fines de lucro.

En caso de citar este documento por favor utiliza la siguiente referencia:

Vargas-Mendoza, J. E. (2006) Organismos digitales, robots y modelos computacionales de organismos operantes: Apuntes para un seminario México: Asociación Oaxaqueña de Psicología A.C.

CONTENIDO :

ORGANISMOS DIGITALES .

1. BREVE HISTORIA DE LA VIDA DIGITAL.
Claus O. Wilke y Christoph Adami
2. EL METABOLISMO COMPUTACIONAL.
Claus O. Wilke y Christoph Adami
3. LA BIOLOGIA DE LOS ORGANISMOS DIGITALES.
Claus O. Wilke y Christoph Adami
4. ESTUDIANDO LA EVOLUCION CON ORGANISMOS DIGITALES.
Henry Bortman
5. EVOLUCION, ECOLOGÍA Y OPTIMIZACION DE LOS ORGANISMOS DIGITALES.
Thomas S. Ray

ROBOTS .

6. EL DESARROLLO DE LA MEMORIA A CORTO PLAZO EN LOS ROBOTS BASADOS EN LA CONDUCTA.
Peter Kronberg
7. ¿ PUEDE UNA COMPUTADORA SENTIR DOLOR ?
David W. Croft

MODELOS COMPUTACIONALES DE ORGANISMOS OPERANTES .

8. APRENDIZAJE POR REFORZAMIENTO COMPUTACIONAL.
Richard S. Sutton
9. SUBIGUALAMIENTO EN UN ORGANISMO SIMULADO POR COMPUTADORA.
Matt J. Morris
10. SE NECESITAN MODELOS CONDUCTUALES COMPLETOS PARA EL PROGRESO DE LAS CIENCIAS DE LA CONDUCTA.
William R. Hutchison

BREVE HISTORIA DE LA VIDA DIGITAL

Claus O. Wilke y Christoph Adami
California Institute of Technology

A mediados de los 1980s, un número creciente de investigadores se vieron fascinados con la idea de la existencia de programas de computadora auto replicables. Cada vez era mas frecuente encontrar reportes sobre virus de computadoras (programas que podían propagarse autónomamente de una computadora a otra) y que, ciertamente, inspiraron la investigación sobre la ecología de los programas de computadora que se reproducían en ambientes controlados. Las primeras de estas investigaciones tenían la forma de juegos. En el "Core War" (1984), los participantes humanos competían entre sí escribiendo programas en un ambiente espacialmente limitado (la memoria de la computadora). Una estrategia ganadora era la de escribir programas auto replicables, que finalmente llenaban el espacio disponible con copias de ellos mismos, desplazando así a los programas competidores. Las mutaciones no ocurrían y consecuentemente, los programas no evolucionaban. Otras de las primeras investigaciones se interesaban en preguntas sobre el origen de la auto replicación, mas que con la evolución y la ecología. Rasmussen y colaboradores (1990) estudiaron la emergencia de los programas auto replicables en un ambiente ruidoso, pero no observaron la evolución de programas complejos. Tom Ray (1991), (un entrenado ecólogo de plantas tropicales), fue el primero en lograr la creación de una verdadera evolución darwineana de los programas de computadora auto replicables. En su mundo "TIERRA", los programas auto replicables tenían que enfrentarse a variaciones aleatorias en su código, mismos que conducían a una rápida diversificación de la población de programas y eventualmente, a un incremento significativo de su adaptación (fitness). Después de varias generaciones, los programas en la población se multiplicaban con mayor rapidez de lo que lo hacían sus ancestros escritos a mano, que habían iniciado la población. No obstante, esta ganancia adaptativa generalmente era obtenida al acortar el tamaño del programa. La evolución de programas complejos a partir de replicadores simples, se observó por primera vez en el mundo de "AVIDA"

(1998), mediante la evolución de vías y reacciones computacionales. Finalmente el mundo de "AMOEBA" (1996), se desarrolló para estudiar la emergencia de los programas auto replicables, a partir de los no replicadores.

EL METABOLISMO COMPUTACIONAL.

Claus O. Wilke y Christoph Adami
California Institute of Technology

Los organismos bioquímicos frecuentemente obtienen ventajas adaptativas debido a sus metabolismos particulares. Una situación análoga existe en el caso de los organismos digitales, los que ejecutan sus programas a velocidades variables, determinando así su tasa reproductiva. Entre mayor es la velocidad de ejecución del programa, mas rápido es como se reproduce el organismo digital. A su vez, la velocidad de ejecución, está determinada por el metabolismo computacional del organismo.

El metabolismo computacional es el conjunto de todas las reacciones computacionales que ejecuta un organismo. Las reacciones computacionales ocurren como sigue: los organismos digitales pueden obtener números de su ambiente (estos números pueden compararse con las moléculas químicas que están presentes en el ambiente de los organismos). Con el código genético adecuado (equivalente a la secuencia de código de una enzima que cataliza una reacción particular), los organismos pueden llevar a cabo operaciones con estos números. Las operaciones recompensadas son operadores de la lógica simbólica como la conjunción AND. Los resultados se depositan nuevamente en el ambiente. Si la operación se juzga que será benéfica en el ambiente dado, el organismo experimenta un incremento en la velocidad de ejecución del programa, como resultado. Debido a que diferentes reacciones computacionales pueden conjuntarse para producir aún mas beneficios computacionales, el conjunto de genes computacionales puede considerarse como un sendero computacional.

Este metabolismo computacional es clave para la evolución de organismos complejos. En ausencia de senderos computacionales recompensados, la única forma en que un organismo puede incrementar su adaptación es recortando su genoma lo mas posible, para así reducir el tiempo que le toma copiarlo para reproducirse. Así, esta dinámica es muy

parecida a los experimentos de transferencia serial de Spiegelman. Sin embargo, cuando se posibilitan los senderos computacionales, el organismo experimenta un conflicto entre las instrucciones adicionales y el tiempo que toma ejecutar los cálculos, que reducen la tasa de replicación, y el incremento en la velocidad de ejecución del programa resultante de completar exitosamente los cálculos. En general, cuando los senderos computacionales se recompensan con una aceleración suficiente, el esfuerzo de hacer los cálculos vale la pena. En tal caso, uno puede observar, al paso de varios cientos de generaciones, un tremendo incremento en la complejidad de estos organismos, produciéndose eventualmente organismos que pueden efectuar arriba de 50 o más operaciones distintas.

GLOSARIO.

Metabolismo Computacional: el total de reacciones computacionales que un organismo puede realizar.

Reacción Computacional: una operación (lógica matemática) ejecutada por un organismo con los números provistos por el ambiente. Para lograr tal reacción, el organismo debe tener un gene computacional (código) o un sendero para esa reacción.

Mutación Explícita: cambio en el genoma entre padres e hijos, que es causado por el ruido en el ambiente (mutación de copia, mutación de inserción, mutación de borrado, etc).

Genoma: el programa que el organismo ejecuta (compuesto por instrucciones dispuestas en forma circular).

Instrucción: la unidad básica de información en el genoma de un organismo digital. Cada instrucción se corresponde típicamente con una única acción ejecutada por la unidad de procesamiento central.

Mutación Implícita: cambio en el genoma entre padres e hijos que es causado por un mal funcionamiento en el proceso de copiado del organismo procreador.

LA BIOLOGÍA DE LOS ORGANISMOS DIGITALES.

Claus O. Wilke y Christoph Adami
California Institute of Technology

1. Los organismos digitales son programas de computadora auto replicables, que viven en un ambiente controlado.
2. La selección ocurre debido a que el ambiente donde viven los organismos digitales es un espacio limitado, esto es, con el nacimiento de un nuevo organismo, se retira de la población uno viejo (típicamente elegido aleatoriamente). Luego entonces, aquellos organismos que producen mas crías, remplazan, a lo largo del tiempo, a otros menos eficientes en replicarse.
3. En los organismos digitales ocurren las mutaciones por la vía de los errores genómicos explícitos (como mutaciones puntuales que se dan durante el proceso de copiado), o como mutaciones implícitas que son el resultado de algoritmos de copia defectuosos. Por ejemplo, un organismo puede perder parte de su genoma durante la replicación o replicar parte de su genoma dos veces. Las mutaciones por copia se dan porque el copiado de una sola instrucción en el genoma tiene una cierta probabilidad de fallar, que ocasiona que otra instrucción se escriba en el genoma de la hija. Otras mutaciones explícitas son cambios aleatorios en el genoma del organismo que ocurren independientemente del proceso de copia (mutaciones de rayos cósmicos), o inserciones y/o borrado aleatorio de instrucciones individuales. La tasa de las mutaciones explícitas está bajo el control del investigador, en tanto que las mutaciones implícitas típicamente no pueden ser controladas.
4. ¿Qué clase de preguntas pueden contestarse con los organismos digitales? En este documento se subrayan tres: la dinámica de las adaptaciones a largo plazo, la distribución de las interacciones epistáticas entre las mutaciones y la dinámica quasi-especie.

5. Adaptación a largo plazo .- Una piedra angular de la biología evolutiva es la que se refiere a la influencia de la mutación y la selección sobre los organismos en periodos de tiempo largos (del orden de miles de generaciones o mas), debido a que la teoría de Darwin predice la macroevolución y la emergencia de novedad en esa escala de tiempo. Para los organismos digitales, este tipo de propagación puede lograrse en cuestión de días y por ello no es de sorprender que los experimentos con organismos digitales, tradicionalmente se enfoquen en las rutas evolutivas de largo plazo. En las primeras observaciones realizadas con estos experimentos, se vio que la evolución ocurre de una forma intermitente. Periodos breves de rápido incremento en la adaptación se intercalan con periodos largos, durante los cuales la adaptación promedio de la población permanece constante.
6. Otros estudios de la evolución a largo plazo, empleando organismos digitales, sustentan la idea del aumento de complejidad en los sistemas evolutivos, la diferenciación del genoma o la influencia del azar y la historia sobre la adaptación.
7. Interacciones epistáticas .- El entendimiento de las interacciones epistáticas entre mutaciones es clave para muchas cuestiones importantes en la biología evolutiva. Por ejemplo, la hipótesis mutacional determinista del origen del sexo, necesita que el producto de diversas mutaciones deletéreas se refuerce (sinérgicamente) en sus efectos, en contraste a un efecto mitigante (antagonista). Los datos de las investigaciones con organismos digitales, en este caso, no respaldan esta teoría.
8. Sin embargo, otras investigaciones sobre el sinergismo y el antagonismo de las mutaciones, muestran resultados muy similares a los obtenidos con bacterias como la E. coli, lo que sustenta la hipótesis de que en muchos aspectos, los sistemas evolutivos se gobiernan por principios generales.
9. Dinámica quasi-especie .- En pocas palabras, el concepto de quasi-especie plantea que los organismos asexuales evolucionan como un grupo cohesivo de mutantes cercanamente relacionados y que la selección actúa en

estas nubes de mutantes (quasi-especies) y no en los organismos individuales.

10. Una de las predicciones más espectaculares de la teoría de quasi-especies es la de que un replicador rápido estará menos adaptado que un replicador lento, que tenga un mejor soporte mutacional. Recientemente ha sido posible demostrar este efecto con organismos digitales.

DIRECCIONES DE LA INVESTIGACIÓN FUTURA.

11. Genética De la vida digital .- A primera vista, el genoma de un organismo digital evolutivo aparece como una colección aleatoria de instrucciones para la computadora, construido sin ningún plan u organización. Sin embargo, una inspección detallada revela que estos genomas están sorprendentemente bien organizados y que frecuentemente se pueden subdividir en distintos bloques funcionales, que merecen el nombre de genes. Estos genes pueden descubrirse como sigue: uno reemplaza sistemáticamente cada instrucción del genoma, una a la vez, con una instrucción nula especial que no tiene función alguna. Entonces, uno evalúa a cada uno de los organismos obtenidos de esta manera en su habilidad para replicarse, en su velocidad de replicación y en las reacciones computacionales que son capaces de realizar. De esta manera, uno obtiene un mapa de las partes del genoma que son esenciales para la replicación, de las partes que juegan un papel en ciertas reacciones computacionales, pero que no son vitales y las partes que no tienen una función discernible (genes basura). Este método de mapeo de los genomas, luego puede aplicarse a las mutaciones cercanamente relacionadas, por ejemplo a una secuencia de descendientes sucesivos, tomada de una población en evolución. Los mecanismos que subyacen y dirigen la evolución serán revelados en detalle en tal tipo de investigación.

12. Ecología evolucionaria .- Una dirección obvia para el trabajo futuro está en la creación de interacciones

ecológicas emparejando la eficiencia del metabolismo computacional de los organismos con la presencia o ausencia de recursos externos.

13. Conclusión.- Recientemente Lipson y Pollack mostraron que los principios de los robots de auto replicación simple están creciendo en la tecnología actual. Eventualmente, tales robots y el software que los dirige pueden evolucionar sin la interacción humana, hasta el punto que se vuelvan parte del ecosistema en el que vivimos.

ESTUDIANDO LA EVOLUCION CON ORGANISMOS DIGITALES.

Henry Bortman
Astrobiology Magazine

1. Todos los organismos biológicos compiten por recursos limitados. Los que son exitosos se reproducen y sus especies sobreviven otra generación. Y cuando las especies están en competencia directa, la que gana es la que se reproduce mas rápido (de acuerdo a las reglas tradicionales de la genética poblacional).
2. No obstante, investigaciones recientes indican que la reproducción rápida no siempre es el boleto para un triunfo evolutivo. Cuando las tasas de mutaciones son bajas, nos dice Chris Adami del Instituto de Tecnología de California, la regla estándar es aplicable. Pero cuando las tasas de mutaciones son altas, se aplica un principio diferente, que Adami y colaboradores denominan "sobrevivencia aplanada". Sin embargo, Adami no llegó a esta conclusión estudiando organismos biológicos. Los organismos que él y sus colegas estudiaron eran digitales.
3. Estos organismos existen como secuencias de código de computadoras auto replicables. El código determina como se reproduce el organismo y como utiliza los recursos disponibles: ciclos de memoria y procesamiento. Para sobrevivir, los organismos tienen que encontrar la manera de incrementarse o al menos de mantenerse, para lo que comparten el control de los recursos. Interconstruida en sus códigos está la habilidad para cambiar o mutar. El código, nos dice Adami, es su genoma. Adami considera que se pueden aprender valiosas lecciones sobre la evolución estudiando como los organismos digitales se adaptan y sobreviven.
4. Los investigadores han encontrado que los organismos expuestos a tasas altas de mutaciones, en el tiempo, se "robustecen", desarrollan una tolerancia a la mutación. Y ellos sospechan que en un ambiente de alta mutación, estos organismos robustos arrollarán a otros

- competidores mas prolíficos, pero menos robustos. Los datos así lo indican (Adami, Wilke y Wang, 2001; Lenski y Ofria, 2001).
5. Estos científicos suelen representar la competencia entre especies, en el seno de un ecosistema, con una gráfica que muestra un piso con picos de diferente altura. Cada especie se muestra como habitando un "pico adaptativo" que sobresale de la línea base, semejando una montaña que se levanta sobre la planicie. Las especies bien adaptadas para sobrevivir en condiciones de tasas bajas de mutación (que es el estado típico que se encuentra en la naturaleza), aparecen como picos altos y agudos. Pero para estas especies, los cambios pueden ser desastrosos. Si la tasa de mutaciones repentinamente se incrementa debido a alguna presión ambiental, muchos organismos resultan mal adaptados y fracasan en reproducirse.
 6. Puntualicemos también que las especies expuestas a una tasa de mutación alta (asumiendo que sobrevivan), con el tiempo responden desarrollando resistencia hacia las mutaciones dañinas. Mientras cualquier organismo individual quizá no sobreviva a cualquier mutación particular, la "nube" de organismos que conforma la población, puede contener algunos organismos que puedan sobrevivir. Adami se refiere a esta nube como "quasi-especie", debido a que ningún organismo contiene el genoma de la especie. En lugar de ello, existe un rango de genomas y es esta variedad lo que permite que la quasi-especie sobreviva, de frente a las mutaciones dañinas. En la gráfica de adaptación, tales especies están representadas con picos bajo y aplanados. De ahí viene el concepto de la sobrevivencia aplanada.
 7. La información que es la secuencia genética de estos organismos digitales es codificada de manera física, en diferencias de voltaje, en la memoria de la computadora. Y, como tal, es tan físico como la información codificada en una secuencia de ácido nucleico.
 8. Por ejemplo, los virus, exhiben un comportamiento quasi-especie superior al de los mas robustos de estos organismos digitales, nos dice Adami. En la evolución de los virus, claramente tenemos tasas de mutación del orden que hemos referido (altas). Es claro que los virus

no tienen una secuencia particular. Los virus no son una especie pura. De hecho, son, una nube, una nube mutante, que vive en picos aplanados. En ellos se presentan muchísimos genotipos.

9. Entender estas formas diferentes de vida puede ayudar a reconocer y entender la posibilidad de vida extra terrestre, en mundos con diferente fundamento químico, en comparación con la vida en la Tierra.

EVOLUCION, ECOLOGÍA Y OPTIMIZACION DE LOS ORGANISMOS DIGITALES.

Thomas S. Ray

School of Life & Health Sciences, Univ. of Delaware

1. **INTRODUCCION** .- Idealmente, la ciencia de la biología debería abarcar a todas las formas de vida. Sin embargo, en la práctica, se ha restringido al estudio de una sola forma de vida, la vida en la tierra. La vida en la tierra es muy diversa, pero se puede decir que toda ella es parte de una sola filogenia. Como la biología se basa en una muestra de un solo sujeto, no podemos saber qué características de la vida son peculiares a la tierra y cuales son características generales, de toda forma de vida.
2. Una alternativa práctica ante una biología interplanetaria o mítica es la de crear vida sintética en una computadora. El objetivo no necesariamente sería crear formas de vida que sirvan como modelo para el estudio de la vida natural, sino crear formas de vida radicalmente diferentes, basadas en una física y una química completamente diferente y dejar que estas formas de vida evolucionen en su propia filogenia, llegando a las formas que sean naturales a sus bases físicas únicas. Estas instancias de vida verdaderamente independientes, podrían servir como base de comparación, para saber algo de lo que es general y lo que es peculiar en biología. Aquellos aspectos de la vida que prueben ser lo suficientemente generales para ocurrir en ambos sistemas natural y sintético, podrán ser estudiados con mayor facilidad en los sistemas sintéticos.
3. La vida debe definirse de forma tal que no se restrinja a las formas basadas en el carbón. Yo considero que un sistema está vivo si se auto replica y es capaz de una evolución con un final abierto. La vida sintética deberá auto replicarse y evolucionar sus estructuras y procesos, sin preconcepción ni previo diseño de su creador. >Los programas Core Wars, los virus de computadora y los gusanos pueden auto replicarse, pero

- afortunadamente, no evolucionan. Es improbable que tales programas puedan alguna vez estar completamente vivos, ya que no es probable que lleguen a evolucionar. Muchas simulaciones evolutivas no tienen un final abierto.
4. La auto replicación es crítica para la vida sintética porque sin ella los mecanismos de selección también estarían predeterminados por el simulador.
 5. Generalmente se acepta que el origen de la vida es un fenómeno de primer orden. Hay otro fenómeno en la historia de la vida, que es menos conocido, pero tiene una importancia comparable y es el origen de la diversidad biológica y de la vida multicelular macroscópica durante la explosión Cámbrica, hace 600 millones de años. Este evento involucra la amplia diversificación de las formas de vida. Repentinamente aparecieron docenas de filias. El trabajo al que se refiere este documento, busca un paralelismo con este segundo fenómeno en la historia de la vida, el origen de la diversidad.
 6. Esta aproximación ha producido comunidades rápidamente diversificadas de organismos auto replicables que exhiben evolución con un final abierto mediante la selección natural. A partir de un solo ancestro rudimentario, conteniendo solo el código de auto replicación, han emergido espontáneamente interacciones como parasitismo, inmunidad, hiper-parasitismo, socialización y engaño.
 7. **MÉTODOS** : La Metáfora .- La vida orgánica se ve como el empleo de energía, en su mayor parte derivada del sol, para organizar la materia. Por analogía, la vida digital puede ser vista como el uso del tiempo del CPU (unidad de procesamiento central de una computadora), para organizar la memoria. La vida orgánica evoluciona mediante la selección natural de los individuos que compiten por los recursos (luz, alimento, espacio, etc), de manera que los genotipos que dejan mas descendientes, se incrementan en frecuencia. La vida digital evoluciona mediante el mismo proceso, conforme los algoritmos replicantes compiten por el tiempo del CPU y el espacio de la memoria, los organismos desarrollan estrategias para aprovecharse de los otros. El tiempo del CPU es la

- analogía de los recursos energéticos y la memoria es la analogía de los recursos espaciales.
8. La memoria, el CPU y el sistema operativo de la computadora son elementos de este ambiente (físico) "abiótico". Así, se diseña una "criatura" para estar específicamente adaptada a las características de este ambiente computacional. La criatura está formada por un programa ensamblador auto replicable. Los lenguajes ensamblador son memorandos para los códigos máquina y son ejecutados directamente por el CPU. Estos códigos máquina tienen la característica de invocar directamente el grupo de instrucciones del CPU y los servicios proporcionados por el sistema operativo.
 9. Las instrucciones de la máquina proporcionan las bases más naturales de una química artificial para criaturas diseñadas para vivir en una computadora. En la analogía biológica, las instrucciones de la máquina se considera que son más parecidas a los aminoácidos que a los ácidos nucleicos, porque estos son "químicamente activos".
 10. En la computadora, un bloque de memoria RAM (memoria de acceso al azar), se utiliza como una "sopa" que puede ser inoculada con criaturas. El "genoma" de estas criaturas se forma con secuencias de instrucciones máquina que hacen de la criatura un algoritmo auto replicable. El prototipo de esta criatura consta de 80 instrucciones máquina, así que el tamaño del genoma de la criatura es de 80 instrucciones y su "genotipo" está en la secuencia específica de esas 80 instrucciones.
 11. La Computadora Virtual (simulador TIERRA) .- La computadora que usamos es una computadora de propósito general, que entre otras cosas, es capaz de emular mediante el software, la conducta de cualquier otra computadora que se haya construido o que se pueda construir. El trabajo que se describe en este documento tiene lugar en una computadora virtual conocida como TIERRA (nombre de nuestro planeta, en español). Tierra es una computadora paralela del tipo MIMD (instrucciones múltiples, múltiples datos), con un procesador (CPU) para cada criatura. Cada CPU virtual es implementado mediante programación en lenguaje C.
 12. El grupo de instrucciones de un CPU típicamente ejecuta operaciones aritméticas simples o manipulaciones de

bits. Algunas instrucciones mueven los datos entre los registradores en el CPU o entre estos y la memoria RAM. Otras instrucciones controlan la ubicación y movimiento de un "marcador de instrucción" (IP). El IP indica la dirección en RAM donde la máquina codifica el programa ejecutivo (es decir, el organismo digital).

13. El CPU ejecuta perpetuamente ciclos (ir a-decodificar-ejecutar-incrementar_IP). En la máquina, la instrucción en código vigente en el IP, es llevada al CPU, su patrón de bits es decodificado para determinar qué instrucción es la que le corresponde y esta instrucción es ejecutada. En seguida, el IP avanza un renglón secuencialmente a la siguiente posición en la RAM, de donde la siguiente instrucción es conducida. No obstante, algunas instrucciones como JMP, CALL y RET manipulan directamente al IP, haciendo que la ejecución brinque a alguna otra secuencia de instrucciones en la RAM.
14. El Lenguaje Terráqueo .- Al desarrollar este nuevo lenguaje virtual, denominado "Terráqueo", se prestó mucha atención a las propiedades estructurales y funcionales del sistema informativo de las moléculas, en los sistemas biológicos: DNA, RNA y proteínas. Se tomaron prestadas dos características del mundo biológico, que se consideraron críticas para hacer evolutivo al lenguaje terráqueo. Primero, el conjunto de instrucciones del lenguaje terráqueo fue definido para ser de un tamaño del mismo orden que la magnitud del código genético. La información se codifica en el DNA mediante 64 codons, que se traducen en 20 aminoácidos. Así, en su forma actual, el lenguaje terráqueo consiste de 32 instrucciones, que pueden representarse con 5 bits. Una segunda característica es lo que se denomina como modo de localización, que es del tipo "ubicación por diseño". En muchos códigos máquina, cuando se ubica un los datos o cuando el IP brinca a otra pieza de código, se especifica en el código de la máquina la dirección numérica exacta. En contraste, considérese que en los sistemas biológicos, para que la molécula de proteína A, en el citoplasma de una célula, interactúe con la molécula de proteína B, no se especifican las coordenadas exactas donde se localiza B.

En lugar de ello, la molécula A presenta un diseño en su superficie que es complementario a cierta superficie en B. La difusión hace que estos se junten y su complementariedad confirmada permite que interactúen.

15. La ubicación por diseño se ilustra en el lenguaje terráqueo con la instrucción JMP (brincar). Cada instrucción JMP es seguida por una secuencia de instrucciones NOP (no-operación), de las cuales hay dos tipos: NOP_0 y NOP_1. Supongamos que tenemos una parte de código con cinco instrucciones en el orden siguiente: JMP NOP_0 NOP_0 NOP_0 NOP_1. El sistema buscaría a lo largo de las dos direcciones de la instrucción JMP, para localizar la ocurrencia más próxima del patrón complementario: NOP_1 NOP_1 NOP_1 NOP_0. Si encuentra este patrón, el puntero de instrucción se moverá al final del patrón complementario continuando con la ejecución del programa. Si no encuentra el patrón complementario, se pasará a una condición de error (flag) y se ignorará la instrucción JMP. El lenguaje terráqueo se distingue por dos características: un grupo de instrucciones verdaderamente pequeño, sin operandos numéricos y un direccionamiento o ubicación por diseño.
16. El Sistema Operativo Terráqueo .- La computadora virtual terráquea necesita de un sistema operativo virtual que hospede a los organismos digitales. El sistema operativo determinará los mecanismos de comunicación interproceso, la localización en la memoria y la ubicación del tiempo del CPU entre los procesos competitivos. Los algoritmos evolucionarán, de manera que exploten estas características para su provecho. Mas que ser simples aspectos del ambiente, el sistema operativo junto con el grupo de instrucciones determinará la topología de las posibles interacciones entre individuos, como la habilidad de pares de sujetos para exhibir relaciones predador-presa, parásito-huésped o mutualismo.
17. (A) Ubicación en la Memoria .- La computadora terráquea funciona en una porción de RAM de la computadora verdadera, que se aparta con este propósito. Esta porción de RAM es conocida como la "sopa" y está formada de 60,000 bytes, que pueden sostener a un número igual de instrucciones máquina terráqueos. Cada

"criatura" ocupa una parte de esta porción de memoria en la sopa.

18. Se considera que las criaturas terráqueas son de tipo celular, en el sentido de que están protegidas por una "membrana semi-permeable" de ubicación en la memoria. El sistema operativo terráqueo proporciona servicios de ubicación en la memoria. Cada criatura tiene privilegios exclusivos de escritura, en su propia ubicación en la memoria. El "tamaño" de la criatura es exactamente el tamaño de su porción ubicada (ejem.: 80 instrucciones). Esto generalmente corresponde al tamaño del genoma. Esta "membrana" se dice semi-permeable debido a que, mientras los privilegios de escritura están protegidos, no lo están los privilegios de lectura y ejecución. Una criatura puede examinar el código de otra e incluso ejecutar su programa, pero no puede escribir sobre él. Cuando las criaturas terráqueas se "dividen", la célula madre pierde los privilegios de escribir en el espacio de la célula hija.
19. (B) Tiempo Compartido (La Rebanadora) .- El sistema operativo terráqueo debe ser multi-tarea, para que una comunidad de criaturas individuales puedan vivir simultáneamente en la sopa. El sistema fragmenta pequeñas rebanadas de tiempo de CPU para cada criatura de la sopa en turno.
20. (C) Mortalidad (El Cosechador) .- Las criaturas auto replicables dentro de una sopa de tamaño fijo, rápidamente llenarán la sopa, bloqueando el sistema. Para prevenir que no ocurra esto, es necesario incluir la mortalidad. El sistema operativo terráqueo incluye un "cosechador", que va matando a las criaturas de una fila, cuando la memoria se llena en un nivel específico (ejem.: 80%). La muerte ocurre cuando se desubica su memoria y se retiran de la fila del cosechador y de la rebanadora. Su código "muerto" no es retirado de la sopa. En nuestro sistema, el cosechador emplea una fila lineal. Cuando una criatura nace entra al fondo de la fila. El cosechador siempre mata a la criatura que está hasta delante de la lista, Sin embargo, los individuos pueden moverse hacia delante y hacia atrás en la fila, dependiendo de su éxito o fracaso al ejecutar ciertas instrucciones. Cuando la criatura ejecuta una

instrucción que genera una condición de error, avanza hacia delante una posición en la fila. El efecto de la lista del cosechador es el de causar que los algoritmos fracasados se vayan al frente y mueran. Los algoritmos vigorosos tienen una mayor longevidad, pero en general, la probabilidad de morir se incrementa con la edad.

21. (D) Mutación.- Para que ocurra la evolución, debe haber algunos cambios en el genoma de las criaturas. Esto puede ocurrir durante el tiempo de vida de un individuo o pueden darse errores al elaborarse el genoma de las crías. Con el propósito de asegurarse que haya estos cambios genéticos, el sistema operativo en forma azarosa lanza pedazos en la sopa y las instrucciones del lenguaje terráqueo se ejecutan imperfectamente.
22. Las mutaciones se dan en dos circunstancias. Con una tasa subyacente, los pedazos (bits de información) se seleccionan aleatoriamente de la totalidad de la sopa (ejem.: 60,000 instrucciones hacen un total de 300,000 pedazos) y son lanzados. Esto es una analogía de las mutaciones causadas por los rayos cósmicos y tiene el efecto de prevenir que alguna criatura sea inmortal, haciendo que eventualmente la mutación lleve a la muerte. La tasa de mutación subyacente generalmente se calibra a razón de un pedazo lanzado por cada 10,000 instrucciones terráneas ejecutadas por el sistema.
23. Adicionalmente, al copiar las instrucciones durante la replicación de las criaturas, los pedazos son lanzados al azar en las copias, con cierta tasa. La tasa de mutación por copiado es la más alta de las dos y produce errores en la replicación. La tasa de mutación por copia generalmente se establece a razón de un pedazo lanzado por cada 1,000 a 2,500 instrucciones. En los dos tipos de mutaciones, el intervalo entre mutaciones varía al azar, dentro de un rango, para evitar posibles efectos periódicos.
24. Las mutaciones producen la aparición de nuevos genotipos, mismos que son vigilados por un administrador del banco genético. Una función de este administrador, cuando un nuevo genotipo se replica dos veces, produciendo una cría genéticamente idéntica al menos una vez, es la de otorgarle un nombre único y guardarlo en el disco. Cada nombre de genotipo contiene un código con

dos partes, un número y tres letras. El número representa la cantidad de instrucciones en el genoma. El código de tres letras es usado como base de un sistema de 26 numerales, para asignar una etiqueta única a cada genotipo de un tamaño clasificado. Al primer genotipo en aparecer, de un tamaño determinado, se le asigna la etiqueta aaa, al segundo se le asigna la etiqueta aab y así subsecuentemente. Así, el ancestro es llamado 80aaa y el primer mutante, de tamaño 80, es llamado 80aab. La primera criatura de tamaño 45 se llamaría 45aaa.

25. El Ancestro Terráqueo .- Yo use el lenguaje terráqueo para escribir un solo programa auto replicable con 80 instrucciones de largo. Este programa es referido como el "ancestro" o alternativamente como genotipo 0080aaa. El ancestro es un algoritmo mínimo auto replicable, que originalmente se escribió para usarse durante la depuración del simulador. Ninguna funcionalidad se diseñó para el ancestro, mas allá de la habilidad de auto replicarse. No había ningún diseño de potencial evolutivo específico.
26. El ancestro se examina a sí mismo para determinar donde empieza y termina en la memoria. El inicio del ancestro está marcado con el diseño de cuatro no-operaciones: 1 1 1 1, y su final está marcado con 1 1 1 0. Habiendo determinado su ubicación, resta estos dos valores para calcular su tamaño y localiza un trozo de memoria de ese tamaño para la célula hija. Entonces, activa el procedimiento de copiado del genoma entero a la memoria de la célula hija, una instrucción cada vez. El inicio del procedimiento de copiado se marca con un diseño de cuatro no-operaciones: 1 1 0 0. Cuando el genoma se ha copiado, se ejecuta la instrucción DIVIDE, que hace que la criatura pierda sus privilegios de escritura en la memoria de la célula hija y le proporciona a esta un puntero de instrucción (también mete a la célula hija en la fila de la rebanadora y el cosechador). Luego de su primera replicación, la célula madre no se examina de nuevo, procede directamente a la localización de otra célula hija, luego el procedimiento de copia es seguido por la división celular, en un rizo que se perpetúa.

27. **RESULTADOS** : Evolución .- Las corridas evolutivas en el simulador se inician inoculando en la sopa cerca de 60,000 instrucciones con un solo organismo, con un genotipo ancestral de 80 instrucciones. El paso del tiempo se mide en términos de qué tantas instrucciones terráqueas han sido ejecutadas por el simulador. La célula ancestral original ejecuta 839 instrucciones en su primera replicación y 813 en cada replicación adicional. La célula inicial y sus hijas replicables llenan rápidamente la sopa de memoria al nivel del umbral de 80%, lo que inicia al cosechador. Típicamente, el sistema ejecuta cerca de 400,000 instrucciones al llenar la sopa con cerca de 375 individuos de talla 80 (y sus células hijas gestantes). Una vez que el cosechador se inicia, la memoria se mantiene próxima a 80% llena con criaturas.
28. (A) Micro-Evolución .- Si no hubiera mutaciones, no habría evolución. Sin embargo, los trozos (de información) lanzados como resultado de los errores de copiado o las mutaciones de fondo (subyacentes), producen criaturas cuya lista de 80 instrucciones (genotipo) difiere de la del ancestro, usualmente por un solo trozo de información en una sola instrucción. Las mutaciones no pueden producir cambios en el tamaño de la criatura, solo alteran las instrucciones en su genoma. No obstante, al alterar el genotipo, las mutaciones pueden afectar el proceso mediante el cual la criatura se auto examina y calcula su tamaño, potencialmente causando que produzca una hija de diferente tamaño.
29. parásitos Un ejemplo del tipo de error que puede producir una mutación sobre un diseño, es la que ocurre en la instrucción 42 del ancestro. Se trata de una del tipo NOP_0, el tercer componente del diseño para el procedimiento de copia. La mutación la convierte en NOP_1, cambiando el diseño de 1 1 0 0 a 1 1 1 0. Esta nueva configuración será reconocida como el diseño que sirve para marcar el final (del tamaño) de la criatura y no el procedimiento de copiado.
30. Una criatura nacida con una mutación en la instrucción 42 calculará su tamaño como de 45 y ubicará su célula hija en un tamaño de 45, copiando las instrucciones de

su genoma de la 0 a la 44. Entonces la célula hija, no incluirá el proceso de copiado. El genotipo de la hija, que consta de 45 instrucciones, se llamará 0045aaa. Este genotipo no puede auto replicarse en una cultura de aislamiento.

Pero, recordemos que las criaturas pueden igualar diseños con el código localizado en la memoria de otras criaturas, incluso pueden ejecutar ese código. Así, la criatura 0045aaa ubicada en una cultura mixta con otras 0080aaa, cuando quiera operar el procedimiento de copiado y no encuentre el diseño en su propio genoma, lo hará basándose en el genotipo de otra criatura del tipo 0080aaa, estableciendo una relación parasitaria.

31. inmunidad a los parásitos Al menos algunos genotipos de tamaño 79 han demostrado cierta resistencia a los parásitos. Si un genotipo 45aaa se introduce en la sopa, flanqueado a cada lado por con individuos del genotipo 0079aab, los 0045aaa inicialmente se reproducen un poco, pero rápidamente son eliminados de la sopa.
32. evitando la inmunidad a los parásitos Ocasionalmente estos sistemas evolutivos dominados por el tamaño 79 fueron invadidos exitosamente por parásitos tamaño 51. Cuando el genotipo inmune 0079aab fue probado con un 0051aao (un descendiente directo, de un paso, de un 0045aaa en el que la instrucción 39 se reemplazó con la inserción de siete instrucciones de origen desconocido), se vio que entraban en un ciclo de estabilidad. Evidentemente el 0051aao había evolucionado de alguna manera para evitar la inmunidad a los parásitos procesados por 0079aab. Los catorce genotipos 0051aaa hasta el 0051aan, también fueron evaluados con un 0079aab y ninguno pudo invadir.
33. hiper-parásitos Se han descubierto los hiper-parásitos (ejem.: 0080gai, que difiere por 19 instrucciones, de su ancestro). Su habilidad para trastornar el metabolismo energético de los parásitos se basa en dos cambios. El procedimiento de copia no regresa, pero salta hacia atrás, directamente a la dirección apropiada del rizo reproductivo. De esta manera dimensiona con efectividad al puntero de instrucción desde el parásito. Además, luego de cada reproducción, el hiper-parásito se re-examina, actualizando su localización y su tamaño. Luego

- que el puntero de instrucción del parásito pasa por este código, el CPU del parásito contiene la localización y el tamaño del hiper-parásito y el parásito, consecuentemente, replica el genoma del hiper-parásito.
34. hiper-parásitos sociales Los hiper-parásitos llevan a los parásitos a la extinción. Esto trae como resultado, una comunidad con un relativo alto nivel de uniformidad genética y una alta relación genética entre los individuos de la comunidad. Estas son las condiciones que fundamentan la evolución social y los hiper-parásitos sociales rápidamente dominan la comunidad. Los hiper-parásitos sociales se ven con un tamaño de clase de 61 instrucciones. Por ejemplo, el 0061acg es social en el sentido de que solo se auto replica cuando existe en agregados. Los vecinos cooperan compartiendo el puntero de instrucciones. También, pareciera que la presión selectiva para la evolución de la socialidad es de tal forma, que facilita la reducción de tamaño. Las especies sociales son 24% menores que el ancestro. Esto lo logran acortando sus diseños de cuatro instrucciones a solo tres.
 35. los tramposos: hiper-hiper-parásitos El sistema social cooperativo de los hiper-parásitos puede ser trampeado e invadido por hiper-hiper-parásitos. Estos tramposos (ejem.: 0027aab) se posicionan entre los hiper-parásitos gregarios, de manera que, cuando el puntero de instrucciones pasa entre ellos, estos lo capturan.
 36. auto-examen novedoso Todas las criaturas de las que hemos hablado aquí marcan su inicio y final mediante diseños, localizan la ubicación de los dos diseños correspondientes y determinan el tamaño de su genoma restando un valor del otro. No obstante, algunas criaturas evolucionan sin un diseño que marque su final. Estas criaturas localizan la ubicación del diseño que marca su inicio y luego, la ubicación del diseño a la mitad del genoma. Estas dos direcciones se restan para calcular la mitad del tamaño y este valor es multiplicado por dos, para calcular el tamaño total.
 37. una adaptación intrincada Las razas ramificadas descritas en los párrafos anteriores evolucionaron en el periodo correspondiente a la ejecución de un billón de instrucciones por el sistema. Cuando se dejó correr el

- sistema por quince billones mas y se examinó a una de las criaturas que habían evolucionado, se le notó una adaptación intrincada consistente en una técnica de optimización conocida como "desenrollar el rizo".
38. (B) Macro-Evolución Cuando se pone a trabajar el simulador durante periodos largos de tiempo, cientos de millones o billones de instrucciones, emergen varios patrones. Bajo la selección de tallas pequeñas, hay una proliferación de pequeños parásitos, que generan un ambiente ecológico interesante. Cuando la selección es sobre criaturas largas, se observa un incremento continuo de talla, evolucionando genomas tan largos como de 23,000 instrucciones de tamaño.
39. Diversidad .- Muchas de las observaciones de diversidad en las criaturas de Tierra, se basan en la diversidad de clases de tamaños. Es claro que las criaturas de diferente tamaño son genéticamente diferentes, pues sus genomas son de diferente tamaño. Las criaturas de diferente tamaño tendrían alguna dificultad en recombinarse si fueran sexuales, así que es probable que serían de diferentes especies. En una corrida de 526 millones de instrucciones, se generaron 366 clases de diferente tamaño, 93 de las cuales abundaron en cinco o mas individuos. En una corrida de 2.56 billones de instrucciones, se generaron 1180 clases de diferente tamaño, 367 de las cuales abundaron en cinco o mas individuos.
40. Optimización Evolutiva .- Con objeto de comparar el proceso evolutivo entre diferentes corridas del simulador, se necesita una medida simple y objetiva de la evolución. Una de estas medidas (escalas) es el grado con que las criaturas mejoran su eficiencia, lo que también nos ayuda a evaluar el potencial de los sistemas de vida sintética ante el problema de la optimización del código máquina.
41. La eficiencia de la criatura puede evaluarse de dos maneras: el tamaño del genoma y el número de ciclos del CPU necesarios para ejecutar una replicación. Es claro que los genomas pequeños pueden replicarse con menos tiempo de CPU, sin embargo, durante la evolución, las criaturas también reducen la razón de instrucciones ejecutadas en una replicación, el tamaño del genoma. El

número de instrucciones ejecutadas por instrucción copiada, decrece substancialmente.

42. El incremento en eficiencia del algoritmo replicable es aún mayor que el decremento en el tamaño del código. El ancestro es de 80 instrucciones de largo y necesita 839 ciclos de CPU para replicarse. La criatura de tamaño 22, solo necesita 146 ciclos de CPU para replicarse.
43. Complejidad Incrementada .- El rizo desenrollado es un ejemplo de la habilidad de la evolución para producir un incremento en la complejidad, lo que ocurre gradualmente en un periodo largo de tiempo. Lo interesante de la técnica de optimización del rizo desenrollado, es que requiere de un código mas complejo. La criatura resultante tiene un tamaño de genoma de 36, comparado con su ancestro de tamaño 80, aunque ha compactado un algoritmo mucho mas complejo en menos de la mitad del espacio. Este es un ejemplo clásico de un diseño intrincado en la evolución.
44. Biología Sintética .- Los biólogos entrenados tenderán a ver la vida sintética en los mismos términos en que llegaron a conocer la vida orgánica. Estando yo entrenado en ecología y evolución, he apreciado en mis comunidades sintéticas, muchas de las propiedades ecológicas y evolutivas que son bien conocidas en las comunidades naturales. Los biólogos entrenados en otras especialidades, probablemente observen otras propiedades familiares. Parece que solo vemos lo que conocemos. Es posible que nos tome todavía mucho apreciar las propiedades únicas de estas nuevas formas de vida.

EL DESARROLLO DE LA MEMORIA A CORTO PLAZO EN LOS ROBOTS BASADOS EN LA CONDUCTA.

Peter Kronberg
Chalmers University of Technology

1. **Introducción** .- Este documento es una tesis de maestría donde se expone un modelo fenomenológico de memoria a corto plazo.
2. La memoria a corto plazo normalmente es mencionada en el contexto del aprendizaje y de la representación del conocimiento. Sin embargo, Staddon propone un modelo de memoria dinámica que utiliza un sistema de integradores en caída de cascada. Este modelo de "caída" está preparado para mostrar muchas de las propiedades de las formas elementales de aprendizaje, compartidas por muchos animales, como el decremento gradual de la respuesta refleja ante estimulación sucesiva (habituación), así como la curva logarítmica de olvido identificada inicialmente por Ebbinghaus (1850-1909).
3. Nehmzow dispuso de memoria asociativa a una red neuronal artificial, que permitía a un robot adquirir de manera autónoma, la habilidad de esquivar obstáculos (mediante aprendizaje por reforzamiento). También, con la evitación de obstáculos, Chang y Gaudiano utilizaron una red neuronal artificial con instalador múltiple, que aprendía a evitar obstáculos en forma reactiva, mediante esquemas de condicionamiento clásico y operante. La memoria a corto plazo se implementó en la entrada del instalador de la red y permitía al robot aprender la relación causal entre una colisión y el estímulo sensorial presente, justo antes de ella.
4. En tanto que Staddon nos alienta para establecer un modelo funcional de memoria dinámica, la mayoría de los demás estudios en el área, enfocan la memoria a corto plazo como un tipo de protección informática para su uso en alguna regla de aprendizaje, como es el caso en los estudios de Chang y Gaudiano y de Nehmzow.

5. Es un problema, pero es un reto también, el desarrollo de robots u otros agentes artificiales, que puedan llevar a cabo tareas complejas, como la navegación evitando obstáculos (sin un conocimiento predefinido o a priori del mundo). Esta tarea trae como resultado la actualización de diversos tópicos relacionados con la manifestación de inteligencia (algo tan controvertido en el área de los robots basados en la conducta y tan bien logrado en la naturaleza). Pero, aunque el esfuerzo por hacer inteligentes a las máquinas ha sido difícil, esto se debe, en parte, a que el mismo concepto de inteligencia es difícil de definir. En los primeros tiempos de la inteligencia artificial, Alan Turing definía a la inteligencia mediante un experimento donde una persona tenía que descubrir si estaba interactuando con un ser humano o con un sistema artificial. Si era incapaz de identificar a la máquina, esta se consideraba inteligente. En contraste con el enfoque antropocéntrico tradicional descendente de la inteligencia artificial, los robots basados en la conducta utilizan un enfoque ascendente, construyendo sistemas sofisticados, creciendo a partir de un conjunto de comportamientos básicos. Se puede decir que el enfoque basado en la conducta es más razonable, asumiendo que se puede aprender mucho del estudio de conductas simples emitidas por organismos simples.
6. Inspirándose en la biología, el enfoque de los robots basados en la conducta, define a la inteligencia como la habilidad de sobrevivir en un ambiente no estructurado (un ambiente que cambia seguido, de manera inesperada).
7. A pesar de todo, el término inteligencia, como se utiliza comúnmente, implica cierto grado de procesamiento interno (reflexión o habilidad de adquirir y hacer uso del conocimiento). En este respecto, la memoria y el aprendizaje son cruciales. Así, un agente reactivo puede construirse para evadir obstáculos eficientemente, pero un agente reflexivo, con memoria a corto plazo, puede aprender la localización de los obstáculos y consecuentemente planear su ruta de avance.
8. La idea central de esta tesis es, entonces, que la memoria es un aspecto integral de cualquier organismo inteligente. Así, aunque el ambiente sea impredecible,

- debe haber al menos alguna regularidad, de otra manera, la memoria sería inútil.
9. La memoria a corto plazo se define como cualquier estructura que permita la retención no permanente y la reproducción de la información.
 10. **La Memoria y el Aprendizaje en los Organismos Biológicos**
.- Quizá la primera cuestión que llama la atención a muchas personas acerca de la capacidad humana de recordar, no es tanto la fuerza del recuerdo sino su debilidad. Aunque, irónicamente, el olvido es esencial. El olvido es un método para filtrar la experiencia del mundo, despojándola de percepciones sensoriales irrelevantes. Cuando guardar información y cuando recordarla es tan importante como qué es lo que se guarda y cómo lo hace uno.
 11. Aún los organismos simples tienen la habilidad de modular su conducta en base a los estímulos externos, de adaptarse a sus ambientes aprendiendo sobre estímulos particulares (recordándolos). En su forma más elemental, la *memoria* es simplemente la retención de experiencias previas para su reproducción futura.
 12. En la literatura científica, se identifica para los humanos tres tipos de memoria. La memoria a corto plazo (STM), la memoria a largo plazo (LTM) y la memoria de trabajo (WM). Como lo dice su nombre, la memoria de trabajo es primordialmente un mecanismo de soporte que está activo mientras ejecutamos ciertas operaciones (como una fuente temporal para cálculos intermedios, como las sumas parciales durante la multiplicación mental de dos números). Así, la WM es en buena medida voluntaria. En comparación la memoria a corto plazo puede ser vista como un tipo de postimagen residual de una situación, que inevitablemente se mantiene por un periodo corto de tiempo (como un mecanismo reflexivo).
 13. Respecto al tiempo, WM, STM y LTM son muy diferentes. WM es breve y se vacía en cuestión de segundos, STM funciona en cuestión de minutos, mientras que LTM es esencialmente permanente. Adicionalmente, la repetición puede fortalecer la retención y también transformar un breve recuerdo vivo, en memoria a largo plazo.
 14. Actualmente se sabe que hay una diferencia fisiológica fundamental entre la memoria a corto plazo, que se

- manifiesta mediante cambios plásticos en la sinapsis, y la memoria a largo plazo, que se asocia con cambios estructurales a nivel celular.
15. Estudios recientes, empleando bloqueadores químicos, han obtenido evidencias de que la memoria a corto y a largo plazo, son en buena medida procesos separados, siendo posible bloquear una mientras la otra permanece normal. Esto quiere decir que ambas memorias son paralelas y no secuenciales.
 16. Mecanismos auxiliares como la atención, el procesamiento visual y el razonamiento, están fuertemente conectados con el proceso de la memoria. Incluso las emociones afectan a la memoria al modular el comportamiento de las células nerviosas, tanto para inhibir como para facilitar el proceso de consolidación de los recuerdos. Por ejemplo, una experiencia traumática se almacena en la memoria inmediatamente y perdura un largo tiempo, sin la necesidad de aprendizaje repetitivo.
 17. En términos generales, el proceso de la memoria se puede descomponer en los siguientes elementos: **Atención:** ya que el mundo contiene mas información de la que es posible procesar, es necesario un mecanismo que pueda extraer los aspectos mas importantes. Mediante la atención selectiva se decide qué aspectos se procesarán en el cerebro. **Almacenamiento:** es el proceso de retención de la información, el que puede ser momentáneo, transitorio o perdurable. **Reproducción:** la presencia de algún estímulo puede traer una experiencia guardada (a la conciencia) o puede producir una respuesta (inconsciente). **Olvido:** la información se puede perder por una codificación inicial débil, por interferencia entre recuerdos o simplemente como resultado del tiempo.
 18. **Un Modelo simple de Memoria a Corto Plazo .-** Uno entiende muy bien que los mecanismos que subyacen la memoria biológica son extremadamente complejos como para copiarlos con detalle, lo que está definitivamente mas allá del propósito de este trabajo. En lugar de ello, nos enfocamos en capturar las características observables de la memoria biológica, para producir un modelo fenomenológico que sea útil en los robots basados

- en la conducta. Por lo que es necesario proporcionarle al sistema artificial la posibilidad, por un lado, de distinguir entre información relevante e irrelevante y por el otro, permitirle la retención y la reproducción de esta información.
19. Para el propósito de esta tesis se construyó un simulador con el lenguaje Borland Delphi (Pascal orientado a objetos). Se definió como estímulo un vector binario de tres bits. En la simulación que se presenta aquí, el sistema era excitado por un estímulo a la vez, para estimular a un robot que se movía en un ambiente, mientras leía (veía) el mundo, mediante sensores binarios de tres bits.
 20. Algunos estímulos estaban asociados con acciones o comportamientos preferidos.. El mantener una memoria de los estímulos relevantes, permitía al robot predecir que conducta era la mejor la próxima vez que se presentara el mismo estímulo. Esto, en cierta forma, podría verse como una situación de aprendizaje supervisado.
 21. La arquitectura del sistema artificial, que regulaba la dinámica de la memoria a corto plazo estaba compuesta de un conjunto de ecuaciones de memoria. El modelo propuesto debía ser capaz de manejar las siguientes situaciones: (a) mantener un estímulo en la memoria cierta duración, (b) aprender por repetición, (c) descartar recuerdos y reaprender, (d) mostrarse robusto ante el ruido, y (e) manejar estímulos con intervalos entre estímulos variables.
 22. En los casos donde el sistema debía funcionar en un ambiente desconocido, los algoritmos evolutivos proporcionan ventajas, se trata de algoritmos de búsqueda y optimización que se inspiran en la evolución biológica. Mas específicamente, aquí se utilizó un algoritmo genético estándar, para lograr un procedimiento de optimización. Los aspectos asociados con una mayor tasa de recompensas serían inevitablemente optimizados.
 23. Los resultados de nuestro trabajo mostraron que el modelo propuesto de STM tenía la habilidad de reconocer los estímulos y recordar la asociación entre ellos. Adicionalmente, mostró la capacidad de adaptarse a los cambios y reaprender, así como eficientemente sortear

las discrepancias ocasionales, para aprender a comportarse. El modelo se vio capaz de manejar tareas mas difíciles, con estímulos de diferente naturaleza. En ese sentido, el modelo propuesto confrontó su ambiente de una forma que corresponde a la memoria biológica de corto plazo.

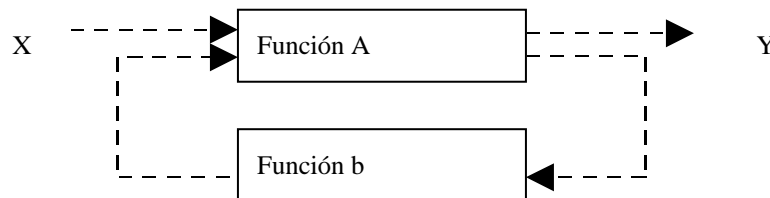
¿ PUEDE UNA COMPUTADORA SENTIR DOLOR ?

David W. Croft
California Institute of Technology

1. **Introducción** .- Para mí el dolor es un input continuo, con propósitos de optimización, en un sistema de feedback. Procedo a aclarar y limitar los términos definatorios en el contexto adecuado. Luego evalúo lo robusto de la definición, demostrando su compatibilidad con el enfoque biológicamente aceptado, intuitiva y filosóficamente. Concluyo que si un dispositivo computacional se diseñara para cumplir con la definición, podría decirse que la computadora sentiría dolor.
2. **Función** .- Una función relaciona un grupo de inputs con un solo output. Para comprender esto veamos las dos definiciones de "función" en seguida : (a) hay una función cuando una variable está relacionada con otra, de manera que por cada valor que asume una, hay un valor determinado en la otra; (b) una función es una regla de correspondencia entre dos grupos, de manera que hay un elemento único asignado en un grupo, para cada elemento del otro.
3. De lo anterior se desprende que una función simplemente relaciona un grupo de puntos entre sí, como en la ecuación de la línea donde consideramos a x como el input y a y como el output: y es una función de $x=f(x)=y=m*x+b$. También podemos replantear diciendo que x es una función de $y=f(y)=x=(y-b)/m$.
4. Una función determinista no azarosa producirá el mismo output y , cada vez que un input x se presente. Esto es, el input x determina completamente el output y . La aleatoriedad es un mito. Por lo que, si uno observa algún output y "al azar", de una función a partir de un input x , uno solo puede concluir que (1) no se han identificado todos los inputs, (2) que el input x subsecuente es ligeramente diferente, lo que, especialmente en las funciones no lineales,

produce una diferencia mensurable en el output y, o (3) que la "función" es en realidad un sistema.

5. **Sistema** .- A diferencia de las funciones, los sistemas son propositivos. Consideremos la siguiente definición: un sistema es la combinación de dos o mas elementos, casi siempre físicamente separados cuando funcionan, necesarios para ejecutar una función operativa. Un sistema, en forma general, se define como la combinación e interconexión de diversos componentes, para llevar a cabo una tarea deseada. Así pues, a diferencia de las funciones, un sistema siempre tiene un propósito deseado, objetivo, tarea o meta.
6. **Sistema de Feedback** .- Un sistema de feedback es capaz de determinar su output, a partir de inputs pasados y presentes. La interconexión de componentes funcionales separados, un sistema, puede permitir que los outputs de una o mas funciones sean redireccionadas para servir como input de una o mas funciones. Así, el sistema ejecutará como una función cuyos inputs incluyen los inputs actuales y cierta forma de la "historia" de los inputs anteriores.



7. En la figura de arriba, el output es una función (A) de los inputs, que incluyen x además de una función (B) de los inputs previos. Se dice que hay una conexión de "re alimentación" al input del sistema. Con esto, por supuesto, se asume que hay una demora real, en lo que el input se propaga a través del sistema y de la conexión de feedback, para determinar el output.

8. **Optimización del Input en un Sistema de Feedback** .- Un input optimizado para un sistema de feedback es un input que cambia la relación de input a una de output o función, dentro de un sistema plástico, para ejecutar mejor su tarea. "Optimizar" es hacer las cosas "lo mejor o mas efectivamente posible".
9. En esta línea de pensamiento, también deberíamos considerar otro tipo de input en un sistema de feedback, el input no optimizado, mismo que puede ser de dos tipos, aquellos inputs que no afectan la ejecución del sistema y aquellos otros que van en detrimento del sistema. Un input que no afecta la ejecución, una "sensación", puede procesarse simplemente como una entrada que determina una salida de una manera funcional dependiendo del estado actual del sistema. Por un lado, puede ser simplemente ignorado y no tener efecto alguno sobre el output. Los inputs que van en detrimento del sistema para ejecutar su tarea, incluyen inputs propositivos, como la desinformación y aquellos que son accidentales, como los daños no racionales o ejecuciones accidentales.
10. **Input continuo, propositivo y optimizador para un Sistema de Feedback** .- Este tipo de input mejora el sistema mientras persista, esto es, todo el tiempo que esté presente, el sistema estará en el proceso de optimización. Debe producir continuamente cambios plásticos en la relación input output de la función del sistema.
11. **Lo que no es el Dolor** .- El dolor es un input continuo, propositivo y optimizador de un sistema de feedback. El dolor no es un input para una función. Una función carece de plasticidad. El dolor es entonces, un simple input, como una sensación, que se relaciona con un output, como la conducta. Mas aún, esta conducta puede no tener ninguna intención, como un objeto en movimiento que cambia de dirección con cada colisión. El dolor no es un input para un sistema de feedback. No todos los inputs de un sistema de feedback son dolorosos. Podemos no considerar el input como significativo o feedback relevante para la tarea que enfrentamos. El dolor no

es un input optimizador de un sistema de feedback. Uno puede percatarse de que no hay información en el input, que el feedback es irracional. El dolor no es un input propositivo optimizador de un sistema de feedback. No necesariamente nos permite hacer mejor la tarea. El que la carrocería de un vehículo se deforme cuando este choca con un obstáculo, en sí mismo, no hace que se logre mejor la meta.

12. **Dolor** .- Así, el dolor es un input continuo, propositivo y optimizador de un sistema de feedback. Para que un input detector de colisiones se considere como dolor, el vehículo debe intentar diferentes trayectorias para llegar a su meta, durante el tiempo que persistan las colisiones. No importa que el vehículo nunca evite completamente todos los obstáculos, el requisito es que continuamente intente optimizar su paso, mientras persista la detección de colisiones. Tampoco importa que el dolor no produzca una mejor solución o que incluso ocasione degradación en la ejecución, mientras el propósito del input continúe siendo el proceso de optimización.
13. Como un ejemplo biológico de esto, consideremos el dolor crónico, donde a pesar de no contar con un remedio, el propósito de la señal de feedback es claro (algo falla en el cuerpo).

APRENDIZAJE POR REFORZAMIENTO COMPUTACIONAL.

Richard S. Sutton
JQBA, 1998

En la década anterior surgió un nuevo enfoque de inteligencia artificial inspirado por y muy cercano a las teorías del aprendizaje. El aprendizaje por reforzamiento, que así se ha denominado a este enfoque, se basa en conceptos computacionales análogos al reforzamiento, las expectativas, los reforzadores secundarios, las huellas de estímulos y las simulaciones mentales. Estas ideas han conducido hacia algoritmos, dentro de las aplicaciones de ingeniería, desde los juegos por computadora, hasta la ubicación de canales en las redes de radio celular. Este documento presenta las ideas principales que han conducido al éxito del aprendizaje por reforzamiento. También se considera la manera en que este enfoque podría pagar su deuda con la psicología. El aprendizaje por reforzamiento proporciona una nueva perspectiva computacional para las teorías psicológicas.

SUBIGUALAMIENTO EN UN ORGANISMO SIMULADO POR COMPUTADORA.

Matt J. Morris
JQBA, 1998

Se desarrolló un organismo simulado por computadora que pudiera aprender mediante reforzamiento y castigo. Se le ubicó en un ambiente de dos dimensiones, con cuatro paredes y dos palancas. Se otorgaba castigo cuando chocaba contra las paredes y se le reforzaba contingentemente cuando presionaba las palancas. El organismo podía moverse hacia arriba y hacia abajo, a la izquierda y a la derecha, movía su brazo izquierdo y su brazo derecho. El organismo mostró diversos fenómenos operantes como adquisición, extinción y mantenimiento de la conducta mediante reforzamiento parcial, así como conducta de evitación. El organismo mostró algún control de estímulo, pero no evidenció los patrones típicos asociados con los diferentes programas de reforzamiento. Finalmente, bajo una serie de programas VI concurrentes, el análisis de las tasas de respuesta vs. las tasas de reforzamiento mostró subigualación, con exponente de .83 y un valor de r^2 de 87.5%.

SE NECESITAN MODELOS CONDUCTUALES COMPLETOS PARA EL PROGRESO DE LAS CIENCIAS DE LA CONDUCTA.

William R. Hutchison
JQBA, 1998

En este documento se describe como se construyó un modelo computacional de organismo operante, a partir de consideraciones cuantitativas derivadas del análisis experimental de la conducta. El modelo es susceptible de ser conectado a sensores y activadores robóticos, aunque generalmente se comporta desde un ambiente computacional diseñado para administrar gran variedad de contingencias, que incluyen programas de entrenamiento explícito. Los modelos analíticos conductuales son diferentes de otros modelos relacionados construidos con perspectivas biológicas o computacionales. Para mayor información, ponerse en contacto con Behavior Systems LLC, Boulder, CO.