CURSO 2018-19   ASIGNATURA: ***INTELIGENCIA ARTIFICIAL***

Profesores:

León González Sotos, Dpto. **Ciencias de la Computación**   DN325,
Correl : leon.gonzalez@uah.es

Álvaro Somolinos Yagüe, Dpto. **Ciencias de la Computación**

TEMARIO

Tema 1º Introducción, panorama histórico y conceptual

Tema 2º Búsqueda desinformada

Tema 3º Búsqueda heurística

Tema 4º Problemas de satisfacción de restricciones

Tema 5º Juegos

Tema 6º Computación flexible

Tema 7º Representación, razonamiento, aprendizaje

BIBLIOGRAFÍA GENERAL

Luger, *Artificial Intelligence*, Addison Wesley 2005

Palma-Marín, Inteligencia Artificial: métodos técnicas y aplicaciones, McGraw-Hill 2008

Nilsson, *Inteligencia Artificial*, McGraw-Hill   http://aima.cs.berkeley.edu/index.html

Poole-Mackworth, *Artificial Intelligence*, Cambridge Univ. Press   2010
file:///C:/Users/leon.gonzalez/Downloads/Cambridge_ArtificialIntelligence.pdf

Russell-Norvic*, Inteligencia Artificial*, Pearson

Fdez.Galán-Glez.Boticario,Mira, *Problemas resueltos de Inteligencia Artificial*, Addison- Wesley

CALIFICACIÓN

La evaluación constará de una parte correspondiente a exposición teórica   (50%) y otra a laboratorio (50%). Estas partes se desglosarán:

A) Por evaluación continua: tres pruebas teórico- prácticas (teoría, cuestiones, problemas y ejercicios sobre laboratorio) a realizar:
PEI-1ª 20 febrero y PEI-2ª 27 marzo, en el aula y horario de teoría
PEI-3ª 15 mayo, (se realizará junto a la prueba de los alumnos excluidos de la evaluación continua, en aula y horario a determinar por la escuela)
Estas pruebas puntuarán 10, 15 y 25 puntos respectivamente (para el 50% de parte expositiva) y 4, 7, y 9 puntos respectivamente (para la parte de ejercicios de laboratorio).

El resto de la evaluación de laboratorio consistirá en la entrega de dos ensayos escritos y en la entrega y defensa de dos proyectos, a presentar y defender en fechas que se anunciarán y que completarán los 5 +10 + 5+10 puntos restantes de la nota.

B) Por examen final (para alumnos que obtengan dispensa de la evaluación continua): examen teórico -práctico (80% de la nota), en las fechas establecidas por la Escuela para los exámenes de mayo y extraordinario de junio) más entrega y defensa de un proyecto antes del correspondiente examen (20% de la nota).

DOCENCIA

A) Grupos de teoría: 3A: Aula NA6 (Miércoles 10-12h)

B) Grupos de prácticas: 3A.1: Laboratorio NL6 (Miércoles 8-10 h)

3A.2: Laboratorio NL6 (Miércoles 12-14h)

TEMA1 INTRODUCCIÓN, PANORAMA HISTÓRICO Y CONCEPTUAL

1.-Génesis de las ideas sobre Computación y sobre Inteligencia Artificial

a) En el ámbito formal y abstracto.

b) En el ámbito de la ingeniería y de la Física

c) Neurociencia, Psicología y Economía

d) Características

Observación importante: lo que sigue es una guía de lecturas y estudio. **No** constituye el contenido del tema, ni substituye a los textos.

a) En el ámbito formal y abstracto:

Proceso de formalización de las Matemáticas:

-Mesopotamia, Egipto, China: conocimientos matemáticos empíricos

-Aristóteles (384 a.C) Organon: silogismo, primeros principios, Lógica

-Euclides (325 a.C.): Matemática empírica + Lógica = demostración, método axiomático matemático.

-Raimundo Lulio (+1315), *Ars Magna*, postulación del cálculo lógico y mecánico

-Leibniz (1646-1716), *Characteristica Universalis*, postulación de lenguajes formales para calcular con conceptos.

-Boole (1815), De Morgan (1806) Peirce (1839), Schröder(1841), Frege (1889), álgebra de la lógica, cálculo de proposiciones y de predicados

-Dificultades: demostración del 5º postulado de Euclídes, geometrías no euclídeas; Teoría de conjuntos, paradojas del infinito, enunciados indecidibles (último teorema de Fermat): Crisis de fundamentos, problemas e intento de formalización finitista de Hilbert

-Respuestas (negativas) y logros (positivos) para la Computación: Gödel (1936), Turing,Post Church, Kleene (acotación del concepto de computable, modelos abstractos de la computación, funciones recursivas, λ-cálculo, máquinas de Turing y Post), postulación del pensamiento por máquinas (Turing).

-Tesis de Church-Turing: Toda función efectivamente computable puede llevarse a cabo mediante una máquina universal de Turing (o, equivalentemente mediante el λ-cálculo, o mediante funciones recursivas). Nota: aunque equivalentes, el de máquina Turing es un enfoque desde el punto de vista "hardware"; el λ-cálculo y las funciones recursivas aportan el punto de vista "software" sobre la computación.

b) <u>En el ámbito de la ingeniería y de la Física</u>

-Autómatas mecánicos e hidráulicos (época helenística, Alejandría)

-1642 y 1671: Calculadoras mecánicas de Pascal y de Leibniz

-Máquinas diferencial (1822) y analítica (1833-42) de Babage

-Calculadoras electromecánicas (Zuse y otros)

-Ordenador de Von Neumann (1945)

(Ver Russell-Norvig y Nilsson)

c) <u>Filosofía, Neurociencia, Psicología , Economía</u>

-Gómez Pereira (1554), Descartes (1596), problema cuerpo-mente.

-Ramón y Cajal, McCullogs, Pitts, Redes neuronales.

-von Neumann y Morgenstern, teoría de juegos

(Ver los capítulos introductorios de los libros de Russell-Norvig y de Nilsson)

d) Características importantes de la I.A.

-Se usa más información simbólica que numérica

-Se usan métodos heurísticos más que deterministas

-Se usa conocimiento específico-declarativo e información imperfecta (incompleta o incierta)

2.-Conceptos de inteligencia. Ver:

-- Russell-Norvig cap.1

--Luger, cap.1

--Shane Legg & Marcus Hutter, *Universal intelligence: a definition of Machine Intelligence* https://arxiv.org/pdf/0712.3329.pdf

3.-Inteligencia artificial, fines y medios. Relación con otras disciplinas.

Se trata de construir agentes artificiales (hoy por hoy, sistemas informáticos) que emulen el comportamiento de agentes naturales (personas, animales, acaso grupos o sociedades) cuando resuelven problemas siguiendo fines, percibiendo su entorno y actuando sobre él para adaptarse con éxito al mismo. Esta descripción, genérica, es lo bastante flexible como para poder abarcar el funcionamiento de un simple climatizador, el comportamiento de un termitero o el juego de un gran maestro de ajedrez. También es susceptible de cuantificación: se puede determinar criterios numéricos de comparación de su grado.

Claramente, esto relaciona el campo de la I.A., hoy entroncado en la Computación y encarnado en la Informática, con los de las ciencias formales (Matemática, Lógica, Epistemología, Ontología), la Neurociencia, la Ciencia Cognitiva, la Teoría de la Decisión, la Teoría de Juegos y la Economía.

<u>Paradigma simbólico</u> Hipótesis del símbolo físico (Newell y Simon 1976): la condición necesaria y suficiente para qué un sistema físico muestre comportamiento inteligente es que sea un *Sistema de Símbolos Físico*.

Por S.S.F se entiende algo capaz de representación de signos. La hipótesis postula que el uso de símbolos basta para representar el mundo, y que el diseño de mecanismos de búsqueda simbólica, especialmente heurística, para explorar el espacio de potenciales inferencias que estos sistemas simbólicos puedan presentar basta para emular la inteligencia, con independencia del medio de implementación física.

Nótese que esto equivale, *grosso modo*, a decir que: (a) dando **por anticipado** una lista de instrucciones (lo bastante detalladas y complejas), escritas en un **reglamento escrito permanente y definitivo** a un empleado que se lo tome siempre al pie de la letra actuando a ciegas (sin entender nada de lo que haga), (b) que opere en una oficina con ficheros muy grandes (de fichas de cartulina como las de las bibliotecas antiguas), fichas que **contengan sólo cadenas de cifras** (que pueden representar codificaciones numéricas preestablecidas de símbolos) y un escritorio sobre el que puede ejecutar las instrucciones del reglamento, consistentes sólo en **órdenes para extraer** fichas, colocarlas sobre la mesa, hacer cambios estipulados borrando **parte de su contenido** o **añadiendo contenido** de otras fichas y a **devolverlas posiciones** del fichero, (c) dando tiempo suficiente, es posible emular el comportamiento inteligente de personas o animales que resuelven **problemas nuevos**, no planteados previamente en un ámbito dado. Una apuesta audaz.

Paradigma conexionista (redes neuronales y algoritmos genéticos)

El comportamiento inteligente puede surgir a partir de la capacidad de entrenamiento de redes de neuronas artificiales y el aprendizaje (no simbólico) por parte de las mismas, en procesos que imiten la evolución adaptativa.

4.-Aplicaciones y logros (ver cuadro cronológico Timeline of Artificial Intelligence https://en.wikipedia.org/wiki/Timeline_of_artificial_intelligence)

Samuel (1952): Programas para jugar a las damas.

Newel l y Simon (1956): *Logic Theorist*, programa para descubrir demostraciones en lógica proposicional.

Cálculo simbólico (Mathematica, Maple, MatLab, etc)

Verificación de programas

Timesharing (Proyecto MAC Machine-aided cognition, multiple-acess computing del MIT); Interfaces Gráficos de Usuario, Windows (Proyectos deXeros en Palo Alto); Paradigmas de programación(propagación de restricciones, usad a en hojas de cálculo, orientación a objeto, funcional, lógica).

Juegos, razonamiento automátizado, sistemas expertos, reconocimiento de lenguaje natural hablado o escrito, visión por computador, Planificación *scheduling* y robótica, clasificación heurística etc. (ver : http://wwwformal.stanford.edu/jmc/whatisai/node2.html)

5.-Campos de la Inteligencia artificial.

Inteligencia Artificial Lógica, Búsqueda, Reconocimiento de patrones, Representación, Inferencia, Sentido Común y Razonamiento etc.

(Ver: http://www-formal.stanford.edu/jmc/whatisai/node2.html)

APÉNDICE: Sobre la formalización creciente del lenguaje matemático

http://www.euclides.org/menu/elements_esp/indiceeuclides.htm

NÚMEROS DE PEANO:

https://es.wikipedia.org/wiki/Axiomas_de_Peano

AXIOMAS DE LA TEORÍA DE CONJUNTOS:

https://www.uv.es/ivorra/Libros/Axiomas.pdfhttps://www.uv.es/ivorra/Libros/Axiomas.pdf

GEOMETRIA (ELEMENTOS, EUCLIDES) FORMULACIÓN ORIGINAL TRADUCIDA

Euclides, *Elementos*:

DEFINICIONES

1. Un punto es lo que no tiene partes.
2. Una línea es una longitud de anchura.
3. Los extremos de una línea son puntos.
4. Una línea recta es aquella que yace por igual respecto de los puntos que están en ella.
5. Una superficie es lo que sólo tiene longitud y anchura.
6. Los extremos de una superficie son líneas.
7. Una superficie plana es aquella que yace por igual respecto de las líneas que están en ella.
8. Un ángulo plano es la inclinación mutua de dos líneas que se encuentran una a otra en un plano y no están en línea recta.
9. Cuando las líneas que comprenden el ángulo son rectas el ángulo se llama rectilíneo.
10. Cuando una recta levantada sobre otra recta forma ángulos adyacentes iguales entre sí, cada uno de los ángulos iguales es recto y la recta levantada se llama perpendicular a aquella sobre la que está.
11. Angulo obtuso es el (ángulo) mayor que un recto.
12. Angulo agudo es el (ángulo) menor que un recto.
13. Un límite es aquello que es extremo de algo.
14. Una figura es lo contenido por uno o varios límites.
15. Un círculo es una figura plana comprendida por una línea [que se llama circunferencia] tal que todas las rectas que caen sobre ella desde un punto de los que están dentro de la figura son iguales entre sí.
16. Y el punto se llama centro del círculo.
17. Un diámetro del círculo es una recta cualquiera trazada a través del centro y limitada en ambos sentidos por la circunferencia del círculo, recta que también divide el círculo en dos partes iguales.
18. Un semicírculo es la figura comprendida entre el diámetro y la circunferencia por él cortada. Y el centro del semicírculo es el mismo que el del círculo.
19. Figuras rectilíneas son las comprendidas por rectas, triláteras las comprendidas por tres, cuadriláteras las comprendidas por cuatro, multiláteras las comprendidas por más de cuatro rectas.
20. De entre las figuras triláteras, triángulo equilátero es la que tiene los tres lados iguales, isósceles la que tiene sólo dos lados iguales, y escaleno la que tiene los tres lados desiguales.
21. Además, de entre las figuras triláteras, triángulo rectángulo es la que tiene un ángulo recto, obtusángulo la que tiene un ángulo obtuso, acutángulo la que tiene los tres ángulos agudos.
22. De entre las figuras cuadriláteras, cuadrado es la que es equilátera y rectangular, rectángulo la que es rectangular pero no equilátera, rombo la que es equilátera pero no rectangular, romboide la que tiene los ángulos y lados opuestos iguales entre sí, pero no es equilátera ni rectangular; y llámense trapecios las demás figuras cuadriláteras.
23. Son rectas paralelas las que estando en el mismo plano y siendo prolongadas indefinidamente en ambos sentidos, no se encuentran una a otra en ninguno de ellos.

POSTULADOS

1. Postúlese el trazar una línea recta desde un punto cualquiera hasta un punto cualquiera.
2. Y el prolongar continuamente una recta finita en línea recta.
3. Y el describir un círculo con cualquier centro y distancia.
4. Y el ser todos los ángulos rectos iguales entre sí.
5. Y que si una recta al incidir sobre dos rectas hace los ángulos internos del mismo lado menores que dos rectos, las dos rectas prolongadas indefinidamente encontrarán en el lado por el que están los (ángulos) menores que dos rectos.

NOCIONES COMUNES

1. Las cosas iguales a una misma cosa son también iguales entre sí.
2. Y si se añaden cosas iguales a cosas iguales, los totales son iguales.
3. Y si de cosas iguales se quitan cosas iguales, los restos son iguales.
4. Y las cosas que coinciden entre sí son iguales entre sí.
5. Y el todo es mayor que la parte.

Los axiomas de Euclides para la geometría (EN TÉRMINOS ACTUALES)

Postulados

1. (Es posible) trazar una línea recta desde cualquier punto a cualquier otro.
2. (Es posible) prolongar continuamente en línea recta una recta dada.
3. (Es posible) trazar un círculo con cualquier centro y distancia (radio).
4. Todos los ángulos rectos son iguales entre sí.
5. Si una recta incide sobre otras dos formando del mismo lado ángulos internos menores que dos rectos, al prolongarlas indefinidamente se encontrarán por el lado en que los ángulos sean menores que dos rectos.

Nociones comunes

1. Cosas que son iguales a una misma cosa son también iguales entre sí.
2. Si a cosas iguales se suman cosas iguales, los totales son iguales.
3. Si a cosas iguales se restan cosas iguales, los restos son iguales.
4. Cosas que encajen cada una en la otra son iguales entre sí.
5. El todo es mayor que la parte.

Observaciones:

1. El postulado 5 puede ser reformulado como

   5*. Dado una línea recta y un punto exterior a ella puede trazarse una única línea paralela a la recta dada que pase por ese punto.

EN LENGUAJE NATURAL

Los axiomas de Peano para la aritmética

Los axiomas de Peano (debidos en realidad a Richard Dedekind) tal como fueron escritos (en latín) originariamente son:

(1) 1 es un número natural.
(2) El sucesor inmediato de un número natural también es un número natural.
(3) 1 no es el sucesor inmediato de ningún número natural.
(4) Dos números naturales distintos no tienen el mismo sucesor inmediato.
(5) Toda propiedad verificada por 1 y por el sucesor inmediato de todo número que también verifique esa propiedad, es verificada por todos los números.

En términos matemáticos los axiomas suelen expresarse de este modo (donde la letra S representa al sucesor inmediato de un número, y $S(n)$ debe pensarse como el número que sigue a n, es decir, $n+1$):

(1) 1 es un número natural.
(2) Si n es un número natural, $S(n)$ es un número natural.
(3) No existe n tal que $S(n) = 1$.
(4) Si $n \neq m$, entonces $S(n) \neq S(m)$.
(5) Si P es una propiedad tal que 1 verifica P y vale que si n verifica P, entonces también $S(n)$ verifica P, puede concluirse que todo número natural verifica P.

EN LENGUAJE FORMALIZADO (CALC. PREDICADOS)

Los ***cinco axiomas*** de Peano son:

$$\begin{aligned}
&A_1 : N(1)\\
&A_2 : \forall x(N(x) \rightarrow N(x'))\\
&A_3 : \neg\exists x(N(x) \wedge 1 = x')\\
&A_4 : \forall x\forall y((N(x) \wedge N(y) \wedge x' = y') \rightarrow x = y)
\end{aligned}$$

Dos variantes del 5° axioma: en lógica de primer orden (en realidad un esquema de axioma).

$$A_5 : \phi(1) \wedge \forall x\Big((\phi(x) \rightarrow \phi(x')) \rightarrow \forall x\ \phi(x)\Big)$$

En lógica de segundo orden:

$$A_5' : \forall\phi\Big(\phi(1) \wedge \forall x\big((\phi(x) \rightarrow \phi(x')) \rightarrow \forall x\ \phi(x)\big)\Big)$$

Además ***dos definiciones ,de la suma y de la multiplicación***, que a veces se presentan como axiomas.

- Definiciones de suma y multiplicación:

$$D_1 : \begin{array}{l} n + 1 = n' \\ n + m' = (n + m)' \end{array}$$

$$D_2 : \begin{array}{l} n \times 1 = n \\ n \times m' = (n \times m) + n \end{array}$$

- Axiomas de la suma y de la multiplicación:

$$A_6 : \begin{array}{l} \forall n(n + 1 = n') \\ \forall n\forall m(n + m' = (n + m)') \end{array}$$

$$A_7 : \begin{array}{l} \forall n(n \times 1 = n) \\ \forall n\forall m(n \times m' = (n \times m) + n) \end{array}$$

AXIOMATIZACION DE LA TEORIA DE CONJUNTOS

La teoría de conjuntos de Zermelo-Fraenkel

La teoría se formula en la lógica de primer orden con identidad y tiene un símbolo de relación binario $\in$. Los axiomas son los siguientes:

(1) $\forall xy(x = y \leftrightarrow (\forall z)(z \in x \leftrightarrow z \in y))$ (Extensionalidad)
Intuitivamente, los conjuntos x e y son iguales si sólo si x e y tienen los mismos elementos.

(2) $\exists x \forall y(\neg y \in x)$ (Conjunto vacío)
Intuitivamente, existe un conjunto sin elementos. Puede probarse que es único con esta propiedad y se lo denota $\varnothing$.

(3) $\forall xy \exists z \forall u(u \in z \leftrightarrow u = x \vee u = y)$ (Pares)
Intuitivamente, si x e y son conjuntos, también es un conjunto $\{x, y\}$.

(4) $\forall x \exists y \forall z(z \in y \leftrightarrow (\exists w)(z \in w \wedge w \in x))$ (Uniones)
Intuitivamente, si x es un conjunto, entonces también es un conjunto $\cup x$.

(5) $\forall x \exists y \forall z(z \in y \leftrightarrow (\forall w)(w \in z \rightarrow w \in x))$ (Conjunto de las partes)
Intuitivamente, si x es un conjunto, también es un conjunto el que tiene por elementos a todos los subconjuntos de x.

(6) $\exists x(\exists y(y \in x) \wedge \forall y(y \in x \rightarrow \exists z(y \in z \wedge z \in x)))$ (Infinito)
Intuitivamente, existen conjuntos infinitos. En particular, consideremos los siguientes conjuntos:
$\varnothing$, que llamamos «0»
$\{\varnothing\}$, un conjunto con un solo elemento, que llamamos «1»
$\{\varnothing, \{\varnothing\}\}$, un conjunto con dos elementos, que llamamos «2»
$\{0, 1, 2\}$, un conjunto con tres elementos, que llamamos «3»
$\{0, 1, 2, 3\}$, que llamamos «4»
etcétera.
El conjunto $\{0, 1, 2, 3, 4, \ldots\}$ satisface la condición del axioma: es no vacío, y para cada elemento n del conjunto, hay otro $(n + 1)$ tal que $n + 1$ pertenece al conjunto y n pertenece a $n + 1$.

Este axioma postula la existencia de un conjunto infinito actual, dado «todo a la vez».

(7) $\forall x(\exists y(y \in x) \rightarrow \exists y(y \in x \wedge \neg \exists z(z \in y \wedge z \in x)))$ (Regularidad)
Intuitivamente, todo conjunto no vacío es disjunto de alguno de sus elementos.

(8) $\forall x(\exists! z \varphi(x, z, u, v_1 \ldots v_n) \rightarrow \exists y \forall z[z \in y \leftrightarrow \exists x(x \in u \wedge \varphi(x, z, u, v_1 \ldots v_n))]$
(Reemplazo) donde φ es una fórmula en la que la variable y no ocurre y $\exists! z$ significa «hay un único z».
Intuitivamente, si $F(x)$ es el único z tal que satisface $\varphi(x, z \ldots)$, entonces $\{F(x) : x \in u\}$ es un conjunto.

En la teoría de Zermelo-Fraenkel, gracias al axioma (6) de infinito, puede probarse la consistencia de la aritmética. Los números naturales se obtienen como

$$
\begin{aligned}
0 &= \varnothing \\
1 &= \{\varnothing\} \\
2 &= \{\varnothing, \{\varnothing\}\} \\
3 &= \{0, 1, 2\} \\
4 &= \{0, 1, 2, 3\} \\
&\text{etcétera.}
\end{aligned}
$$

BIBLIOGRAFÍA Y ENLACES

-Nilsson, *I.A.* cap.1

-Russel-Norvig, *I.A.* Cap .1

-Stemberg (ed.) *Handbook of Intelligence*, Cambridge U. Press 2000

-Trillas*, La I.A. y su entorno conceptual*, en: La ciencia en tus manos, Espasa 2001

http://www.turing.org.uk/turing/

http://www-formal.stanford.edu/jmc/whatisai/ (Artículos: What is A.I.?)

-Copeland, *The essential Turing,* Oxford University Press 2004

-Davis, *La computadora universal .De Leibniz a Turing,* Debate 2002

-Chris Dixon, *How Aristotle created the computer*, The Atlantic 2017

-Eco, *El sueño de la lengua perfecta,*

-Gómez Pereira, *Antoniana Margarita,* (Medina del Campo 1554)

http://www.filosofia.org/cla/per/1993band.htm

http://symploke.trujaman.org/index.php?title=G%F3mez_Pereira

-Gödel, *Obras completas*, Alianza 1989

-Graubard (comp.), *El nuevo debate sobre la Inteligencia Artificial,* Gedisa 1999

-Gustafsson, *El extraño animal del norte,* Anaya-M. Muchnik 1992

-Feynman, *Conferencias sobre computación,* Crítica 2003

-Shane Legg & Marcus Hutter, *Universal intelligence: a definition of Machine Intelligence* https://arxiv.org/pdf/0712.3329.pdf

-Minsky, *La máquina de las emociones,* Debate 2010

-Mosterín, *Los lógicos,* Espasa 2000

-Martínez-Piñero, *Gödel para todos,* Seix Barral 2009

-R. de Soto-Trillas, *Algunos genios de la computación,* Universidad de León 2006

-Sotos Ochando, *Gramática de la lengua universal,* Madrid 1863

-Sánchez Vega, *Estudio comparativo del mecanicismo animal en Pereyra y Descartes*, en: http://www.filosofia.org/cla/per/1954veg9.htm

-Francisco Suárez, *De Anima* (Lyon, 1621)

-Francisco de Vallés, *De Sacra Philosophia* (1587) N.B.: Suarez y Vallés (el divino Vallés, sepultado en la capilla de la universidad) fueron profesores en Alcalá, estudiaron entre otras cosas el problema del automatismo animal, relacionado con el problema cuerpo-mente(ver http://www.filosofia.org/cla/per/1993band.htm) Probablemente Vallés transmitió a Descartes ideas de Gómez Pereira que éste usó sin reconocer: compárese "Quidquid noscit est;

ergo sum" (Gómez Pereira, *Antoniana Margarita*, 1554) con "Cogito,ergo sum" (Descartes*,Discurso del método* 1637) http://www.filosofia.org/pereira.htm http://www.filosofia.org/cla/per/1954veg9.htm

TEMA 2 BÚSQUEDA SIN INFORMACIÓN (a ciegas, no heurística)

1.- Formulación de problemas en espacios de estados.

2.- Ejemplos de problemas. Concepto de solución.

3.- Búsqueda en anchura.

4.- Búsqueda en profundidad. Profundidad limitada. Profundidad iterativa.

5.- Búsqueda de coste uniforme.

6.- Búsqueda bidireccional.

7.- Características y comparación de las distintas búsquedas.

BIBLIOGRAFÍA

-Poole-Mackworth, secs. 3.1-3.5

-Russell-Norvig II.3

-Nilsson II.8

-Galán-Boticario-Mira, Problemas resueltos de IA, Addison-Wesley, cap.1

1.- FORMULACIÓN DE PROBLEMAS EN ESPACIOS DE ESTADOS.

Los problemas de que se va a tratar en este tema son los más sencillos. Se trata de situaciones en las que el agente tiene unos fines o metas, que puede perseguir buscando sin incertidumbre en la representación que posea sobre un micromundo , representación que consiste en un modelo formal (estructuras de datos sobre los que se pueda computar) del mismo, que lo represente abstractamente en sus rasgos fundamentales (en relación con el problema) como un **espacio de estados** (estado que el agente puede identificar cuando está en ellos)donde uno o algunos de los mismos son **iniciales** o de partida, desde los cuales se ha de llegar o encontrar, mediante **transformaciones o cambios estado** a uno o varios **estados meta** , **solución** u **objetivo.**

Los estados meta se identifican o valoran mediante algún o algunos **test** o **criterios** de que dispone el agente. Los cambios de estado se producen mediante la aplicación de **operadores de cambio de estado** o de **funciones sucesor** que determinan totalmente los estados a los que se pueda pasar a partir de uno dado (estados sucesores del mismo), y que el agente conoce perfectamente, como conoce el **coste** (si existe) de dichos cambios.

2.-EJEMPLOS DE PROBLEMAS. CONCEPTO DE SOLUCIÓN.

Para ejemplos, ver los clásicos (dos jarras, misioneros y caníbales, n-puzzle , n-reinas etc.) presentados en la relación de ejercicios 2.1 y en las secciones introductorias de las referencias bibliográficas dadas.

Por **resolución** del problema se entiende el proceso de determinar la serie de acciones de cambio de estado que hay que realizar (el camino a seguir) para llegar desde un estado inicial a uno, o varios estados meta o solución, proceso sometido o no, según el contexto del problema, a la consecución de un coste mínimo (**solución óptima**) o a la obtención de estados que, de acuerdo con algún criterio de valoración dado, sean lo bastante satisfactorios

Cuando un problema haya sido formulado de este modo, basta identificar cada uno de sus estados con un nodo de un grafo, y cada uno de los operadores de cambio de estado con la arista que une los nodos correspondientes al estado de partida y al de llegada, y se tendrá el problema convertido en uno de búsqueda en grafos.

PROCEDIMIENTO GENERAL DE BÚSQUEDA EN GRAFOS

1: **Procedure** *Búsqueda*(*G*, *S*, *EsMeta*)
2:     **Inputs**
3:         *G*: grafo con nodos *N* y arcos *A*
4:         *S*: conjunto de nodos iniciales
5:         *EsMeta* : función booleana de los estados (test de solución)
6:     **Output**
7:         camino de un elemento de *S* a nodo para el que *EsMeta* dé verdad
8:         o $\perp$ (no encontrado camino) si no hay caminos solución
9:     **Local**
10:         *ABIERTOS*: conjunto de caminos(listas de nodos de cada camino)
11:     *ABIERTOS* $\leftarrow\{\langle s\rangle : s \in S\}$
12:     **while** (*ABIERTOS* $\neq \{\}$)
13:         **select** y **remove** $\langle s_0,\ldots,s_k\rangle$ de *ABIERTOS*
14:         **if** (*EsMeta*(s_k)) **then**
15:             **return** $\langle s_0,\ldots,s_k\rangle$
                Stop
16:         *ABIERTOS* $\leftarrow$ *ABIERTOS* $\cup \{\langle s_0,\ldots,s_k,s\rangle : \langle s_k,s\rangle \in A\}$
17:     **return** $\perp$

Nota: *ABIERTOS* suele ser llamada también *FRONTERA.*

Los diversos métodos de búsqueda (anchura , profundidad, optimal o de coste uniforme etc.) se diferencian esencialmente en cómo se seleccione - línea 13 del algoritmo- el elemento de la lista *ABIERTOS* (tratándola como cola, como pila, como lista de prioridad según el coste acumulado etc) :

LOS PRINCIPALES MÉTODOS DE BÚSQUEDA DESINFORMADA SON:

3.-BÚSQUEDA EN ANCHURA

La lista *ABIERTOS* se gestiona como una cola, donde los caminos se van extrayendo de la lista en el orden en que entraron a ella (FIFO), y sus nodos terminales se van examinando (mediante la función test *EsMeta*) para presentar al camino como solución (si lo fuera) o generar los sucesores del nodo terminal e incorporar los caminos resultantes de “empalmar” las aristas correspondientes a cada sucesor al camino examinado e incorporar estos caminos alargados a la lista *ABIERTOS*.

4.-BÚSQUEDAS EN PROFUNDIDAD

-Búsqueda en profundidad*: ABIERTOS* se gestiona como pila (LIFO).

Variantes de profundidad:

- Profundidad acotada *L*: se procede como en la búsqueda en profundidad, pero hasta un límite preestablecido ***L*** para la misma, es decir, “podando” las ramas del árbol de búsqueda con dicha longitud ***L***.

-**Profundidad iterativa**: se establece una sucesión creciente de limites $L_1 < L_2 < L_3 < \ldots$

y se repite, <u>desde el principio</u>, una serie de búsquedas en profundidad limitada L_i hasta que una de ellas encuentre la solución.

Nota: Tanto en la búsqueda en anchura como en la de profundidad y sus variantes, se da por supuesto que los costes de las aristas (es decir, de aplicar los operadores para transitar de unos nodos a otros) son siempre 1, de modo que los “gastos” hechos al recorrer los caminos dependen sólo de (son iguales a) el número de etapas de los mismos, que coincide con la profundidad de las ramas del árbol de búsqueda correspondientes.

5.-BÚSQUEDA DE COSTE UNIFORME

-Búsqueda llamada “**optimal**”, “**de menor coste”** o **“de coste uniforme”** (sic), se aplica cuando los costes de los arcos entre nodos no son constantes, de modo que el gasto hecho al seguir los caminos no depende sólo del número de etapas de los mismos, sino que caminos con menor número de etapas que otros puedan resultar más caros que éstos. La lista *ABIERTOS* se trata como una lista de prioridad, eligiendo primero para ser examinado con *EsMeta* aquel camino de la lista que menor gasto acumulado lleve (o uno de los éstos si hubiera varios con idéntico menor gasto).

6.-BÚSQUEDA BIDIRECCIONAL

La **Búsqueda bidireccional** es aplicable sólo cuando se conocen los operadores de paso inversos (es decir, se sabe cómo generar todos los “padres” de cada nodo) y además es conocido explícitamente algún estado solución. En esta situación, se podría resolver el problema del modo habitual, *buscando hacia adelante* desde el estado inicial hacia la meta, o también se podría *buscar hacia atrás*, es decir desde la meta hacia el inicial.

Pero una tercera posibilidad es hacer ambas búsquedas simultáneamente, en pasos alternados, hasta que los segmentos de caminos generados se "empalmen" al coincidir en un estado común (es decir, cuando aparezca un estado común en ambas listas de abiertos) . Ello ocurrirá necesariamente si por lo menos una de las dos búsquedas se efectúa en anchura, con lo cual siempre aparecen en su lista *ABIERTOS* nodos de cada uno de los caminos solución.

Se podría hacer las dos búsquedas en anchura, pero hacerlo combinando los dos tipos permite economizar gasto en memoria en la dirección en que el factor de ramificación sea mayor. En cualquier caso, el número de operaciones de examen y generación de sucesores disminuirá sustancialmente, al llegar cada una de las dos búsquedas sólo hasta la mitad de profundidad. Esta ventaja en coste computacional hay que ponerla en balance frente a la desventaja de tener que comprobar en cada paso (dado que en principio la longitud del camino solución será desconocida) si ha aparecido alguna coincidencia entre los nodos de las dos listas de abiertos.

7.- CARACTERÍSTICAS Y COMPARACIÓN DE SOLUCIONES

Para comparar la eficacia de los métodos se estudian generalmente las siguientes características:

Completitud: un método es completo si siempre encuentra solución, cuando esta existe.

Optimalidad: cuando encuentra una solución, esta es la más cercana al estado de partida, sea en número de pasos, sea en coste acumulado de recorrerlos cuando este no es constante.

Complejidad espacial o en **memoria**: dada por la estimación del tamaño de las listas a gestionar y mantener, en función del factor de ramificación y la profundidad de las soluciones

Complejidad temporal o **en operaciones**: dada por la estimación de la cantidad de nodos o estados a generar y examinar, también en función del factor de ramificación y la profundidad de las soluciones.

Cuestiones 2.1

1) Comprobar que la búsqueda en anchura es un caso particular de la búsqueda o optimal.

2) Comprobar, mediante ejemplos, que las búsquedas en profundidad y en profundidad acotada, pueden fallar, es decir, no encontrar ninguna meta, aunque las haya accesibles desde un estado inicial (es decir, estas búsquedas no son **completas**).

3) Razonar por que las búsquedas en anchura, profundidad iterativa y optimal no pueden fallar, es decir, si existe alguna meta accesible desde un estado inicial, acaban encontrándola (es decir, son **completas**).

4) Comprobar que las búsquedas en anchura y optimal son **óptimas**, es decir que siempre encuentran primero el mejor camino (el de menor longitud=profundidad= gasto o menor gasto respectivamente)

5) Comprobar que para un problema con soluciones, en la búsqueda en anchura, siempre hay en la lista *ABIERTOS* algún nodo de cada camino solución. ¿Ocurre lo mismo en la búsqueda en profundidad? ¿y en la optimal?

6) Comprobar que si $\boldsymbol{L_{i+1}=L_i+1}$ para cada $\boldsymbol{i}$ también la búsqueda en profundidad iterativa es óptima. En otro caso puede ser subóptima, es decir, puede encontrar una solución con un camino cuya longitud difiera de la del óptimo como máximo en $\boldsymbol{L_{i+1} - L_i}$ para los $\boldsymbol{L_i}$ y $\boldsymbol{L_{i+1}}$ correspondientes.

Ejercicios 2.1

Para los siguientes problemas interprete, abstraiga y formalice (es decir, describa formalmente en términos computables): (a) alguna formulación del espacio de estados; (b) descripción de los estados iniciales; (c) funciones u operadores de transición entre estados (o acciones de un agente que cambien el estado); (d) función que establece el coste del paso de unos estados a otros; (e) conjunto de estados meta o solución y criterio que permite valorarlos como tales. Cuando se pueda, presente varias formulaciones alternativas.

1.- Problema de las jarras: Dadas dos jarras de tres y de cuatro litros de capacidad, sin ninguna graduación, averiguar cómo llegar a tener en una de ellas exactamente dos litros sin usar medidas de capacidad auxiliares.

2.- Misioneros y caníbales: Tres misioneros acompañados de tres caníbales han de atravesar en su marcha un río usando una barca de remos de dos plazas.

3.- N-reinas: Colocar en un tablero como los de ajedrez de NxN casillas tantas reinas como quepan sin poder agredirse.

4.- N-puzzle: En un tablero cuadrado de N=mxm casillas hay N-1 fichas numeradas. Si éstas pueden moverse hacia una casilla contigua que esté vacía, obtener una configuración predeterminada a partir de otra inicial mediante los movimientos correspondientes.

5.- Coloreado de mapas: Dado un mapa (de visión de una zona o zonas cerradas del plano en subzonas distintas de área no nula), pintarlo con la menor cantidad de colores posibles.

6.- Construcción de crucigramas: Dados casilleros rectangulares, con segmentos dados de filas o columnas coloreados de negro, asignar palabras a los segmentos verticales u horizontales completos restantes de modo que se ocupen todos.

7.- Problemas criptoaritméticos: Dadas ecuaciones formadas por palabras (como HOCUS + POCUS = PRESTO) encontrar cifras que cambiadas por las letras originen operaciones aritméticas correctas.

8.- Laberintos: Dada una superficie divisible en casillas, con algunas de ellas señaladas como obstáculos (prohibidas), encontrar un camino constituido por casilla adyacentes desde una dada de ellas (entrada) hasta otra dada (salida).

9.- Granjero: Un granjero que transporta un conejo y una col al mercado en compañía de su perro ha de atravesar un río en una barca de remo en la que caben dos entes.

Ejercicios 2.2

1) Sea un problema formulado como de búsqueda en espacio de estados. Si el árbol de búsqueda tiene un factor de ramificación constante ***r*** y hay una sola solución a profundidad ***p,*** estúdiese para los distintos tipos de búsqueda cual sería el número de nodos a generar y analizar en el mejor de los casos posible, en el peor y en uno intermedio.
2) Calcúlese el tamaño de la lista de abiertos en la situación anterior, para los diversos métodos.
3) Compárese el coste espacial y temporal de la búsqueda en profundidad iterativa con los de las búsquedas en anchura y en profundidad (no es esencialmente peor)
4) Dar ejemplos de problemas en los que sea preferible la búsqueda en profundidad sobre la de anchura y al revés.
5) Un turista exótico trata de encontrar la dirección de un hotel, situado en una calle de una urbanización a lo largo de una carretera, en la que los inmuebles están indicados (rotulados) con numeración romana, cuyo sistema desconoce. Formular el problema como uno de búsqueda, examinar la pertinencia de los diversos métodos y resolver, en un caso concreto, mediante uno de ellos.
6) Resolver formalmente, mediante varios métodos de búsqueda en el espacio de estados el problema que, formulado tiene como estados E={0,1,2,3,4,5,6,7,8,9,a,b,c} como transiciones **T** = {(0,b), (b,0), (8,a), (5,a), (5,6), (8,6), (6,3), (3,6), (b,9), (5,4), (4,5), (7,4), (1,8), (3,b), (4,1), (4,7), (3,4), (6,5), (2,c), (c,2), (b,2), (b,3), (a,5) y (a,8)} , en con **a** como el estado inicial y donde las metas son **9** y **1**.

TEMA 3 BÚSQUEDA INFORMADA (O HEURÍSTICA)

1.- Funciones Heurísticas. Búsqueda informada.

2.- Métodos voraces.

2.- Algoritmo $\boldsymbol{A^*}$.

3.- Propiedades de las heurísticas: comparación de heurísticas, admisibilidad, consistencia, monotonía,

4.- Relajación y diseño de heurísticas.

5.- Búsqueda heurística con memoria limitada: algoritmos $\boldsymbol{IDA^*}$ y $\boldsymbol{SMA^*}$

6.- Mejora iterativa y búsqueda local.

BIBLIOGRAFÍA

-Nilsson, cap. 9

-Poole-Mackworth, secs. 3.6, 3.7.4,

-Russell-Norvic, secs. 4.1, 4.2

-Palma-Morales, *I.A. métodos técnicas y aplicaciones*, McGraw-Hill, cap. 9

-Galán-Boticario-Mira, *Problemas resueltos de I.A.,* Addison-Wesley, cap. 2

En los métodos de búsqueda no informada del tema anterior, la búsqueda se realizaba explorando sistemáticamente los nodos del grafo, a veces teniendo en cuenta lo ya gastado en la exploración de la parte del camino recorrido (coste uniforme), pero sin utilizar ninguna información disponible (que en ocasiones puede existir) sobre preferencia de unas vías sobre otras o sobre proximidad de las metas en las partes de los caminos aún por recorrer.

Cuando se dispone de alguna clase de información en este segundo sentido se habla de ***heurística***s y los métodos correspondientes son los ***métodos heurísticos.***

En ellos, básicamente lo que se hace es utilizar dicha información para elegir el desarrollo de unos estados frente a otros, tomando para desarrollar de la lista de abiertos antes aquellos que aparenten estar más cerca de la meta.

EJEMPLOS

En el problema del 8-*puzzle*: el número de piezas por colocar o la suma de las distancias de cada pieza descolocada a su posición definitiva son indicadores de la “distancia” de cada estado hasta la solución.

En problemas de laberintos o de mapas: distancias (aérea o en cuadrícula) desde cada estado a la meta.

FUNCIONES HEURÍSTICAS. BÚSQUEDA HEURÍSTICA

La información sobre proximidad a la meta consiste en una función numérica ***h*** del espacio de estados a ***[0, ∞ [= R⁺U {0}*** , que estime la "distancia" de cada estado al objetivo más próximo, valiendo **0** en las metas.

Esta estimación es información incierta (de no serlo, desaparecería el problema), y puede conducir por sí sola a engaños. Al usarla se sacrifica certeza para obtener economía, "podando" el árbol de búsqueda para centrarse en las ramas o direcciones más prometedoras.

Combinando adecuadamente la información heurística (sobre el futuro, incierta) con la del gasto realizado (sobre el pasado, conocida), se puede establecer métodos seguros y económicos, que combinan útilmente las ventajas respectivas de ambas informaciones

MÉTODOS VORACES

Búsqueda en *Escalada (o irrevocable, ingl. Hill Climbing)*

Esta búsqueda, rudimentaria, sin memoria, consiste en seguir siempre sólo el camino hacia el que, según lo apuntado por la función heurística, parezca mejor sucesor, mientras éste mejore al estado en que se esté:

1. Empezar con *ACTUAL* = lista formada por el estado inicial
2. Hasta que *ACTUAL* = meta o no haya cambios en *ACTUAL* , hacer:
   a) Presentar la lista de *ACTUAL* si su estado final es una meta y parar
   b) Tomar los sucesores de *ACTUAL* y usar *h* para puntuar cada uno de ellos.
   c) Si uno de los sucesores tiene mejor (o sea, menor valor de *h*) puntuación que *ACTUAL*, hacer *ACTUAL* igual a la lista anterior aumentada en el sucesor.
3. Presentar FALLO y terminar

CUESTIONES.

a) Comprobar mostrando ejemplos adecuados que esta búsqueda no explora más que parte de un único camino de los existentes, que puede no seguir dicho camino hasta el final, que puede entrar en ciclos infinitos, que no explora sistemáticamente el espacio de estados, y en suma, que no es completa ni óptima aunque la heurística sea totalmente fidedigna.
b) Estudiar la complejidad en tiempo y espacio de este "método"
c) Relacionar la primera deficiencia de (a) con la existencia de mínimos locales de la función *h* en el grafo de búsqueda y pensar posibles modificaciones del algoritmo para tratar de subsanarla.

Búsqueda *Primero el mejor*

Como la anterior, usa la heurística para elegir siempre al nodo sucesor aparentemente mejor situado, pero usa memoria (manteniendo abiertas las posibilidades alternativas mediante una lista de ABIERTOS) y así varios caminos posibles, no sólo uno sin alternativas.

Funciona de modo análogo al de la búsqueda (desinformada) optimal o de coste uniforme gestionando la lista ABIERTOS como una cola de prioridad, pero usando sólo la función heurística *h* para ordenarla y priorizar, examinando antes los nodos que tengan valor heurístico más bajo (que, según la información aportada por *h* estén "más cerca" del objetivo).

Por supuesto, la obtención de resultado y la calidad del mismo, dependerá de lo fidedigna que sea la función heurística.

1. Empezar con *ABIERTOS* = lista formada por el estado inicial
2. Mientras *ABIERTOS* no esté vacio hacer:

   2.1 Quitar de *ABIERTOS* la lista con el mejor nodo terminal y ponerla en *ACTUAL*

   2.2 Si el nodo terminal de *ACTUAL* es meta, presentar su lista y terminar.

   2.3 En otro caso, tomar los sucesores de dicho nodo, usar *h* para puntuar cada uno de ellos e incorporar las listas resultantes de ampliar *ACTUAL* con dichos nodos a la lista *ABIERTOS* en su orden.

3. Presentar FALLO y terminar.

Si se dispone de una buena función heurística, la búsqueda *primero el mejor* puede lograr en la práctica sustanciales descensos en el coste computacional de su ejecución, si bien no queda garantizado que el camino obtenido sea óptimo. Hay que tener además en cuenta el coste suplementario que la evaluación de *h* en cada nodo suponga.

CUESTIONES.

a) Comprobar mostrando ejemplos adecuados que esta búsqueda puede no ser óptima,

El ALGORITMO A^*

La idea en la que se basa la búsqueda **A*** consiste en intentar combinar las ventajas de la búsqueda desinformada de coste uniforme (completitud y optimalidad) con las de la búsqueda heurística primero el mejor (eficiencia, por la poda del árbol de búsqueda).

Ello se consigue priorizando en la lista de abiertos aquellos caminos en que menor resulte una **combinación** de los valores del **gasto hecho** en la parte del camino recorrido (suma de los costes de sus aristas etapa) con la **distancia** restante hasta la meta según la información dada por *h* en el trozo de camino por recorrer:

Para hacerlo, en el paso 2.3 del algoritmo anterior se substituye la función ***h*** por la ***f*** dada por

$$f(nodo)=g(nodo)+h(nodo)$$

lo que ordena a los nodos de *ABIERTOS* según la media aritmética de los valores de ***g*** y de ***h*** o por alguna otra combinación adecuada que agregue ambos valores.

Si la heurística es adecuada, con este método se consigue un algoritmo completo, óptimo y con un coste computacional menor, equivalente en la práctica una disminución del factor de ramificación

PROPIEDADES DE A^* Y DE LAS HEURÍSTICAS

Si se denotan con h^* a la heurística ideal, que daría la información perfecta sobre el coste óptimo, por el mejor camino, desde cada nodo hasta la meta más cercana, y con $f^*(n)=g(n)+h^*(n)$ al coste total de la mejor solución que pase por n, dada una posible función heurística ***h***, puede ocurrir:

a) Que $h(n)=0$ para todo ***n***, es decir, que la heurística no dé información. En este caso, A^* se convierte en la búsqueda optimal.
b) Que $h(n)=h^*(n)$ para cada ***n***, entonces no habría problema, pues en cada paso se sabría qué dirección hay que seguir para llegar a la meta más cercana del inicio por el mejor camino.
c) Que $h(n) > h^*(n)$ para algún ***n.*** Entonces A^* puede resultar no óptimo.
d) Que $h(n) \leq h^*(n)$ para cada ***n.*** Entonces A^* termina y resulta ser completo, óptimo y con coste computacional tanto más reducido cuanto mayor sea ***h*** (es decir, cuanto más parecida sea ***h*** a h^*), oscilando entre la búsqueda – a ciegas- de coste uniforme del caso (a) y la ausencia se búsqueda del caso (b).

Las heurísticas consideradas en (d) se llaman ***admisibles.***

Una heurística ***h*** es ***más informativa*** que otra ***h´*** si cumplen $0 \leq h´(n) \leq h(n) \leq h(n)^*$ para cada ***n***. Cuanto más informativa sea una heurística, más económica resultará la búsqueda mediante ella con A^*, es decir, menos nodos habrá que examinar para encontrar la solución.

Una heurística es ***consistente*** cuando para cada par de nodos adyacentes ***n*** y ***n´*** con el segundo sucesor del primero cumple que $h(n)-h(n´) \leq C(n,n´)$. Es ***monótona*** si se cumple que siempre que ***n´*** sea un sucesor de ***n*** se tendrá que $f(n´) \geq f(n)$. Cuando una heurística es consistente, entonces también es monótona y se cumple que la primera vez que un nodo sea escogido de la lista *ABIERTOS* para ser examinado, se habrá llegado desde el inicial hasta él por el camino más corto posible (es decir, que en la aplicación de ***A**** no habrá que “rectificar” ningún camino)

Nota: las pruebas de estas afirmaciones, véase Nilsson 9.2.1-9.2.4 o Poole-Mackworth 3

<u>CUESTIONES</u>

a) Mostrar con ejemplos la necesidad de las condiciones exigidas para garantizar las conclusiones correspondientes.
b) Describir funciones numéricas del espacio de estados que, usadas en vez de la función $f=g+h$ en A^* ocasionen las búsquedas en anchura, profundidad, de coste uniforme y primero el mejor.
c) Dadas dos heurísticas admisibles h_1 y h_2, describir para qué valores de los coeficientes α y β resultaría admisible la función $\alpha h_1 + \beta h_2$ como heurística.
d) Compruebe que toda heurística consistente es admisible, pero no toda admisible es consistente.
e) ¿Tiene sentido considerar búsquedas heurísticas bidireccionales?
f) ¿Puede plantearse algún tipo de búsqueda en profundidad iterativa heurística?