

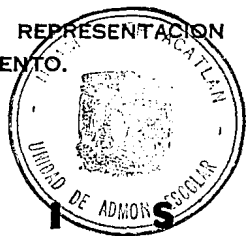
12  
2 eje



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES,  
ACATLAN

INTELIGENCIA ARTIFICIAL: REPRESENTACION  
DE CONOCIMIENTO.



T E S I S

QUE PARA OBTENER EL TITULO DE:  
LICENCIADO EN MATEMATICAS  
APLICADAS Y COMPUTACION  
P R E S E N T A ;  
VICTOR HUGO GARCIA BLANCO

ASESOR:  
M. EN C. SERGIO V. CHAPA VERGARA



ACATLAN, EDO. DE MEX.

1994.

TESIS CON  
FALLA DE ORIGEN



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL  
AVENIDA DE  
MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES "ACATLAN"

DIVISION DE MATEMATICAS E INGENIERIA  
PROGRAMA DE ACTUARIA Y MAC

SR. VICTOR HUGO GARCIA BLANCO  
Alumno de la carrera de Matemáticas  
Aplicadas y Computación,  
P r e s e n t e

De acuerdo a su solicitud presentada con fecha 16 de agosto de 1988, me complace notificarle que esta Jefatura tuvo a bien asignarle el siguiente tema de tesis: "INTELIGENCIA ARTIFICIAL: REPRESENTACION DE CONOCIMIENTO", el cual se desarrollará como sigue:

INTRODUCCION

- CAP. I.- Breve historia de la IA.
  - CAP. II.- Inteligencia Natural Vs. Inteligencia Artificial.
  - CAP. III. Teorías epistemológicas sobre la inteligencia.
  - CAP. IV.- Sentido común y razonamiento.
  - CAP. V.- Lógica.
  - CAP. VI.- Algoritmos. Versus. Heurística.
  - CAP. VII. Representación de conocimiento.
  - CAP. VIII Lenguajes de la inteligencia artificial
- CONCLUSIONES Y PERSPECTIVAS.

Asimismo fué designado como Asesor de Tesis el - M.en C. SERGIO V. CHAPA VERGARA, profesor de esta Escuela.

Ruego a usted tomar nota que en cumplimiento de lo especificado en la Ley de Profesiones, deberá presentar servicio social durante un tiempo mínimo de seis meses como requisito básico para sustentar examen profesional, así como de la disposición de la Coordinación de la Administración Escolar en el sentido de que se imprima en lugar visible de los ejemplares de la tesis el título del trabajo realizado. Esta comunicación deberá imprimirse en el interior de la tesis.

A T E N T A M E N T E  
"POR MI RAZA HABLARA EL ESPIRITU"  
Acatlán, Edo. Méx., diciembre 13 de 1993.

ACT. LAURA MARÍA RIVERA BECERRA  
Jefe del Programa

*A Mis Abuelas,  
¿ Como compaginar  
la aniquiladora  
idea de la muerte  
con este incontenible  
afán de vida ?.*

*A Marco, Elo y Bety,  
La historia tañe sonora  
su lección como campana  
para gozar el mañana  
huy que pelear el ahora.*

*A Alejandro,  
Contigo no es necesario  
decir las cosas, porque  
ya las sabes desde antes,  
desde siempre ...*

*A Martha,  
En recuerdo y homenaje  
a su amistad sin sombras,  
a su lección inolvidable.*

*A Paty,  
Si te quiero es porque sos  
mi amor mi cómplice y todo  
y en la calle codo a codo  
somos mucho más que dos.*

## INDICE

INTRODUCCION .....	1
CAPITULO I	
BREVE HISTORIA DE LA INTELIGENCIA ARTIFICIAL .....	4
CAPITULO II	
INTELIGENCIA NATURAL VERSUS INTELIGENCIA ARTIFICIAL .....	14
2.1: Inteligencia Natural .....	14
2.2: Inteligencia Artificial .....	16
CAPITULO III	
TEORIAS EPISTEMOLOGICAS DE LA INTELIGENCIA .....	20
3.1 Teoría de los Esquemas de Bergson .....	20
3.2 Teoría Piagetiana sobre los esquemas sensorimotores y Representativos .....	22
3.3 Teoría de Estructuras o Cuadros de Minsky .....	28
3.4 Aprendizaje de las Estructuras .....	30
3.5 Teoría de los Esquemas de Arbib .....	32
CAPITULO IV	
SENTIDO COMUN Y RAZONAMIENTO .....	35
4.1 Sentido Común .....	35
4.2 Razonamiento bajo Incertidumbre .....	46
4.2.1 Incertidumbre asociado a la probabilidad .....	47
4.2.2 Incertidumbre y Conjuntos Difusos .....	48
4.2.3 Incertidumbre e Inferencia Bayesiana .....	49
4.2.4 Incertidumbre en Medios de Creencia y Soporte .....	51
4.2.5 Conjuntos Difusos .....	52
CAPITULO V	
LOGICA .....	54
5.1 Lógica Proposicional .....	55
5.2 Lógica de Predicados .....	57
5.3 Principio de Resolución y Unificación de Robinson .....	62

CAPITULO VI

ALGORITMOS VERSUS HEURISTICA ..... 69

CAPITULO VII

REPRESENTACION DE CONOCIMIENTO ..... 74  
7.1 Representación con Cálculo de Predicados ..... 75  
7.2 Redes Semánticas ..... 80  
7.3 Reglas de Producción ..... 87

CAPITULO VIII

LENGUAJES DE LA INTELIGENCIA ARTIFICIAL ..... 89  
8.1 Lisp ..... 90  
8.2 Prolog ..... 99

CONCLUSIONES Y PERSPECTIVAS ..... 110

BIBLIOGRAFIA ..... 114

## INTRODUCCION

El objetivo de este trabajo es proporcionar un panorama claro y comprensible de los conceptos y técnicas de la Representación de Conocimiento, una de las áreas más relevantes de la Inteligencia Artificial (IA). La característica más fascinante y poderosa de los sistemas, desarrollada a través de la aplicación de dichas técnicas, consiste en la capacidad para enfrentar problemas que constituyen un reto del mundo real, por medio de la aplicación de procesos que reflejan el discernimiento y la intuición del hombre.

Actualmente las técnicas de la Inteligencia Artificial se están empleando en una amplia variedad de aplicaciones que comprenden, entre otras, el diagnóstico, la planeación, la predicción, el diseño, la interpretación, el control, el monitoreo de estado y la instrucción. En el futuro, a medida que se produzcan nuevas arquitecturas de equipos que sustenten de una manera más directa la ejecución de estos sistemas y que se perfeccione la tecnología, es razonable esperar un desarrollo de sistemas que se aproximen asintóticamente al comportamiento humano.

El desarrollo de tales sistemas nos permitirá no solamente ofrecer técnicas idóneas y potentes, sino también alimentar nuestro propio conocimiento sobre el proceso del pensamiento humano. Este es otro de los objetivos buscados por esta investigación y que se hace patente en el desarrollo metódico diacrónico del trabajo, si queremos saber cuales son actualmente las bases para hablar de Inteligencia Artificial, es necesario hacer un repaso de su historia. Tal es el objetivo del CAPITULO I, donde se presenta un resumen de algunas tendencias de este desarrollo.

¿Qué es exactamente la Inteligencia Artificial?. Aunque la mayoría de los intentos por definir los complejos y amplios términos anteriores: Inteligencia y Artificial, han fracasado, es útil esbozar mínimamente una aproximación al concepto para ofrecer un horizonte heurístico al presente trabajo. Dicho esbozo es el tema del CAPITULO II.

En el CAPITULO III se da una descripción acerca de las diversas teorías sobre el estudio de la inteligencia. Estas teorías pueden considerarse como los primeros intentos para correlacionar la función cerebral con los procesos inteligentes; al respecto, analizaremos la inteligencia desde el punto de vista filosófico, psicológico y de la Inteligencia Artificial.

Una característica de los sistemas desarrollados bajo el



marco de IA es tratar con hechos, proposiciones y reglas que son inciertas (es decir, que no sabemos con absoluta seguridad si son verdaderas). El objetivo del CAPITULO IV es conocer la manera de formalizar y representar el sentido común de manera que sea posible incorporarlo a una base de datos y manipularlo simbólicamente para optimizar búsquedas de soluciones; asimismo se introducen los más importantes métodos para el procesamiento de la incertidumbre, como son la incertidumbre asociada a la probabilidad, conjuntos borrosos y lógica difusa, incertidumbre e inferencia bayesiana, incertidumbre en medios de creencia y soporte.

El conocimiento representa la piedra angular de la potencia de estos sistemas; pero para darnos cuenta de este poderío, debemos ser capaces de representar abstractamente el conocimiento y utilizarlo para apoyar el proceso de razonamiento del sistema. Esta representación puede hacerse mediante el uso de la lógica matemática, en la cual encontramos dos ramas que han sido muy utilizadas en la Inteligencia Artificial: el cálculo proposicional y el cálculo de predicados. En el CAPITULO V, hablaremos de ambas ramas, así como el principio de resolución y unificación de Robinson, que es una técnica para realizar la inferencia en un sistema de lógica formal.

En el CAPITULO VI se hace una diferencia entre las dos formas de solucionar problemas, la algorítmica y la no-algorítmica. Tradicionalmente el manejo de algoritmos nos lleva a encontrar la mejor solución a un problema dado, sin embargo en la mayoría de los casos nos conformamos con una buena solución (por ejemplo, al buscar un lugar en el estacionamiento dejamos el carro en el primer lugar que encontramos, aunque podría existir un lugar mejor). De este modo se introduce la idea de técnica heurística.

Para resolver los complejos problemas que nos encontramos en el ámbito de la Inteligencia Artificial, se necesita a la vez de una gran cantidad de conocimiento y algunos mecanismos para manipular dicho conocimiento a fin de crear soluciones a problemas nuevos. Las técnicas de representación de conocimiento tratadas en el CAPITULO VII son mediante el cálculo de predicados, de redes y de reglas de producción.

Puesto que el fin último de la Inteligencia Artificial es la construcción de programas para resolver problemas, en el CAPITULO VIII se presenta una somera visión panorámica de los dos lenguajes más populares : el LISP y el PROLOG. Remarcando sus ventajas y desventajas.

Este trabajo contiene, a lo largo del mismo, buen número de referencias a los escritos de investigación sobre el tema. Estas referencias son importantes porque permiten proseguir con los temas individuales en una mayor profundidad, así como garantizar el acceso no solamente a un punto de vista único, sino a tantos como sea posible.

## CAPITULO I

### BREVE HISTORIA DE LA INTELIGENCIA ARTIFICIAL.

La creación de un ser inteligente por parte del hombre, ha sido y es uno de los sueños más antiguos de la humanidad. La leyenda y la historia están sembradas de oráculos, seres inmortales, autómatas, bustos y monstruos dotados de propiedades maravillosas para resolver toda clase de problemas. Las primeras manifestaciones del sueño de crear seres inteligentes, tienen lugar en la mitología griega cuando Homero (850 a.c.) describe los autómatas que ejecutaban trabajos que resultaban demasiado pesados para los dioses; posteriormente las encontramos en Fray Roger Bacon<sup>1</sup>, Ramon Lull (con su ARS MAGNA<sup>2</sup>), las leyendas sobre el golem<sup>3</sup>, el pato de Jacques de Vaucanson<sup>4</sup>, Leibnitz y Hegel, entre otros.

Mención aparte merece el británico Charles Babbage<sup>5</sup> (1792-1871) quien, asistido por la bella y matemática condesa Ada Lovelace (1815-1852), ideó unas máquinas de cálculo y análisis (máquina diferencial y máquina analítica) que, aunque no funcionaron con demasiada precisión debido a las deficiencias mecánicas de la época, son el primer recuerdo de alguna forma de inteligencia ajena a la mente humana. Lady Lovelace hizo la siguiente observación sobre la máquina analítica :

"El mecanismo operativo puede accionarse incluso independientemente de cualquier objeto sobre el cual operar (aunque naturalmente entonces no se obtendría ningún resultado). Por otra parte, podría actuar sobre otras entidades además de números, tales como objetos cuyas relaciones mutuas fundamentalmente pudieran ser expresadas mediante aquellas que la ciencia abstracta de las operaciones, y que asimismo pudieran ser susceptibles de adaptación a las acciones de la notación operativa y al mecanismo de la

---

<sup>1</sup> [HEPP, 87], pp. 25.

<sup>2</sup> [MONT, 86], pp. 9.

<sup>3</sup> [HEPP, 87], pp. 27.

<sup>4</sup> [HEPP, 87], pp. 32.

<sup>5</sup> [HYMA, 81]

máquina. Suponiendo, por ejemplo, que las relaciones fundamentales de los sonidos tonales de la ciencia de la armonía y de la composición musical fueran susceptibles de tal expresión y adaptaciones, la máquina podría componer piezas de música elaboradas y científicas de cualquier grado de complejidad y extensión."<sup>6</sup>

En 1937 el matemático Alan Mathison Turing publicó un artículo titulado "On Computable Numbers"<sup>7</sup>, en el cual afirmaba que si podía construir una máquina calculadora automática, capaz de llevar a cabo un conjunto simple de operaciones básicas con interruptores de dos posiciones, abierto o cerrado, la máquina podría efectuar cualquier cálculo matemático que pudiera completarse en un número finito de pasos. En un principio, el ensayo defiende la incapacidad de las máquinas para tales tareas para concluir con una serie de argumentos y conjeturas a favor de las posibilidades de construir máquinas inteligentes.

Dos años después, en 1939, estalló la guerra. Rápidamente la Gran Bretaña, con ayuda de agentes en Polonia, capturó una máquina codificadora Enigma, utilizada para transmitir mensajes de alto secreto. Si los británicos descifraban el código, podrían leer las comunicaciones alemanas y anticipar su estrategia. Con este fin y en el mayor secreto, se organizó un proyecto de criptoanálisis en Bletchey Park, al norte de Londres, bajo la dirección de Turing. Se construyó Colossus, una computadora de válvulas que leía los datos de una cinta perforada a razón de 5000 caracteres por segundo, y para febrero de 1940, era capaz de descifrar las comunicaciones nazis.

El Colosuss de Turing hizo avanzar los límites de su arte, pero sólo podía hacer una cosa : descifrar códigos de Enigma. La primera gran computadora electrónica se terminó de construir en febrero de 1946, en la Universidad de Pennsylvania bajo la dirección de J. Presper Ecker y John Mauchly, siendo bautizada con el nombre de ENIAC (Electronic Numerical Integrator And Computer).

Esta máquina, a pesar de grandes diferencias en sus bases técnicas, sólo carecía de dos rasgos fundamentales de las computadoras actuales : trabajaba basada en el sistema decimal y sólo podía programarse efectuando conexiones en cuadro, es decir, metiendo clavijas una por una en sus enchufes.

---

<sup>6</sup> [LOVE,61], pp. 248.

<sup>7</sup> [TURI,37]

En 1950, Turing publicó en la revista MIND un conocido trabajo, hoy uno de los clásicos de la Inteligencia Artificial: *Computing Machinery and Intelligence*<sup>8</sup>, en el cual proponía un test para discernir si una máquina es o no inteligente, conocido universalmente como "Test de Turing". El test se define como un juego en el que deben intervenir un hombre, una mujer y un interrogador; este último puede ser indistintamente de uno u otro sexo. El interrogador debe situarse de forma que quede aislado de los otros y el test consistía en que el interrogador debía averiguar quién es el hombre y quién la mujer, mediante una serie de preguntas realizadas a través de un teclado, sabiendo que tanto él como ella tratan de hacerse pasar por el hombre. Bien, ahora debe sustituirse a la mujer por una máquina "inteligente"; si ésta es capaz de ponerle difíciles las cosas al interrogador, entonces se podrá afirmar que es realmente inteligente, de otra manera no. De esta forma Turing presentó lo que entendía por una máquina inteligente, rehuendo una definición formal.

De las máquinas inteligentes fabricadas hasta hoy, ninguna de ellas ha sido capaz de dar un resultado esperanzador siguiendo el Test de Turing.

John Von Neumann<sup>9</sup>, matemático de Princeton, tuvo la idea de que la computadora debería almacenar su programa, utilizando el mismo código electrónico de los datos que manipulaba. Un programa de computadora en lugar de ser una secuencia fija de pasos, podía incluir un largo repertorio con todas las secuencias posibles, y sus operaciones reales podían no ser predecibles de antemano. La computadora podría llevar a cabo un cálculo, comparar el resultado con otro número y luego utilizar la comparación como una orden para escoger una nueva secuencia de pasos a ejecutar. Con estas ideas surge el EDVAC de Von Neumann, en 1947, que disponía tanto de aritmética binaria como de almacenamiento de programas en una memoria electrónica; y de esa forma se ha procedido desde entonces.

La reputación de Claude Shannon comienza cuando demostró que el álgebra de Boole podía utilizarse para diseñar circuitos eléctricos. Posteriormente desarrolló muchas ideas fundamentales de la teoría de las comunicaciones, el estudio formal de la transmisión por un canal de datos e informaciones. En 1950 escribió el artículo "Programming a computer for playing chess"<sup>10</sup>; en él decía que las computadoras podrían jugar un buen ajedrez (aunque no perfecto) estudiando un gran número de posibles jugadas y secuencias de jugadas, y luego escogiendo la mejor.

---

8 [TURI, 63]

9 [HEIN, 80]

10 [SHAN, 50]

Es obvio que las computadoras también pueden jugar damas, y este trabajo fue realizado por Arthur Samuel de IBM, en la década de los 50's. El programa de Samuel aprendía de las probables respuestas de sus contrincantes, como si fuera un jugador humano que planea una estrategia. Dicho programa podía tomar nota de sus anteriores partidas y por lo tanto, no sólo podía escoger la siguiente jugada mirando hacia adelante, sino recordando lo que había sucedido la última vez con aquella posición<sup>11</sup>.

Las ideas en este campo comenzaron a cristalizarse en el verano de 1956 cuando John McCarthy (entonces en Dartmouth), Claude Shannon y Marvin Minsky (en aquel entonces matemático de Harvard), mandaron una propuesta a la Fundación Rockefeller: "Proponemos que se lleve a cabo un estudio de dos meses con diez personas sobre Inteligencia Artificial...<sup>12</sup>". Esta fue la primera vez que se utilizó la expresión "Inteligencia Artificial", la cual fue propuesta por McCarthy. El estudio fue llevado a cabo en ese verano con la participación de Arthur Samuel, Nathaniel Rochester y varios más que asumirían la vanguardia en este nuevo campo.

Entre ellos se encontraban dos invitados de último momento, Alan Newell, de Rand Corporation y Hebert Simon, de Carnegie Tech (hoy la Universidad de Carnegie-Mellon). Y fueron precisamente ellos quienes aportaron el primer programa real, ayudados por Cliff Shaw, que podía considerarse razonablemente como una herramienta para que una computadora pensara. Lo llamaron el Teórico Lógico (Logic Theorist) y podía demostrar los teoremas de la Principia Mathematica<sup>13</sup>, obra central sobre los fundamentos de las matemáticas escrita por Bertrand Russell y Alfred North Whitehead. De hecho, el Teórico Lógico presentó una demostración más breve y satisfactoria para uno de los teoremas.<sup>14</sup>

Posteriormente, lograron coronar su éxito con el GPS (General Problem Solver), diseñado para aplicar ciertos procesos generales a la solución de problemas, incluyendo el análisis de medios y fines y la planificación. El análisis de medios y fines se basa en ver el lugar en donde uno está, compararlo con el lugar donde se desea ir, y buscar maneras para reducir esa diferencia. La planeación significa identificar varios objetivos a lo largo del camino, el seguimiento de los cuales quizá podría ser más directo y llevarnos más cerca del estado buscado. Como dijo Newell, este

---

<sup>11</sup> [SAMN, 63]

<sup>12</sup> [HEPP, 87], pp. 44.

<sup>13</sup> [WHIT, 50]

<sup>14</sup> [NEWE, 63]

programa tenía "pretensiones de generalidad"<sup>15</sup>. Funcionaba eficazmente con varias tareas, entre ellas problemas lógicos y rompecabezas, que encantan a los aficionados de los juegos matemáticos.

El GPS se basaba en la idea de que podía utilizar métodos normalizados para resolver un problema, sin disponer de muchos conocimientos especializados sobre el problema en sí. Pero este enfoque sólo podía llegar hasta este punto. Un enfoque distinto fue aportado por McCarthy, ahora en el Instituto de Tecnología de Massachusetts (MIT); él quería escribir un programa, llamado "Advice Taker", que mejoraría sus resultados si recibía información mientras trabajaba. McCarthy confiaba que su programa podría deducir las consecuencias lógicas de la información y combinarlas con lo que ya sabía. Este enfoque resultó aún más difícil de aplicar que el de Newell, Shannon y Shaw, pero por lo menos tuvo una consecuencia significativa: el concepto de tiempo compartido.

Uno de los campos con mayor actividad fue el de la comprensión del lenguaje natural, es decir, que una persona pueda hablar con la computadora en inglés ordinario. Si una computadora comprendiese el inglés, podría realizar un buen trabajo traduciendo idiomas. Al principio se pensó que únicamente se requería de un diccionario bilingüe y algunas reglas de gramática, sin embargo, este enfoque fracasó por diversos factores, como los distintos significados de una palabra, el idioma y la sintaxis.

Uno de los primeros programas lingüísticos fue Eliza<sup>16</sup>, creado por el profesor Joseph Weizenbaum, del MIT, en 1966. El programa era la parodia de un psicoterapeuta, quien demostraba su capacidad para alentar a un paciente a hablar. Eliza esquivaba la comprensión real del lenguaje recurriendo a un hábil sistema de respuestas fijas. Si alguien utilizaba la palabra "madre", Eliza contestaba con una frase típica como: "Dime más cosas sobre tu madre". Weizenbaum comparó el sistema con los intercambios educados dentro de una reunión, dónde uno puede sostener una conversación con frases como: "Si, claro" o "Cuéntame más sobre esto, por favor". A pesar de ello muchas personas pidieron a gritos conversar con el telétipo de Eliza y en ocasiones el resultado fue muy real:

Paciente : Todos los hombres son iguales.  
Computadora: ¿En qué sentido?  
Paciente : Siempre nos están explotando en una forma u otra.  
Computadora: ¿Su compañero la trajo aquí?  
Paciente : Dice que casi todo el tiempo estoy deprimida.  
Computadora: Siento mucho saber que usted está deprimida.

---

<sup>15</sup> [NEWELL, 63A]

<sup>16</sup> [WEIZ, 66]

Paciente : Es cierto, me siento desdichada.  
 Computadora: ¿Cree usted que haber venido aquí le ayudará a no sentirse desdichada?  
 Paciente : Necesito que se me ayude y eso es lo que creo.  
 Computadora: ¿De qué le serviría la ayuda?  
 Paciente : Quizás mejoraría las relaciones con mi madre.  
 Computadora: Hábleme de su familia ...<sup>17</sup>

Lo más notable es que Eliza fue rebautizada "Doctor" por los científicos en la década de los 60's. Otro programa, Parry, fue programado para estimular comentarios de los pacientes con el fin de formar estudiantes en medicina<sup>18</sup>, se tuvo la idea de lograr una conversación entre ambas computadoras, el "enfermo" reprobaba a su "terapeuta": "ya sería tiempo que demostrara más interés por mis emociones..."<sup>19</sup>.

Terry Winograd del MIT, en 1972, crea un programa de lenguaje natural, llamado SHRDLU, el cual mediante una interfase artificial de "bloques de palabras" logró integrar el lenguaje natural con un muy buen análisis semántico, sintáctico y con un grupo de palabras que conformaban un conocimiento.<sup>20</sup>

La Universidad de Carnegie Mellon diseñó a HEARSAY II<sup>21</sup>, programa que fue todo un éxito ya que podía trabajar con oraciones sobre un vocabulario de 1,000 palabras y un 90% de efectividad. Posteriormente un sistema compilado de arquitectura para redes de trabajo, HARPY, tenía una comprensión del 95% sobre el mismo vocabulario de HEARSAY II.

Desde ese momento no han cesado los intentos dentro de este campo, pero es hasta mediados de los setentas cuando un sistema experimental de los Laboratorios Bell, es capaz no sólo de escuchar a una persona, sino también de extraer información de sus palabras y luego elaborar una respuesta adecuada, codificada en inglés, y emitir la respuesta por medio de un sintetizador de voz. El sistema simulaba una agencia de viajes con un vocabulario de 14 palabras. Saltar de este nivel a la capacidad de leer a Sartre o Benedetti es un gran paso que aún no han dado las computadoras.

---

17 [FOGE,83], pp. 78.

18 [COLB,75]

19 [FOGE,83], pp. 79.

20 [WINO,72]  
 [WINO,73]

21 [ERMA,80]

Es irónico, que a pesar de que ha sido muy difícil codificar la comprensión del mundo como lo entiende nuestro sentido común, sí ha sido posible codificar una serie de conocimientos limitados y especializados. Los programas resultantes se conocen como sistemas expertos, los cuales intentan entrar a un mundo altamente técnico como es el de los especialistas. Imitar a este especialista ha sido mucho más fácil que hacer que la computadora comprenda el lenguaje cotidiano, quizás porque maneja problemas restringidos que no requieren conocimientos sobre el mundo<sup>22</sup>. Algunos de estos sistemas expertos son :

- \* DENDRAL, interpreta datos de instrumentos químicos y proporciona consejos sobre la estructura de compuestos desconocidos.<sup>23</sup>
- \* EURISKO, diseñado por Doug Lenat de la Universidad de Stanford, aprende matemáticas y descubre 250 teoremas distintos.<sup>24</sup>
- \* INTERNIST, trata problemas de medicina interna<sup>25</sup>. En una evaluación de 43 casos tomados del New England Journal of Medicine este programa formuló diagnósticos correctos en 25 casos, mientras que los médicos que trataban a los pacientes acertaron 28 veces.<sup>26</sup>
- \* MACSYMA, diseñado en 1961 por Slage del MIT, es un programa heurístico que lleva a cabo complejas manipulaciones simbólicas de cálculo y matemáticas superiores.<sup>27</sup>
- \* MYCIN, prescribe tratamientos para meningitis e infecciones de la sangre; diseñado en 1976 por Shortliffe de la Universidad de Stanford.<sup>28</sup>

---

22 (STEF, 82)

23 (LIND, 80)

24 (LENA, 83)

25 (POPL, 76)

26 (HEPP, 87), pp. 53.

27 (MACS, 77)

28 (SHOR, 76)



- \* MOLGEN, Ayuda a varios genetistas a cortar y analizar los tejidos de DNA que se encuentran en los genes.<sup>29</sup>
- \* PROSPECTOR, empleado en geología y en 1982, encuentra yacimientos de molibdeno en monte Toleman.<sup>30</sup>
- \* PUFF, diagnostica las enfermedades respiratorias en el Centro Médico del Pacífico, en San Francisco.<sup>31</sup>

Una de las áreas relevantes de la Inteligencia Artificial es la visión por computadora<sup>32</sup>, la cual consiste en hacer que la computadora "vea". Los primeros trabajos en este campo incluían el procesamiento de imágenes y los patrones de reconocimiento en dos dimensiones; posteriormente el reconocimiento de patrones se convirtió en un campo nuevo dentro de la ciencia de las computadoras.<sup>33</sup>

Uno de los pioneros en el campo de la visión por computadora es Roberts, quien en 1965 diseñó un programa que entendía las escenas de bloques geométricos. El programa encontraba los extremos de la figura mediante las derivadas espaciales de la intensidad de la imagen, y con los elementos resultantes trazaba una línea. Utilizando características simples, como el número de vértices, podía relacionar los objetos con la línea dibujada y almacenarla en tres dimensiones para verificar si el resultado estaba listo para ser reconocido.<sup>34</sup>

Gleason y Agin, de SRI (conocido entonces por Stanford Research Institute), desarrollaron el Módulo de Visión SRI, que es un sistema prototipo en los sistemas industriales de visión. El sistema, que utiliza una luz especial para producir una imagen binaria (silueta) de una pieza industrial, es capaz de extraer los extremos por medio de un scanner simple, y es la base de varios sistemas comerciales de visión.

---

29 (STEF, 81A)

30 (DUDA, 79)

31 (KULI, 80)

32 (WINS, 75)

33 (BINF, 87)  
(MARR, 82)

34 (ROBE, 77)

Otra área importante es la lógica computacional, en la cual la resolución, es decir un método automático que determine la solución de un problema, juega un papel básico. Green, en 1969, desarrolló un programa de preguntas-respuestas, llamado QA3, que podía resolver problemas simples como rompecabezas, movimientos de un robot o problemas de química<sup>35</sup>. El inconveniente de este tipo de trabajos es la explosión combinatoria, la cual consiste básicamente en que el número de pasos intermedios para encontrar la solución crecen en forma combinatoria.

Es claro que en las décadas de los 50's y 60's las técnicas de búsqueda tienen grandes dificultades para ser implementadas. En esta época gran parte del trabajo fue aprender la forma de hacer posible la Inteligencia Artificial, y se llegó a la conclusión de que la heurística era necesaria para evitar que las búsquedas crecieran en forma combinatoria.

En los 70's comienzan a capitalizarse las lecciones y surgen nuevas técnicas de representación de conocimiento, las viejas técnicas han madurado y son pulidas y corregidas. Es en esta década que se desarrollaron las herramientas básicas y las técnicas necesarias para la construcción de los sistemas expertos. También se creó, basado en el énfasis en el conocimiento, un subcampo dentro de la representación de conocimiento, que es conocido como "Ingeniería del Conocimiento" o "Arquitectura de la Mente".

Todo esto es la base, de la cual van a surgir los trabajos de la década de los 80's, y va a proliferar el desarrollo de sistemas expertos. Pero quizás lo más importante es el hecho de que la IA se vuelva comercial, es decir que algunas compañías han desarrollado prototipos comerciales basados en las primeras investigaciones sobre el campo. Han sido fabricados "chips" que tienen implementado el principio de resolución y unificación. Actualmente existen computadoras personales que trabajan con el lenguaje LISP y varios compiladores han sido puestos a la venta. Las compañías tienen a la venta programas de comprensión del lenguaje natural, algunos de aplicaciones sencillas sobre visión y mucha bibliografía sobre el tema.

En octubre de 1981 Tohru Moto-Oka dijo : " Hasta ahora, se ha seguido básicamente la dirección señalada por otros países en tecnología informática, pero ha llegado la hora de romper con esta tradición y centrar nuestros esfuerzos en el desarrollo de una nueva tecnología informática basada en nuestras propias concepciones "<sup>36</sup>, en el marco de la Conferencia de presentación del programa de la 5ta Generación de Computadoras. Este proyecto,

---

<sup>35</sup> (GREE, 69)

<sup>36</sup> (MOTO, 81)

diseñado para diez años y con un presupuesto de 1.5 billones de dólares, persigue los siguientes objetivos :

- \* Interfases inteligentes.
- \* Manejador de Bases de Datos.
- \* Resolvedor automático de problemas con capacidad de inferencia.

Todas estas capacidades serán desarrolladas con el uso de técnicas IA. Dichas máquinas no son concebidas como una máquina de Von Neumann, sino como proceso en paralelo y con una capacidad de un billón de inferencias lógicas por segundo.

Los japoneses prefieren usar el PROLOG para realizar sus aplicaciones y la arquitectura de sus máquinas. Usando este lenguaje para el desarrollo de sistemas que resuelven problemas (con guía heurística) el campo de resolución de problemas ha resurgido de una manera sorprendente.

Con el advenimiento de la Quinta Generación, los países europeos, como Francia y la Gran Bretaña, al igual que los Estados Unidos, están poniendo un renovado y especial interés en el campo de la IA.

Actualmente las herramientas de la IA están disponibles y las técnicas están lo suficientemente desarrolladas para algunas aplicaciones. La IA se ha convertido internacionalmente en un campo demasiado importante y por lo mismo se han destinado grandes sumas de dinero a su desarrollo.

Después de este esbozo histórico, entramos al problema de una posible definición de inteligencia, y las diferencias entre una inteligencia natural y una artificial.

## CAPITULO II

### INTELIGENCIA NATURAL VERSUS INTELIGENCIA ARTIFICIAL.

#### 2.1 Inteligencia Natural.

Hablar de la existencia de "máquinas inteligentes", o máquinas que disponen de una Inteligencia Artificial, equivale a abordar el problema fundamental de la definición de inteligencia. Para éste concepto fundamental no existe sólo una definición, hay varias que llegan a ser muy distintas entre sí; sin embargo, todas se muestran igualmente válidas en su contexto, intentaremos ofrecer varias aclaraciones, buscando una aproximación a lo que se podría englobar bajo el término de inteligencia. En un diccionario nos encontramos con la siguiente definición :

"Inteligencia (lat.). Facultad compleja para representaciones, que se consiguen a través de registros espontáneos de conexiones, en situaciones nuevas. 1. Podemos encontrarla en los animales evolucionados; en el hombre es, a lo largo de su vida, de suma importancia; y es un factor decisivo en el desarrollo de la cultura."<sup>37</sup>

El concepto anterior, maneja otras concepciones que juntas, permiten una nueva aclaración : "Facultad compleja", "registros espontáneos", "sucesión de la vida" y "desarrollo de la cultura". Esto no debe extrañarnos, casi todas las definiciones contienen conceptos importantes que deben ser desglosados hasta lo más simple.

Por otro lado, el psicólogo P. R. Hofstaeter menciona las cuatro circunstancias siguientes :

"Inteligencia. 1. Que se trata de una aptitud o grupo de aptitudes, que pueden existir en mayor o menor grado en un ser; 2. Que esta facultad hace posible la resolución de problemas concretos o abstractos y permite también el desarrollo de nuevas situaciones; 3. Que economiza las pruebas redundantes y el aprendizaje; 4. Que esta capacidad se manifiesta en la aplicación, registro,

---

<sup>37</sup> (FISC, 75), pp. 2843.

significación, fabricación de relaciones y dependencias.<sup>38</sup>

Hofstaeter va aún más lejos diciendo que la inteligencia en última instancia -en condiciones normales- se presenta como "éxito de la vida del individuo". En esa medida, la inteligencia permite "una conducta dirigida hacia un fin", y en ella se consigue el "desarrollo de la vida con éxito". De esta definición se desprende que el ámbito de la inteligencia se considera en el desarrollo de la vida con éxito, concepto que resulta sustantivo para alcanzar la definición de Inteligencia Artificial (IA).

Weizenbaum afirma que : " Un organismo se define a través de los problemas que él mismo puede superar. El hombre debe poder aclarar problemas cuando una máquina construida por él no puede hacerlo <sup>39</sup>. De aquí se desprende que el hombre, un ser que elabora información de manera comparable a una computadora, puede a diferencia de ella superar cualquier problema.

Se tienen dos corrientes que podemos considerar útiles para concebir el significado de inteligencia, del cual obtenemos el de IA. Buehler habla de : "la facultad de resolver nuevos planteos, de solucionar nuevos problemas"<sup>40</sup>, mientras que otros investigadores defienden la "facultad de aprendizaje " como característica de la inteligencia, esto es, la facultad de valorar la experiencia adquirida. Además, se tiene en la discusión la pregunta sobre si la inteligencia es una facultad general dentro de la utilización del rendimiento mental, o bien, si pertenece al conjunto de facultades especializadas. En resumen, existe tanto un factor general, como unas capacidades especializadas y consideraciones inteligentes, al lado de la facultad de resolver problemas.

Tal vez el rasgo más distintivo de inteligencia es la facultad de encontrar soluciones originales a una serie de problemas dados, aunque existen muchos hombres que tienen principalmente la facultad de "aprender".

De las dos corrientes de inteligencia que inciden sobre la facultad de resolver problemas y aprender se plantea dentro de la Inteligencia Artificial no sólo hablar de "máquinas inteligentes" o de "máquinas pensantes", sino también de "máquinas que aprenden por sí mismas".

Steinacker da la siguiente definición : " Inteligencia es la

---

38 [HOFS, 57], pp. 172.

39 [WEIZ, 78], pp. 268.

40 [BUEN, 62], pp. 99.

facultad de llegar a nuevas resoluciones a través de la utilización de medios de pensamiento "41. Esta definición se puede trasladar fácilmente al entorno de las computadoras y decir que los medios de pensamiento son la memoria de la máquina y los programas allí contenidos, el cerebro y el sistema nervioso del hombre.

## 2.2 Inteligencia Artificial.

Tanto el pensamiento científico como el popular se han visto dominados por una noción extremadamente simplista de la inteligencia, y esta noción es responsable, en parte, de que se desarrolle una fantasía equivocada acerca de la Inteligencia Artificial. Un organismo se define principalmente por los problemas que afronta, el hombre se enfrenta a problemas que son inasequibles para una máquina. El hombre no es una máquina, y aunque ciertamente procesa información, no lo hace necesariamente como una computadora. Las computadoras y los hombres no son especies del mismo género.

¿ Se puede concluir que las máquinas que se ocupan de las funciones mentales, y que hasta ahora estaban reservadas al hombre, son inteligentes ?. Y si en este sentido fueran inteligentes, ¿ bastan las computadoras para cumplir todas las funciones mentales necesarias ?.

Naturalmente no tiene sentido esperar esto, ya que únicamente las computadoras personales pueden realizar tareas "inteligentes", mientras que los mainframes se encuentran en el límite del problema.

Podemos hablar de computadoras "inteligentes" en tres ámbitos:

- \* El Aprendizaje.
- \* La Toma de decisiones.
- \* Las situaciones concretas (tales como juegos).

Pero no sólo deben de interesarnos los programas que actuen bajo estos ámbitos, sino deben ser de mayor interés los sistemas y máquinas que pueden realizar operaciones tales como :

- \* Reconocer lenguajes.
- \* Traducir lenguajes en forma automática.
- \* Dar informaciones precisas.

Turing concluye en su ensayo, "Máquinas Inteligentes",

---

41 (STE1, B4)

que las máquinas podían pensar y apoya esta afirmación en el ejemplo del niño lactante, cuyo dispositivo mental no está aún programado. Este niño tiene mucho que comprender y aprender, antes de tomar parte activa en el mundo. Por eso educamos, es decir programamos a las pequeñas máquinas pensantes durante años. Asimismo, Turing afirma que : " Si le sorprende el hecho de que el comportamiento de una máquina demuestra una cierta conducta, su comportamiento es también de hecho inteligente "42.

Popper da su particular opinión del tema : " Yo diría sin titubear que no es posible, con todos mis respetos a Turing, que opina lo contrario. Quizás podríamos hacer hablar a un chimpancé... quizás podríamos construir un microorganismo que dentro de un medio ambiente preparado pudiera reproducirse por sí mismo por encima... pero pienso que no es posible construir una computadora electrónica con capacidad de una vivencia subjetiva consciente. "43.

Al definir inteligencia como "la capacidad de conseguir nuevas representaciones a través del pensamiento", podemos trasladar perfectamente dicha definición a las máquinas, tanto a los robots como a las computadoras. Si tomamos esta afirmación como utilizable, las máquinas deberán estar capacitadas para :

- \* Percibir,
- \* Resolver problemas,
- \* Formular conclusiones,
- \* Jugar,
- \* Aprender,
- \* Utilizar conocimientos,
- \* Dar informaciones,
- \* Realizar actividades.

En esta pequeña enumeración se encuentra uno de los problemas más importantes de la investigación dentro del campo de la IA, ya que todo el trabajo científico se realiza con poca orientación : Surge la disciplina y como la alquimia de la edad media, se mezclan ingredientes distintos apenas con una base teórica, y se espera que aparezca algo como "inteligencia". El hecho de hacer "diabluras ciegamente" es de todas formas el comienzo de casi todas las nuevas disciplinas.

Es debido a este hecho que se reciben tantas críticas tanto en el sentido de la investigación, como también de forma general en el concepto de Inteligencia Artificial, del cual se deduce la idea de "máquinas pensantes".

---

42 [RITC, 84], pp. 128.

43 [POPP, 82], pp. 256.

Esta crítica se basa especialmente en que no todo lo que las máquinas están en situación de hacer puede ser comprendido, a causa que no se puede realizar un análisis completo de cada paso para saber si sigue los pasos del cerebro humano.

Weizenbaum comenta : " Acepto la idea de que un sistema de ordenador moderno, complejo y autónomo, se defina como organismo. Puede recibir a su medio ambiente de forma sensorial y también influirlo, pienso que se encuentra en un sentido extremo 'socializado', 'o sea, que se puede modificar a través de sus determinaciones del medio ambiente. Me distancio de la idea de construir un robot capaz de determinarse a sí mismo, en situaciones de aprender y que pueda distinguir entre sus propias partes y los demás objetos, los cuales han de poder autodefinirse por sí mismos, y llevar sus componentes a primera línea antes de llegar a un deterioro y enfrentarse a objetos que no pertenecen a su composición, que pueda construir un modelo de sí mismo, para ser tomado como un tipo de propia conciencia. Si digo que estoy preparado para tratar a este tipo de robot como a un 'organismo', puedo también tratarlo como un tipo de animal. Creo que no hay ninguna posibilidad de fijar un límite a los grados de inteligencia que este organismo pudiera llegar a conseguir"<sup>44</sup>.

" Siempre llegamos a la conclusión de que el ordenador actualmente o en el futuro pueda adquirir aún más inteligencia, llegando esta inteligencia a superar los problemas humanos y a ser extraña al deseo humano "<sup>45</sup>. Esta afirmación aclara que como mínimo en los campos donde la computadora resuelve problemas humanos, podrá dar ciertos resultados que permitan realizar y ahorrar una serie de actividades humanas.

De todas formas, esto no nos da la respuesta sobre si podemos tratar a esa máquina como inteligente. "Si nosotros creemos que el ordenador tiene inteligencia y poder, podrá tener ambas". <sup>46</sup>

Tomando en cuenta las apreciaciones anteriores, podemos concluir que :

Elaboración de IA es la disciplina que intenta conseguir un rendimiento pleno de la computadora; que cuando se utilice completamente, exigirá inteligencia.

Visto el concepto aproximativo de inteligencia, hace falta profundizar en las diversas concepciones del conocimiento que forman parte de nuestro siguiente apartado.

---

<sup>44</sup> (WEIZ, 78), pp. 278.

<sup>45</sup> op. cit. pp. 282.

<sup>46</sup> (WEIZ, 84), pp. 98.



## CAPITULO III

### TEORIAS EPISTEMOLOGICAS DE LA INTELIGENCIA.

Si bien es cierto que se conoce muy poco sobre la forma en que el cerebro da origen a los procesos mentales, han surgido diversas teorías filosóficas, psicológicas, en las ciencias relacionadas con la Inteligencia Artificial y en los estudios teóricos sobre el cerebro, acerca de la posible organización de la inteligencia. Estas teorías se pueden considerar los primeros pasos para correlacionar la función cerebral con los procesos inteligentes.

#### 3.1 Teoría de los esquemas de Bergson.

Bergson es un filósofo que basó su sistema en teorías y resultados biológicos (psicológicos) para aplicarlo en aspectos cognoscitivos en el ser humano. Bergson establece una teoría vital, dualista, de los aspectos cognoscitivos del ser humano, y su pensamiento constituye un análisis crítico y profundo sobre diversas teorías filosóficas y psicológicas relativas a la naturaleza, a través de su teoría de esquemas como mecanismos de intercambio entre lo que él denomina el cuerpo y el espíritu. Es por ello que esta teoría podría ayudar a comprender la compleja naturaleza de los procesos relacionados con la percepción, la memoria y la inteligencia, así como a estudiar cómo sus ideas han encontrado su equivalente experimental en la teoría piagetiana sobre la inteligencia, la teoría arbibeana de los esquemas y la de los cuadros de Marvin Minsky, que veremos posteriormente.

El punto de partida de la teoría de Bergson sobre la memoria, la percepción, la inteligencia y la imaginación, ésta en el cuerpo como centro de acción, dónde se establece la relación que las imágenes del universo pueden ejercer sobre él y, al mismo tiempo, cómo puede actuar sobre ellas. Debido a que la información sensorial pasa al cerebro y éste define la acción motora, Bergson identifica la función corporal con la cerebral; por ello establece que el papel del cerebro es el de definir ante el mundo material presente cuáles son las acciones virtuales o posibles del organismo sobre ellas o de éstas mismas sobre el cuerpo, formando, por tanto, una estructura básicamente sensoriomotora. Es por tanto un instrumento de análisis de las sensaciones recibidas y de selección respecto al movimiento a ejecutar. Para poder llenar adecuadamente esta función, el cerebro necesita percibir la situación presente y utilizar la información almacenada en la memoria para elegir cuál

de las posibilidades motoras es la más adecuada para esta circunstancia. Es en este punto donde Bergson establece su teoría de los esquemas y donde nos plantea cómo la relación entre la percepción y la memoria definen la conducta específica ante una situación dada.

Bergson propone que hay dos tipos de esquemas (él llama memoria al grupo de esquemas): los sensoriomotores, que son los hábitos generados por el sujeto ante una circunstancia dada, la cual desencadena la reacción apropiada (la estimulación simplemente desencadena los mecanismos motores que la utilizan); y los representativos que, por otro lado, surgen cuando la necesidad de realizar una acción es inhibida y se manifiesta por medio de imágenes.

Sin embargo, estos dos tipos de esquemas están interactuando continuamente ya que, al presentarse la estimulación sensorial, el esquema sensoriomotor crea una especie de vacío interior donde las imágenes de los esquemas representativos se van a manifestar, actualizar y relacionar, de tal modo que pueden definir cuál de las posibilidades es la más adecuada. Esta definición se fundamenta en que los esquemas representativos ofrecen a los mecanismos sensoriomotores todos los recuerdos capaces de guiar su actividad y de dirigir la reacción motriz en el sentido sugerido por la experiencia. Esta relación entre los esquemas sensoriomotores y los representativos es lo que da origen a la inteligencia.

Asimismo, en alguna forma, la actividad de la inteligencia conceptual no es sino la representación de lo que la inteligencia sensoriomotora ya había aprendido; así, la capacidad de generalización representativa surge de la sensoriomotora, pero ahora en el plano de la representación. Por ello, establece que el entendimiento no es más que la imitación del trabajo de la naturaleza, creando sistemas motores, esta vez artificiales, para hacerlos responder a una multitud de objetos individuales; el conjunto de tales mecanismos es lo que da origen al lenguaje.

A partir de la naturaleza sensoriomotora del cerebro y basándose en su teoría de esquemas, Bergson propone que la percepción de los objetos externos no es la extracción de sus características puras, sino que se fundamenta en la posible acción del cuerpo sobre ellas. Por esto, el universo es el conjunto de imágenes cuya percepción son esas representaciones referidas a las acciones posibles de una imagen particular, el cuerpo. La percepción, por lo tanto, no sólo es el resultado de excitaciones externas sino el trabajo de la memoria para definir, con base a la experiencia acumulada, cuál debe ser la acción más adecuada para confrontar dicha circunstancia.

Es en este punto donde Bergson establece una analogía

directa entre la percepción y la memoria, postulando que los procesos son en realidad inseparables, ya que no hay percepción sin memoria. Debido a que la primera tiene como función una posible acción usando a la memoria para ello, en lugar de ser eventos pasivos, la percepción y su interacción con la memoria en realidad son activos. La percepción, más que una actividad que proviene del objeto al sujeto, es una función circular donde la imagen percepción se dirige al cuerpo y la imagen recuerdo se plantea como hipótesis y sale al exterior, buscándose mutuamente una a otra por medio de ajustes internos donde la hipótesis trata de comprobarse en el objeto. Por este juego circular, postula que percibir es más bien adivinar por medio de un conjunto de esquemas motores que se van articulando al intentar seguir el movimiento de la realidad externa.

El punto de desacuerdo entre la teoría bergsoniana y las tesis científicas relativas a las actividades cognoscitivas reside en que Bergson postula que la memoria representativa no es otra cosa que el espíritu, mientras que la memoria sensoriomotora corresponde al cuerpo. Bergson identifica al espíritu con la duración subjetiva, indivisible y continua capaz de almacenar todos los acontecimientos y de representarlos activa y creativamente. De esta manera, la relación que se establece entre memoria representativa y memoria sensoriomotora no es otra que la relación entre espíritu y cuerpo.

Posteriormente veremos cómo la teoría de los esquemas de Bergson ha encontrado su contraparte experimental tanto en las teorías del desarrollo de la inteligencia de Piaget como en las de Arbib y Minsky.

### **3.2 Teoría Piagetiana sobre los esquemas sensoriomotores y representativos.**

Piaget integró la visión filosófica con la científica en sus estudios sobre inteligencia infantil, estudiando las diferentes etapas por las que atraviesa el niño hasta alcanzar su estado de madurez intelectual. Las técnicas que utiliza para ello son la observación de la conducta sensoriomotora, preguntas específicas y la conversación espontánea con el niño.

La teoría piagetiana establece que la inteligencia y la concepción del mundo surgen de la necesidad de adaptarse a su medio externo, es decir que la inteligencia es la capacidad de adaptarse al medio ambiente. Este acomodo es resultado de las operaciones que realiza un grupo de estructuras de información (esquemas), las cuales permiten definir para cada caso particular de las circunstancias externas, cuál es la acción a realizar, tanto en el exterior como en estas estructuras. Piaget concuerda en este punto

con la teoría de Bergson de los esquemas como sistemas sensoriomotores.

Los esquemas operativos tienen dos actividades fundamentales, la asimilación, que permite hacer uso del repertorio de conductas ante una circunstancia específica y modificar así el mundo externo de acuerdo con sus necesidades y su estructura interna, mientras que a las operaciones que le posibilitan transformar su organización interna con base en las circunstancias externas se le denomina acomodación. Estas dos operaciones combinadas permiten la adecuada adaptación al medio externo, ya que por medio de la asimilación el individuo se proyecta hacia el mundo, en tanto que el mundo se proyecta al individuo por medio de la acomodación. Sin embargo, ambas deben estar equilibradas, es decir debe haber una relación equitativa entre las dos funciones.

Piaget, al igual que Bergson, establece una diferencia entre la inteligencia sensoriomotora y la conceptual, pero las define en función de las operaciones de asimilación y acomodación. De esta forma renuncia a la idea Bergsoniana de que la representación y la memoria infinita son propiedades exclusivas del espíritu.

El esquema sensoriomotor es, pues, esa estructura de información que se asimila y acomoda al mundo presente, ya que éste exige una acción inmediata; mientras que el esquema conceptual se puede asimilar y acomodar tanto en la realidad presente como a espacios y tiempos subjetivos propios de la capacidad de evocación y representación de la inteligencia adulta. Es por ello que, aunque tanto en los esquemas sensoriomotores como en los conceptuales se hable de las mismas operaciones, la complejidad de éstas se va incrementando durante la transición de la inteligencia sensoriomotora a la inteligencia conceptual. Al respecto, Piaget establece como principales diferencias entre los dos tipos de inteligencia las siguientes :

- 1) La inteligencia sensoriomotora sólo enlaza percepciones y movimientos sucesivos, sin una representación de conjunto que domine los distintos estados en el tiempo de las acciones así organizadas, y que las refleje en un cuadro total y simultáneo.
- 2) El mismo tipo de inteligencia tiende al éxito y no a la verdad, encuentra su satisfacción en la llegada a la meta práctica perseguida y no en la comprobación o en la explicación.
- 3) Dicha inteligencia trabaja sobre la realidad misma y no en cuanto a los signos, símbolos, conceptos y esquemas representativos.

4) La inteligencia sensoriomotora es individual y no social debido a que no usa signos.

Asimismo, Piaget propone que para que la inteligencia sensoriomotora se desarrolle en inteligencia l3gica, se requiere de:

A) Un sistema de operaciones que transponga las acciones exteriores de sentido 3nico en actos mentales m3viles y reversibles, siendo la reversibilidad la que posibilita encontrar un estado anterior a los datos, no contradictorio con el actual y un estado tan real o realizable como este estado actual.

B) Una coordinaci3n individual interna de las operaciones que asegure tanto la reciprocidad de los puntos de vista como la correspondencia del detalle de las operaciones y sus resultados.

A partir de estos postulados, lo que para Bergson era la actividad del esp3ritu, Piaget lo afirma en funci3n de una serie de procesos y eventos que hace factible la representaci3n a partir de la actividad sensoriomotora. A pesar de esta radical diferencia, la relaci3n que Bergson establece entre la inteligencia sensoriomotora y la representativa es equivalente a la de Piaget, ya que la inteligencia representativa de 3ste, como la de aqu3l, tiene como funci3n poder aclarar y definir, tomando en cuenta, no s3lo el presente sino tambi3n el pasado, y el porvenir, la conducta m3s adecuada a realizar en la situaci3n actual.

Piaget establece la analog3a de las operaciones fundamentales del pensamiento (cuando la asimilaci3n y la acomodaci3n est3n equilibradas), con la definici3n matem3tica de grupo, a la que se asignan las siguientes propiedades :

#### I.- Combinatoria

$$x + x^1 = y$$

$$y + y^1 = z$$

donde cualquiera de los elementos del grupo pueden combinarse y dar origen a un nuevo elemento del mismo conjunto.

#### II.- Reversible

$$y - x = x^1$$

$$y - x' = x$$

para cada operación del grupo existe una operación inversa.

### III.- Asociativa

$$(x + x') + y' = x + (x' + y')$$

### IV.- Operador Identidad

$$x - x = 0$$

$$y - y = 0$$

una operación combinada con su inversa se anulan.

### V.- Tautología o Identidad Especial

$$x + x = x$$

$$y + y = y.$$

Con base en esta definición de grupo, Piaget propone las siguientes operaciones fundamentales del pensamiento :

#### 1) Grupos Lógicos :

- a) Jerarquías por la combinación de unas clases con otras.
- b) Asimetrías o seriaciones.
- c) Substitución.
- d) Equivalencia.

#### 2) Grupos Infralógicos, que dan origen a la noción de espacio y tiempo :

- a) Formación del todo por la unión de sus partes.
- b) Localización y desplazamiento cualitativo.
- c) Relaciones simétricas y sustituciones espacio-temporales.

- 3) Grupos de Valores, estableciendo la relación medios-fin.
- 4) Grupos Proposicionales, basados en implicaciones y contradicciones entre funciones, que es lo que constituye el lenguaje de las matemáticas.

De esta forma, Piaget postula que el fin último de la inteligencia del niño es el pensamiento propio de las matemáticas, las cuales se manifiestan tanto en la inteligencia sensoriomotora como en la conceptual.

Es importante hacer notar que los conceptos de grupo de Piaget son muy similares a las categorías kantianas, aunque en lo básico, Piaget sólo llama categorías a los conceptos de objeto, espacio, tiempo y causalidad.

Tomando en cuenta lo anterior, define a la inteligencia como el resultado del equilibrio entre la asimilación y la acomodación tanto en la inteligencia sensoriomotora como en la conceptual, lo cual permite al individuo realizar operaciones de distintos tipos con las mismas propiedades de un grupo matemático. Así, el número de relaciones que se pueden establecer tanto en el espacio como en el tiempo son innumerables.

La elaboración del universo con base en estas categorías se desarrolla en el individuo por la transición de un estado donde los objetos se centran en el individuo que cree que los dirige, aunque sin tener todavía conciencia de sí mismo, hacia otro donde el yo se ubica en un mundo estable e independiente de la actividad personal. Al principio, el universo consiste en imágenes de percepción, móviles y plásticas, que se centran en la actividad personal, para después alcanzar el estado en el que el cuerpo es sólo uno entre todos los objetos del universo y donde los cuerpos pueden actuar tanto entre ellos como con él, y éste sobre los otros.

Por lo tanto, los conceptos de espacio, tiempo, objeto y causalidad, se desarrollan simultáneamente con las funciones de la inteligencia, hasta alcanzar el estado donde ellos se identifican con las nociones científicas de los mismos. En este punto, Piaget propone que las categorías surgen como una integración entre el desarrollo de la inteligencia y el histórico de los conceptos científicos. Por ello, en última instancia, concuerda con Bergson en que los conceptos de espacio y tiempo son esquemas creados para interactuar con su medio ambiente.

Piaget sostiene que el móvil fundamental de la inteligencia es el equilibrio entre la acomodación y la asimilación. Este equilibrio se puede manifestar básicamente en tres campos : en la relación entre el sujeto y el objeto, en la analogía entre los esquemas del individuo al interactuar entre

ellos y en la creación de sistemas de representación que se organizan entre sí según una línea jerárquica específica.

Cualquier elemento que modifique la homeostasis en esos tres campos provoca un desequilibrio y, por lo tanto, causa un movimiento de recuperación de este equilibrio perdido. En tal sentido, Piaget postula que el individuo puede adoptar tres actitudes :

- 1) Simplemente puede ignorar el disturbio.
- 2) Modifica el esquema o los esquemas para poder afrontar de manera adecuada el desequilibrio creado.
- 3) No sólo modifica sus representaciones para enfrentar la situación presente, sino que es capaz de generalizarlos para situaciones similares.

En las dos últimas instancias, es claro que el nuevo esquema es más propio para confrontar las relaciones del sujeto con el objeto y para coordinar los esquemas; esto ha sido denominado por Piaget como reequilibración maximizadora. Asimismo, él considera que hay dos medios fundamentales para modificar el esquema :

- a) Realimentación negativa, específica de las diferencias entre lo que el esquema debe asimilar y lo que es capaz de absorber.
- b) Realimentación positiva, la cual activa un esquema que necesita adquirir nuevos elementos para poder afrontar una nueva situación, estimulando al objeto, a actuar con el fin de construir adecuadamente el nuevo esquema.

Por todo esto, la teoría piagetiana del aprendizaje se identifica con la evolución del conocimiento científico en dónde, debido a ciertos elementos que la estructura científica existente no puede interpretar, se crea una nueva interpretación que incluye a la anterior, pero mejorándola. Sin embargo, aunque Piaget habla de equilibrio entre la asimilación y la acomodación, tanto en la etapa sensoriomotora como en la conceptual, como de períodos críticos de construcción de esquemas, este estado de equilibrio está siendo desequilibrado continuamente por elementos que pueden ser asimilados por los esquemas presentes, postulando con ello un movimiento de desequilibrio y reequilibrio que no termina nunca.

A pesar de la riqueza conceptual de la teoría de Piaget todavía no es posible relacionarla prácticamente con el ser humano.



Por ello, no podrá ser evaluada en toda su riqueza hasta no tener las herramientas necesarias para transparentar la forma en que el cerebro realiza estas operaciones. En la actualidad hay dos campos que intentan estudiar, desde un punto de vista más práctico y objetivo, las diversas operaciones de la inteligencia; éstos son: la Inteligencia Artificial y los estudios teóricos sobre el cerebro. Un ejemplo del enfoque de la Inteligencia Artificial es la teoría de cuadros de Marvin Minsky. Los estudios teóricos sobre el cerebro tratan de investigar cómo podría crear el cerebro estas estructuras de información; un ejemplo de lo anterior lo encontraremos en la teoría de esquemas de Arbib.

### 3.3 Teoría de Estructuras o Cuadros de Minsky.

La teoría de cuadros de Minsky trata de detallar y proponer métodos mediante los cuales una teoría de la inteligencia pudiera ser implementada en una computadora. Su teoría se basa en los estudios sobre Inteligencia Artificial, en la teoría de esquemas de Bartlett y en los paradigmas de Kuhn. Sin embargo, aunque él no lo acepta, su teoría no es sino un intento por definir en forma más concreta, con el objetivo de ser implementada en una computadora, la teoría de los esquemas de Piaget.

Minsky propone que un cuadro es una estructura de datos que representa una situación estereotipada, la cual permite al individuo confrontar de manera adecuada la realidad. Por lo tanto, esta definición es equivalente al concepto de esquemas de Bergson, Piaget y Arbib. El cuadro tiene cierta información que define cómo debe de usarse, qué puede suceder después, o qué hacer si no ocurre lo esperado. Tal vez éste sea uno de los puntos más interesantes de su teoría, ya que plantea específicamente cómo podría ser instrumentado un sistema de expectativas y correcciones como parte del mismo cuadro, aunque el concepto de esquemas de anticipación ya había sido definido anteriormente.

Para Minsky una estructura es una red de nodos y relaciones. Las partes superiores de la estructura son fijas y representan hechos que siempre son reales, mientras que los elementos inferiores tienen muchas terminales que deben ser ocupadas dependiendo de la situación particular. Cada terminal puede definir las condiciones que se deben de cumplir para que algún elemento pueda formar parte de la estructura. A las terminales se les asigna generalmente un elemento usual que ayuda a completar una escena sin necesidad de ver todos los elementos de ella; empero estas terminales pueden cambiarse fácilmente en caso de que se requiera. Asimismo, es factible que las estructuras se unan entre sí para formar un sistema de estructuras; esta interacción puede realizarse mediante la construcción de esquemas de estructuras jerárquicas o por conducto de diferentes estructuras.

que pueden tener las mismas terminales, lo cual permitiría apreciar un evento desde varios puntos de vista. De la misma manera, tanto Piaget como Arbib sostienen que los esquemas interactúan entre sí y se pueden organizar en superesquemmas.

El sistema de estructuras está ligado a un sistema de recuperación de la información y a otro de identificación. El sistema de recuperación de información tiene como función buscar una nueva estructura en caso de que una estructura con sus terminales no se adapte a la situación presente.

Dicho sistema tiene las siguientes funciones :

- 1) Seleccionar y asignar subcuadros para detalles adicionales.
- 2) Encontrar cuadros que sustituyen a otros que no son adecuados para la situación presente.
- 3) Definir qué hacer si no se puede encontrar un cuadro que sea adecuado para la situación presente : modificar uno viejo o construir uno nuevo.

El sistema de identificación tiene como tarea asignar valores a cada terminal de cada estructura. Este proceso es controlado en parte por la información asociada a la estructura, la que incluye cómo actuar ante sorpresas, y por el conocimiento de las metas actuales del individuo. Los componentes del proceso de identificación son los siguientes:

- A) Enlistar metas incluidas en la estructura para decidir qué terminales y condiciones deben de ser usadas para adaptarse a la realidad.
- B) Obtener información para asignar elementos a las terminales que sean diferentes a las preestablecidas.
- C) Probar la utilidad de la estructura y, en caso necesario, cambiarla.

En este punto Minsky establece la analogía de estas operaciones sobre las estructuras con las operaciones piagetianas concretas o, más bien, con las operaciones entre los esquemas para actuar ante una situación dada.

El aplica su teoría de estructuras a problemas de visión y de lenguaje, incluyendo con ello no sólo la representación del lenguaje como tal, sino las relaciones de sujeto-predicado y pronombre-verbo, así como las de causas y efectos.

Asimismo, propone la jerarquía, que enseguida se presentará, de las estructuras para representar el lenguaje :

- \* Estructuras de sintáxis-verbos (convenciones proposicionales).
- \* Estructuras semánticas-acciones (metas, consecuencias).
- \* Estructuras temáticas-tópicos (retratos, etc).
- \* Estructuras narrativa-históricas (explicaciones y argumentos).

Quizás sea en este punto donde la teoría de cuadros supera a todas las anteriores, ya que define específicamente cómo podría representarse el lenguaje por medio de estructuras de información, así como procesos causales.

### 3.4 Aprendizaje de las Estructuras.

La teoría de las estructuras plantea la posibilidad de crear nuevas estructuras cuando una situación no puede confrontarse en forma apropiada; esta creación puede resultar de las modificación de una estructura anterior, o de la creación de una nueva. Así, el concepto de aprendizaje es equivalente a la teoría de aprendizaje de Piaget.

Minsky propone los siguientes mecanismos cuando un esquema no se adecúa a la situación actual :

- 1) Identificar las diferencias entre una estructura ideal y la presente.
- 2) Ignorar estas diferencias si no son fundamentales para la estructura.
- 3) Usar la información de la estructura para las posibles acciones a realizar.
- 4) Resumir porque esta estructura no sirve, para que posteriormente pueda ser excluida con mayor rapidez ante situaciones similares.

Asimismo, propone crear el espacio del problema que es una representación de las posibles soluciones que ayudan a resolver con más rapidez la situación presente, considerando que ésta es más función de la inteligencia, que el simple hecho de encontrar una solución. Estas ideas se basan en el intento de crear el GPS, como

en los sistemas de producción de Simpson. La idea básica de estos programas es que están constituidos por los que se ha denominado los demonios, centinelas que esperan una situación específica y realizan una acción cuando ello ocurre. Este concepto es el equivalente al esquema sensoriomotor de Piaget.

El control de la conducta, sin embargo, requiere de los siguientes procesos :

- a) Métodos para decidir qué procedimiento usar.
- b) Alimentar con la información necesaria el proceso.
- c) Que el proceso activo ofrezca una conclusión o resultado.

Conceptos que son equivalentes a una subrutina.

El sistema para solucionar problemas es definido por los siguientes elementos :

- I) Estado inicial.
- II) Estado final.
- III) Diferencia entre los estados inicial y final.
- IV) Operadores seleccionados con base en la diferencia entre el estado inicial y final.

Simpson propone que para realizar la solución específica se pueden utilizar sistemas de producción capaces de pasar de un estado de premisas a una acción específica; por lo tanto, el problema se reduce a cómo conducirse de un estado inicial a uno final y cuáles son los procesos necesarios para ello.

Al respecto Minsky propone que el aprendizaje es similar a la creación de nuevos paradigmas científicos de acuerdo con la teoría de Kuhn, donde se da una interpretación distinta de un evento dependiendo de los elementos que no concuerdan con la hipótesis anterior, creando un nuevo paradigma como resultado de la solución del espacio del problema. Sin embargo, Minsky no especifica cómo podría suceder realmente este tipo de fenómenos, en contraste con la teoría de Piaget.

Podemos resumir que la teoría de Minsky es un intento por especificar con más detalle una teoría de la inteligencia, no obstante, esta teoría todavía está a un nivel especulativo al igual que las analizadas anteriormente, y sus hipótesis aún deben ser corroboradas, no sólo siguiendo la lógica de la computadora, sino

tratando de estudiar si el individuo puede usar mecanismos similares. Marvin muchas veces interpreta la inteligencia siguiendo la lógica de la computadora y no la del cerebro, por lo cual varias de sus hipótesis sólo tienen una validez metafórica para la comprensión de cómo el cerebro realiza estas funciones.

### 3.5 Teoría de los Esquemas de Arbib.

Esta teoría trata acerca de las estructuras de información que regulan tanto la percepción como la conducta animal; es decir que es una combinación de: la teoría de Piaget, ideas surgidas en el campo de la Inteligencia Artificial, teoría de la computación y de los estudios teóricos del cerebro. Debido a esto Arbib define con mayor detalle el tipo de operaciones que el sistema nervioso debe realizar con el fin de estructurar la información que va a ser usada en el control de la actividad motora.

Arbib sostiene, al igual que Bergson y Piaget, que un esquema es un arreglo de programas, los cuales analizan un segmento de la entrada para determinar una posible acción, partiendo del hecho de que el cerebro es un centro de coordinación sensoriomotora. Por sus diversas funciones los esquemas de Arbib pueden ser relocalizables, sintonizables y pueden unirse a otros esquemas.

Dichos esquemas pueden realizar tres operaciones :

- 1) Rutinas de identificación de la entrada, cuyo objetivo es analizar una entrada multimodal y definir a qué esquema corresponde.
- 2) Rutinas de acción, después de que un esquema ha sido activado por las rutinas de identificación, se definen ciertas conductas que el animal ejercita hacia el objeto percibido. Por lo tanto, estas conductas motoras forman parte del esquema.
- 3) Rutinas de cooperación y competencia, éstas se encargan de definir cuál de los esquemas está más activo, eliminando de este modo la posibilidad de que un esquema menos activo pueda controlar la conducta del individuo.

Cabe mencionar que las dos primeras operaciones son equivalentes a la asimilación piagetiana.

Aquí Arbib propone que para poder definir las diferentes

conductas y su coordinación, se requiere de sistemas de control distribuido, controlados por mapas espaciales a través de la representación interna del universo, y no como copia de éste, sino como instrumento para poder guiar la acción. Y es en este punto donde coincide con el concepto de espacio de Bergson y Piaget.

Además, determina la representación interna del mundo como el grupo de esquemas que lo definen; el cual es representado como una escena (A,e) con una función :

$$e : T \rightarrow \pi Pa ; a \in A.$$

donde :

T = intervalo de tiempo.

Pa = poset o la aproximación a los parámetros a del esquema.

Esta escena es la representación interna de un episodio relativamente homogéneo sin cambios dramáticos, y se activa y extiende en el tiempo, incorporando el aspecto dinámico propio de los episodios de la vida normal del individuo.

Arbib considera que el control de la conducta motora se establece por medio de sistemas de control de adelanto, donde la causa de un desequilibrio activa por adelantado tanto a los sistemas correctores como a los de control de alimentación, donde la acción correctora se mide en función de la diferencia entre el valor real y el esperado. Ambos mecanismos pueden usar un sistema de identificación de parámetros que les permite adaptarse rápidamente a los diferentes cambios del exterior.

Tomando en cuenta que cualquier conducta implica una secuencia compleja de actos que envuelve la actividad coordinada de varios sistemas de control, Arbib propone que este control y coordinación de esquemas puede realizarse mediante un programa que él denomina de control de coordinación motora, esto es, una combinación de sistemas de control con conceptos de programas para computadora. Dicho programa puede representar una interacción de circuitos prefabricados entre sistemas neuronales que definen la secuencia de ciertas conductas motoras; para actitudes más complejas, este programa constituye una estructura distinta del sistema de control; sistema que puede controlar la activación de varios esquemas en el cerebro, posiblemente a través de activar repetitivamente varios subsistemas.

Estos conceptos son equivalentes a la relación jerárquica que Piaget establece entre esquemas rectores de la consecución de los medios para obtener un fin específico, aunque la analogía de

Arbib con un programa computacional esclarece y simplifica las funciones piagetianas de asimilación y acomodación.

Asimismo, Arbib plantea que el programa de control puede establecer adecuados planes o programas de control para obtener un fin específico, así como la capacidad generadora de nuevos programas basados en metas, instintos o situaciones externas, equivalentes a la acomodación de Piaget. Por ello, la combinación del mapa cognoscitivo con el programa de coordinación motora definen la coordinación motora del individuo ante una circunstancia dada.

La teoría de Arbib representa un avance hacia los posibles medios, a través de los cuales el cerebro puede coordinar la actividad sensoriomotora, y la naturaleza y operaciones de los esquemas necesarios para lograr que el individuo se adapte a su medio ambiente.

Esta parte plenamente teórica, aclara aspectos del conocimiento mediante varias posiciones; pero la investigación intenta poner esos aspectos naturales en esquemas de inteligencia artificial, por ello debemos abordar al sentido común, como lo haremos enseguida.

## CAPITULO IV

### SENTIDO COMUN Y RAZONAMIENTO.

#### 4.1 Sentido Común.

En la nueva generación de sistemas expertos, se pretende crear dispositivos inteligentes que realicen deducciones sobre un área específica de conocimiento, mediante la utilización de partículas claves de conocimiento que, sin obligarlos a realizar una búsqueda extensa y exhaustiva de soluciones posibles a un problema particular, los conduzca a una solución del problema tratado en un tiempo razonable, o más bien, a intuir una solución para que una vez que lo hayan realizado procedan, si así fueran requeridos, a comprobar que efectivamente lo es.

Un típico sistema experto actual hace búsquedas exhaustivas de soluciones, lo cual le obliga a considerar posibles espacios de búsqueda combinatoriamente explosivos. Para reducir el tamaño de tales búsquedas se hace necesario incorporar en él una heurística del área de conocimiento tratada, o en otras palabras, es necesario dotar al sistema experto de un procedimiento que lo guíe dentro del espacio de búsqueda.

Para un experto humano tal heurística es parte de su sentido común. De la experiencia que se tiene en un área se puede distinguir si un camino es viable o no, es decir, si un camino en el espacio de búsqueda conduce o no hacia una solución. Me permito en este punto citar a Henri Poincaré acerca de la invención matemática :

"... Además hemos visto que el trabajo matemático no es un simple trabajo mecánico, por lo cual no podríamos confiárselo a una máquina por más perfeccionada que se la supusiera. No se trata solamente de aplicar las reglas y de elaborar todas las combinaciones, de acuerdo con ciertas leyes fijas. Las combinaciones así obtenidas serían demasiado numerosas, inútiles y embarazosas. El verdadero trabajo del inventor consiste en elegir entre estas combinaciones, para eliminar las que son inútiles o, más bien, no tomarse el trabajo de considerarlas. Las reglas que deben guiar esta elección son tan finas y delicadas que es poco más o menos que imposible enunciarlas en un lenguaje preciso.... Podemos sorprendernos de ver invocar la sensibilidad con motivo de demostraciones matemáticas que aparentemente no



podrían interesar más que a la inteligencia... es un sentimiento estético que todos los matemáticos conocen... las combinaciones útiles son precisamente las más bellas, quiero decir, las que pueden excitar más a esa sensibilidad que todos los matemáticos conocen, pero que los profanos ignoran hasta el punto de sonreírse. ¿Qué sucede entonces? Entre las numerosas combinaciones que el subconsciente forma ciegamente, casi todas carecen de interés y de utilidad; pero, por eso mismo, no excitan la sensibilidad estética; la conciencia no las conocerá jamás. Solamente algunas son armoniosas y por consiguiente útiles y bellas a la vez, éstas serán capaces de conmover esta sensibilidad especial del matemático y, una vez excitada, llamará sobre ellas nuestra atención y les dará así la ocasión de volverse conscientes..."<sup>47</sup>.

Ciertamente se puede colocar a la belleza y a la estética como una heurística de las matemáticas. Quienes hemos trabajado con ellas hemos sentido más de vez el placer de encontrar la solución de un problema matemático. Esto por supuesto no es privativo de las matemáticas, las actividades que tradicionalmente se reconocen como artísticas tienen esta motivación. Los grandes maestros ajedrecistas poseen también un talento especial desarrollado por sus experiencias en el ajedrez para distinguir la existencia de caminos efectivos hacia un "dar jaque al rey" entre la abrumadora cantidad de posibles caminos que se le presentan en un juego.

Ahora bien, un experto humano basa cualquier diagnóstico que haga en una intuición similar, adquirida durante su experiencia en esa área del conocimiento, que la más de las veces queda tan intrínseca depositada en sus neuronas que él mismo puede ser incapaz de explicar las motivaciones que tuvo para formular un diagnóstico, si bien tras haberlo formulado puede explicarlo completamente. Estas intuiciones en un área específica de conocimiento distinguen al verdadero experto del que no lo es y constituyen el sentido común del experto.

El razonamiento cualitativo pretende formalizar el "sentido común" y representarlo de manera que sea posible incorporarlo a una base de datos y manipularlo simbólicamente para optimizar búsquedas de soluciones.

El razonamiento cualitativo se ha desarrollado como un mecanismo de deducción automática que trata de simular los procedimientos intuitivos que un experto humano realiza al analizar

---

<sup>47</sup> (POIN, 81)

un tema, objeto de su área de experiencia<sup>48</sup>. Si bien aún se encuentra en una etapa inicial y los métodos que se han presentado se trata todavía de generalizaciones difíciles, en diversos sistemas expertos se han implementado sus métodos. Este razonamiento tiene como sustentación teórica, o más bien, como teoría formal deductiva, la así llamada Física Intuitiva<sup>49</sup>, la cual, sin contraponerse con la Física Clásica, ni querer explicar fenómenos físicos de una manera alternativa, tiene como propósito modelar un fenómeno físico, por ejemplo, el funcionamiento de una bomba de agua, el sistema de enfriamiento de un horno nuclear, la posición relativa de satélites asincrónicos con órbitas en planos de diferente inclinación, etc., atendiendo únicamente a las magnitudes y a los cambios de magnitud de los parámetros que los determinan y las relaciones que aquéllos guardan entre sí. De esta manera, mediante tales relaciones logra establecer una descripción relativamente completa del fenómeno estudiado.

El estudio de un fenómeno físico consiste esencialmente en una aprehensión de los cambios que en él se efectúan. Los cambios quedan determinados por diversos procesos.

Clásicamente, los cambios de un fenómeno se modelan por ecuaciones diferenciales que relacionan a las variables involucradas en el fenómeno y a las variaciones que sufren unas respecto a otras. El estudio de los fenómenos tratando de resolver esas ecuaciones diferenciales puede ser muy difícil pues los modelos analíticos que resuelven tales ecuaciones sólo son eficientes para casos relativamente sencillos. Por otra parte, el resolverlas numéricamente también puede ser complicado; sin temor a exagerar, podríamos afirmar que en cualquier método numérico, el desarrollo en Serie de Taylor de la función que establece a la ecuación diferencial aparece implícito, por lo que en caso de tener varias variables en función de algunas otras el problema se complica sustancialmente. En este problema aparece todavía otro factor de dificultad dado por la estabilidad del problema. Son bien conocidos diferentes ejemplos en los que una ligera variación en los valores, aún contando con altos niveles de precisión y propagación despreciable de errores, ocasiona valores numéricos indefinidos o arbitrariamente grandes por caer en alguna singularidad del problema. Así pues, aunque algunas ecuaciones analíticas o diferenciales, pueden describir completamente un fenómeno, el modelo presentado puede resultar inoperable para hacer un diagnóstico de las posibles causas de una cierta configuración en el fenómeno estudiado o bien para predecir los futuros estados a partir de uno dado, suponiendo que se haya recibido una cadena de estímulos externos. Esta última actividad se conoce como predicción

---

<sup>48</sup> Para una exposición completa de su métodos ver [FORB,84]

<sup>49</sup> [KLEE,84]

en tanto que la primera se conoce, forzando un poco el lenguaje, como postdicción. La tarea de postdecir puede ser, en ocasiones, más difícil que la de predecir.

Usualmente, un experto humano no recurrirá a calcular los valores exactos de las ecuaciones que describen a un fenómeno, sino más bien ponderará unas series de cambios en las variables dependientes en función de los cambios que haya observado en las independientes, en otras palabras, verá cómo varían las salidas, del sistema que observa, al variar las entradas.

A continuación presento un ejemplo de la manera en la que se puede construir un modelo cualitativo de un fenómeno físico. Consideremos un gas dentro de un recipiente. La ecuación de estados de un gas ideal estipula que

$$pV = \mu RT \quad (1)$$

donde  $p$  es la presión de un gas sobre las paredes del recipiente que lo contiene,  $V$  es su volumen,  $\mu$  es la masa del gas en moles,  $T$  es su temperatura y  $R = 1.986 \text{ cal}/(\text{mol} \cdot ^\circ\text{K})$  es la constante universal de los gases.

Suponiendo que no hay pérdida de masas, tenemos que (1) da para cada una de las otras variables :

$$p = \mu R ( T / V ) \quad (2)$$

$$V = \mu R ( T / p ) \quad (3)$$

$$T = ( 1 / \mu R ) V p \quad (4)$$

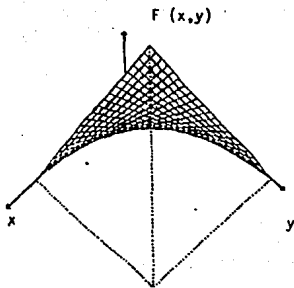


Figura 1  
 $F(x,y) = k \cdot x \cdot y$

Así pues conociendo los valores de dos variables podemos conocer los de la tercera.

Aún en este ejemplo sencillo, si bien es fácilmente calculable una triada de valores ( $p$ ,  $V$ ,  $T$ ) que satisfacen la ecuación de estados de un gas ideal, resulta un tanto complejo visualizar las propiedades variacionales de las variables involucradas, en función unas de las otras. Las ecuaciones (2) y (3) indican que bajo una temperatura constante el volumen del gas es inversamente proporcional a su presión, en tanto que, bajo un volumen constante, la presión es proporcional a la temperatura y, finalmente, bajo una presión constante, el volumen es proporcional a la temperatura. La ecuación (4) indica que la temperatura es proporcional al producto del volumen por la presión. Tales relaciones variacionales pueden visualizarse con las gráficas de las figuras 1 y 2.

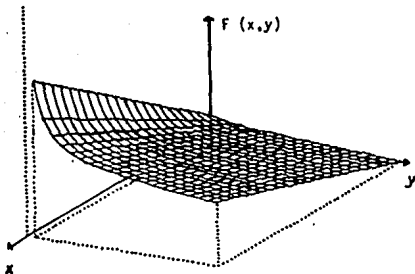


Figura 2  
 $F(x,y) = k \cdot x/y$

Analíticamente, los cambios en las variables los dan sus derivadas. Supondremos que la masa del gas se mantiene constante y que la temperatura, la presión y el volumen cambian respecto al tiempo. Al derivar la ecuación de estados de un gas ideal respecto al tiempo, obtenemos la ecuación :

$$\mu R \delta T / \delta t = p \delta V / \delta t + V \delta p / \delta t \quad (5).$$

Tendremos entonces una descripción completa de los estados del gas con las ecuaciones (1) y (5), las cuales involucran a seis variables : la temperatura, la presión y el volumen así como sus respectivas derivadas.

Si convenimos en que  $\delta T / \delta t$ , o sea, el cambio de temperatura, está en función de  $\delta V / \delta t$  y  $\delta p / \delta t$ , los cambios en el volumen y la presión respectivamente, entonces podemos distinguir tres clases de valores para ellos: los estrictamente positivos, los estrictamente negativos y los nulos. Los primeros nos indican que la variable a cuyo cambio se refieren se incrementa, los segundos a que esa variable se decrementa y los terceros, a que permanece constante.

Ahora bien, la ecuación (5) especifica la manera en que se relacionan las derivadas. En ella, el producto  $\mu R$  es estrictamente positivo. Por tanto, si nos atenemos únicamente a los signos de las variables y convenimos en denotar al signo de la variable  $x$  por  $[x]$ , la ecuación (5) se transforma en :

$$[\delta T / \delta t] = [p \delta V / \delta t + V \delta p / \delta t] \quad (6).$$

Se presenta naturalmente la siguiente pregunta : ¿ Bajo cuáles condiciones el signo de una suma queda determinado por los signos de los sumandos ?.

Resulta claro que, a menos que los signos de los sumandos sean opuestos, el signo de la suma será el del sumando que no sea nulo. Cuando los signos sean opuestos entonces el signo de la suma queda determinado por el sumando cuya magnitud, en valor absoluto, es mayor. En este caso, el conocer únicamente los signos de los sumandos no basta para determinar al de la suma. Así pues, para la operación suma, la propagación de signos queda determinada por la siguiente tabla; donde  $\&$  significa que la propagación de signos queda indefinida y

$[x] = -1$	$\longleftrightarrow$	$x < 0$
$[x] = 0$	$\longleftrightarrow$	$x = 0$
$[x] = 1$	$\longleftrightarrow$	$x > 0$

[x]	[y]	-1	0	1
-1	-1	-1	&	
0	-1	0	1	
1	&	1	1	

Para el producto, la propagación de signos se hace según la conocida ley : "El producto es positivo si los factores son de igual signo, negativo si son de signo opuesto y nulo si algún factor lo es.". Por tanto, para el producto, la propagación de los signos queda determinada por la tabla siguiente:

[x]	[y]	-1	0	1
-1	1	0	-1	
0	0	0	0	
1	-1	0	1	

Por tanto, la ecuación (6) es equivalente a

$$[\delta T / \delta t] = [p] [\delta V / \delta t] + [V] [\delta p / \delta t] \quad (7)$$

Esta última ecuación se dice ser la confluencia asociada a la ecuación diferencial (5).

De la manera en que se propagan los signos, puede verse

que los valores, para las variables involucradas que la satisfacen, son los que se resumen en la tabla siguiente :

$\delta[V]$	$[V]$	$\delta[p]$	$[p]$	$\delta[T]$
0	X	0	Y	0
0	0	X	Y	0
0	X	Y	0	0
X	0	0	Y	0
X	Y	0	0	0
0	E	G.E	X	G
E	0	X	G.E	G
E	X	0	G.E	G
X	E	G.E	0	G
E	F	G.F	G.E	G

donde

$$X, Y \in \{-1, 0, 1\} \text{ y } E, F, G \in \{-1, 1\}.$$

Así pues, tendremos  $5 * 3^2 + 4 * 3 * 2^2 + 2^3 = 101$  asignaciones de valores a las variables involucradas que satisfacen la confluencia.

El estudio que con este enfoque se hace de un gas en un recipiente procura hacer una descripción completa de las variaciones que sufre este fenómeno ante estímulos externos. En este caso podemos distinguir como estados del sistema estudiado las combinaciones posibles que se obtienen cuando las variables involucradas toman los diversos valores que hemos convenido en asignarles y que satisfacen la confluencia (7). En nuestro ejemplo distinguiríamos entonces a 101 posibles estados, los cuales son precisamente los enlistados en la tabla anterior, sin embargo este modelo resulta aún innecesariamente grande. En la práctica, la temperatura, la presión y el volumen sólo pueden tomar valores estrictamente positivos (los valores nulos o negativos son irrealizables en nuestro mundo físico según lo estipulan las leyes de la Termodinámica). Podemos pues establecer que el volumen, la presión y la temperatura sean únicamente positivos, lo cual es una suposición sumamente natural. Entonces nuestro conjunto de estados se reduce a  $1 + 2 + 2 + 2 = 7$  estados, dados por las configuraciones 1<sup>a</sup>, 6<sup>a</sup>, 8<sup>a</sup> y 10<sup>a</sup> de la tabla anterior. Los 7 estados resultantes corresponden a los valores de las variables que se muestran en la tabla siguiente.

A estos 7 estados hay que agregar dos más,  $Indef_1$  e  $Indef_2$ , de indefinición, los que corresponden a cuando  $\delta[V]$  y  $\delta[p]$  tienen signos opuestos.

Por otro lado, hemos convenido en considerar a los cambios en la presión y en el volumen como nuestras variables dependientes, por lo que a los estímulos externos del fenómeno los podemos formalizar como literales de un alfabeto.

	$\delta[V]$	$[V]$	$\delta[p]$	$[p]$	$\delta[T]$
0	0	1	0	1	0
1	0	1	-1	1	-1
2	0	1	1	1	1
3	-1	1	0	1	-1
4	1	1	0	1	1
5	-1	1	-1	1	-1
6	1	1	1	1	1

El alfabeto, bajo las condiciones especificadas, se restringe a 5 literales:  $u_{-1}$ ,  $u_0$ ,  $u_1$ ,  $o_{-1}$ ,  $o_0$ ,  $o_1$ ; las cuales dan valores a  $\delta[V]$  y a  $\delta[p]$ . Sus acciones se definen naturalmente:

- $u_{-1}$  : "Decrementa el volumen",
- $u_0$  : "Mantiene el volumen constante",
- $u_1$  : "Incrementa el volumen",
- $o_{-1}$  : "Decrementa la presión",
- $o_0$  : "Mantiene la presión constante",
- $o_1$  : "Incrementa la presión".

La especificación formal de este fenómeno queda como una estructura de transiciones que consiste de los estados cuando se recibe un estímulo externo. Nos referimos a los estados de esta estructura por la enumeración que hicimos de ellos en la tabla anterior.

Las transiciones en esta estructura están dadas por la tabla siguiente. Los estados 1, 3 y 5 corresponden a que la temperatura se decrementa, los 2, 4 y 6 corresponden a que la



temperatura se incrementa, el estado 0 corresponde a ninguna variación relevante de la temperatura e  $\text{Indef}_1$  y  $\text{Indef}_2$  a un desconocimiento en la variación de la temperatura, en este modelo.

En este ejemplo hemos considerado un fenómeno físico y lo hemos modelado atendiendo únicamente a los cambios de las variables involucradas, representadas éstas por los signos de las derivadas de esas variables. De esta manera, hemos incorporado, un cierto conocimiento intuitivo de una estructura de transiciones, es decir, como un dispositivo formal que consiste de un conjunto de estados y de unas transiciones entre ellos de acuerdo con estímulos provenientes de un alfabeto. Tales estructuras se conocen como semiautomatas finitos<sup>50</sup>.

	$u_{-1}$	$u_0$	$u_1$	$o_{-1}$	$o_0$	$o_1$
0	3	0	4	1	0	2
1	5	1	$\text{Indef}_2$	1	0	2
2	$\text{Indef}_1$	2	6	1	0	2
3	3	0	4	5	3	$\text{Indef}_1$
4	3	0	4	$\text{Indef}_2$	4	6
5	5	1	$\text{Indef}_1$	5	3	$\text{Indef}_2$
6	$\text{Indef}_1$	2	6	$\text{Indef}_2$	4	6
$\text{Indef}_1$	$\text{Indef}_1$	2	6	5	3	$\text{Indef}_1$
$\text{Indef}_2$	5	1	$\text{Indef}_2$	$\text{Indef}_1$	4	6

De esta manera, se elimina la necesidad de hacer evaluaciones cuantitativas de las variables que aparecen en el fenómeno estudiado y se restringe el estudio de la evolución de ese fenómeno a una mera evaluación en una tabla de transiciones. Diseñado ya el semiautomata, tales evaluaciones pueden realizarse de manera prácticamente instantánea. Más aún, si nos interesara hacer un diagnóstico de las posibles causas de una configuración del fenómeno, por ejemplo, si se nos preguntara porqué desciende la temperatura en nuestro modelo, entonces podríamos distinguir a los

<sup>50</sup> Para ampliar el tema de los automatas finitos consúltese [GINZ,67].

estados correspondientes a esa configuración como unos estados finales.

Un semiautomata con un conjunto de estados finales se dice ser un autómata. Así que un modelo cualitativo puede establecerse como un autómata. Un autómata distingue a un conjunto de palabras que, al aplicarse al autómata secuencialmente, lo hacen llegar a un estado final. Una tal palabra se dice ser reconocida por el autómata. Consecuentemente, las palabras reconocidas por el autómata que modela cualitativamente un fenómeno físico nos darán todas las posibilidades que pudieran ocasionar la configuración que nos haya interesado en el fenómeno estudiado.

Los autómatas finitos han tenido un gran desarrollo en esta segunda mitad del siglo. Es bien sabido que los lenguajes que ellos reconocen son precisamente los lenguajes regulares y consecuentemente por las gramáticas llamadas regulares, las que son unas especializaciones de las libres de contexto y que se encuentran en el nivel más bajo de la jerarquía de Chomsky<sup>51</sup>.

Las gramáticas libres de contexto tienen unas reglas sintácticas que pueden especificarse en una Forma Normal de Backus, la cual nos permite escribirlas en cláusulas típicas para ser manipuladas por un sistema de deducción automática.

Así pues, el conocimiento cualitativo de un fenómeno físico que se haya modelado por un autómata finito puede igualmente escribirse como cláusulas.

Por otro lado, si bien únicamente procedimientos regulares pueden ser modelados por autómatas finitos, la clase de lenguajes regulares es tan amplia que mecanismos de funcionamiento complejo pueden ser modelados por ellos. Sin embargo, el número de estados de un autómata puede crecer rápidamente cuando se profundiza en la especificación del fenómeno. En nuestro modelo, hemos visto que con sólo 5 variables, podríamos vernos en la necesidad de considerar un autómata de 101 estados. Aunque esto significa una objeción, a primera vista importante, a esta metodología, la estructura de un autómata es lo suficientemente sencilla para representar su propio funcionamiento por medio de un número menor de reglas generales de la manera en que transita de un estado a otro. Esta es la idea que se encuentra tras los llamados autómatas extendidos<sup>52</sup>, que utilizan a los autómatas finitos para la especificación de protocolos en comunicaciones.

En la modelación que hemos hecho, partimos de un modelo

---

<sup>51</sup> (DENN, 78)

<sup>52</sup> (CHOI, 85)

analítico, continuo, del fenómeno estudiado, después lo discretizamos seleccionando un conjunto finito de valores posibles y posteriormente sustituimos cada una de las variables del modelo continuo por variables discretizadas. En nuestro modelo, esto se ve como el tomar corchetes en cada variable. Diremos que el tomar corchetes es una operación que selecciona valores representativos.

Hasta ahora hemos considerado tres posibles valores para las variables representativas:  $-1$ ,  $0$  y  $1$  que respectivamente indican que los valores representados son negativos, nulos o positivos. Para algunos fenómenos puede ser suficiente considerar únicamente dos posibles valores: positivos y negativos, o bien nulos y no nulos. En algunos casos es posible que sea necesario considerar más de tres valores, por ejemplo, si se habla de circuitos eléctricos entonces el estado de que la entrada a un componente esté prendida o no, depende del voltaje de esa entrada, por lo que podríamos distinguir cinco posibles valores: completamente apagado, apagado pero con cierto ruido, nivel de ambigüedad, prendido pero con cierto ruido y completamente prendido; o bien si se hablase de dinero, se deberán ajustar las cantidades a la unidad monetaria más pequeña, centavos, en el caso mexicano. Sea como fuere, es necesario definir en ellos la manera en la que se componen por las operaciones usuales de la aritmética; en nuestro modelo, la composición de valores estuvo dada por las "reglas de los signos".

Es interesante remarcar que es a partir de un modelo continuo que se construye un modelo discretizado formado por las confluencias resultantes del modelo continuo para, posteriormente, ver al conjunto de arreglos de valores que satisfacen a las confluencias como estados de un autómeta finito sobre el alfabeto que da valores a las variables independientes del fenómeno estudiado.

#### 4.2 Razonamiento Bajo Incertidumbre.

Una de las capacidades más importantes de un experto y que ha presentado mayores dificultades para implementarse en un sistema, es la habilidad para enfrentar efectivamente la información imprecisa, incompleta y algunas veces sin certidumbre.

Existen muchas clases diferentes de incertidumbre que son comunes en dominios de expertos:

- \* **Conocimiento Incierto.** Con frecuencia solamente se tiene un conocimiento heurístico de algunos aspectos del dominio.
- \* **Datos Inciertos.** Aún cuando tengamos la certidumbre del

conocimiento del dominio, podría haber incertidumbre en los datos que describe el ambiente externo.

\* **Información Incompleta.** A menudo se hace necesario tomar decisiones basadas en información incompleta.

\* **Azar.** Algunos dominios son esencialmente azarosos: aunque el conocimiento disponible y la información este completa y además el conocimiento sea cierto, el dominio todavía tiene propiedades estocásticas.

La tecnología actual no tiene la capacidad para tratar la incertidumbre en forma efectiva como lo hace su contraparte, el hombre, y este tema, permanece como tópico importante de investigación. A continuación presentaré varios métodos para manejar incertidumbre; pero, tal como se anotó anteriormente, aunque estos métodos no duplican completamente las capacidades humanas, cada uno de ellos ha demostrado su utilidad en el desarrollo de sistemas reales.

#### 4.2.1 Incertidumbre Asociado a la Probabilidad.

Las técnicas de Teoría de Probabilidades se han empleado ampliamente en varias disciplinas diferentes en un intento de cuantificar la incertidumbre. El atractivo de la probabilidad está dado por el hecho de que se ha establecido sobre una base matemática sólida; las técnicas para emplear las probabilidades han sido ampliamente difundidas.

$P(E)$  es la probabilidad de que ocurra el evento  $E$ ; representa una cuantificación de la posibilidad de esta ocurrencia. En la mayoría de los casos el valor de  $P(E)$  se determina por medio de cálculos estadísticos.

Existen varios problemas que hacen difícil usar probabilidades para tratar la incertidumbre. Por ejemplo, aunque una persona sea experta en un dominio, puede serle muy difícil estimar exactamente las probabilidades.

Dado que las acciones de un sistema experto típicamente resultan de juntar varios fragmentos diferentes de conocimiento, cada una con diferentes características de probabilidad se pueden cambiar mediante el uso de técnicas bien establecidas. Por ejemplo, la fórmula

$$P(E_1 \text{ y } E_2) = P(E_1) * P(E_2)$$

puede emplearse para hallar la probabilidad de que tanto  $E_1$  y  $E_2$

ocurrirán, dadas las probabilidades individuales de  $E_1$  y  $E_2$ , cuando son eventos independientes. Es decir, que si la probabilidad de sacar un as de espadas de un mazo de cartas es  $1/52$ , y la probabilidad de sacar un diamante es  $13/52$ , entonces la probabilidad de sacar tanto un as y un diamante es  $1/52 * 13/52$ .

La teoría de probabilidades se ha aplicado exitosamente en varios Sistemas Expertos, como por ejemplo en PROSPECTOR<sup>53</sup>.

#### 4.2.2 Incertidumbre y Conjuntos Borrosos.

El Razonamiento Difuso se diseñó específicamente para tratar la inexactitud (o difusividad) que está presente en el conocimiento empleado por los expertos.<sup>54</sup>

En matemáticas convencionales, un conjunto determina claramente límites que identifican un grupo de elementos exacto, aunque potencialmente infinito. Dada una comprensión correcta de la definición de conjunto, es posible determinar si un candidato es o no miembro del conjunto. Desafortunadamente es muy difícil, si no imposible, desarrollar exactamente definiciones de conjunto para muchos conceptos y mecanismos de clasificación que el hombre emplea.

Rutinaria y subconscientemente los hombres colocan cosas en clases cuyo significado y significancia están bien comprendidos pero cuyos límites no están definidos.

En razonamiento difuso, el concepto de conjunto borroso corresponde a tales clases, un conjunto borroso es una clase de elementos con límites débilmente definidos. Como ejemplo de conjuntos borrosos se menciona el conjunto de "programadores hábiles", "mujeres lindas" y "gente divertida". Para identificar los miembros de un conjunto borroso, asociamos un grado de membresía a cada uno de los elementos que potencialmente pudieran ser miembros, el grado de membresía es un número, entre 0 y 1 que indica el punto hasta el cual un elemento es miembro del conjunto. Un grado de 1 indica que el candidato es definitivamente miembro, mientras que un grado de 0 indica que el candidato definitivamente no lo es. La transición entre estos extremos es gradual mas que una diferenciación.

Los grados de membresía para conjuntos borrosos se asignan en forma subjetiva de acuerdo con el contexto. En general

---

<sup>53</sup> (DUDA, 80)

<sup>54</sup> (ZADE, 65)

un conjunto borroso se define por una función de membresía que asocia un grado de membresía con cada uno de los elementos candidatos.

En algunos casos el grado de membresía se puede representar en forma inexacta como un número difuso (por ejemplo, un grado de membresía que esté "cerca de 0.2" o alrededor de "0.5"). Un conjunto borroso que utiliza valores difusos en su función de membresía se denomina conjunto ultraborroso.

Conceptualmente, los grados de membresía no se entienden como correspondientes con las probabilidades. En vez de representar la probabilidad de que un elemento sea miembro de un conjunto dado - con base en el análisis probabilístico de los datos - un grado de membresía se entiende como la cuantificación de una comprensión intuitiva del punto hasta el cual un elemento dado es compatible con un concepto dado.

#### 4.2.3 Incertidumbre e Inferencia Bayesiana.

Hasta la fecha, la técnica más empleada en la solución de toma de decisiones bajo incertidumbre es el teorema de Bayes. El teorema de Bayes en este contexto se utiliza para propagar las probabilidades que generan las evidencias independientes que soportan una determinada hipótesis.

El problema de toma de decisiones bajo incertidumbre puede ser expresado asignando una medida de probabilidad de que se cumplan las hipótesis en vista de las evidencias relevantes. Si al conjunto de evidencias relevantes le denominamos E;  $h_i$  a la i-ésima hipótesis;  $P(h_i|E)$  es la probabilidad condicional de que se cumpla la hipótesis en vista de las evidencias E.

El teorema de Bayes nos permite calcular las  $P(h_i|E)$  por medio de la siguiente fórmula:

$$P(h_i|E) = \frac{P(h_i) P(E|h_i)}{\sum_{j=1, n} P(h_j) P(E|h_j)}$$

donde  $P(h_i)$  es la probabilidad a priori de que se presente la hipótesis  $h_i$ , sin consideración alguna de la evidencia.  $P(E|h_i)$  es la probabilidad inversa, o sea la probabilidad de que se presenten las evidencias, a sabiendas de que se cumple la hipótesis.

Existen varias maneras de aplicar el teorema de Bayes en

sistemas de clasificación. Por ejemplo, PROSPECTOR utiliza una técnica de propagación de probabilidades basado en lo siguiente : La formula de Bayes nos dice que

$$P(H|E) = \frac{P(H) P(E|H)}{P(E)}$$

igualmente para el complemento de la hipótesis :

$$P(\bar{H}|E) = \frac{P(\bar{H}) P(E|\bar{H})}{P(E)}$$

dividiendo se obtiene

$$\frac{P(H|E)}{P(\bar{H}|E)} = \frac{P(E|H) P(H)}{P(E|\bar{H}) P(\bar{H})}$$

Duda define ventajas a priori ("priori odds") de H como :

$$O(H) = \frac{P(H)}{P(\bar{H})}$$

y las ventajas a posteriori ("posteriori odds") como:

$$O(H|E) = \frac{P(H|E)}{P(\bar{H}|E)}$$

la razón de probabilidad como :

$$\lambda = \frac{P(E|H)}{P(E|\bar{H})}$$

y la fórmula de ventajas de probabilidad como :

$$O(H|E) = \lambda O(H)$$

Esta ecuación nos dice cómo actualizar o propagar las ventajas de H dada la observación de la evidencia E.

En un sistema experto como PROSPECTOR, se asume que la razón de probabilidad ( $\lambda$ ) la proporciona el experto humano como una medida de veracidad de cada regla.

$\lambda \gg 1$  representa que la evidencia E es suficiente para que se cumpla H. Esto es, que la observación de E transformaría una débil probabilidad a priori en una fuerte probabilidad a posteriori de H.

Una vez que se conocen las  $P(h_i|E)$ , se pueden considerar varios criterios. Por ejemplo la hipótesis más recomendable es la que tiene mayor probabilidad: bien, las hipótesis más recomendables son las de probabilidad más alta, cuya suma sea cercana a la unidad.

Aun más, se puede hacer uso de algunas consideraciones heurísticas o pragmáticas. Por ejemplo, en el caso de diagnósticos médicos se puede presentar una probabilidad baja de una enfermedad crítica y aunque la probabilidad es baja, amerita atención terapéutica, dado que la probabilidad de ocurrencia no es cero y la enfermedad puede ser mortal.

#### 4.2.4 Incertidumbre en Medios de Creencia y Soporte.

Al igual que a cada regla se le puede asociar una medida de probabilidad, también se le puede asociar un factor de certeza (certainty factor: CF). Este CF refleja el grado con el que se cree que la regla se va a cumplir.

Un CF es un número en el intervalo  $[-1, 1]$ , que refleja el grado de creencia de una hipótesis. Mientras mayor sea el CF, mayor es la certeza que se le va a dar a una hipótesis. Cuando el valor de CF es 1, se sabe con completa seguridad que la hipótesis es cierta.

Quando el CF es negativo, expresa una medida de creencia de que la hipótesis es falsa. Mientras menor sea el CF, hay una mayor seguridad de que la hipótesis sea falsa.

En caso de que el CF sea cero, no se puede asegurar nada ni a favor ni en contra de la hipótesis. Esto se puede dar en dos casos; uno a causa de información incompleta, y otro si la



evidencia que soporta la hipótesis está balanceada con la evidencia que sugiere que es falsa.

Dado que el CF debe reflejar tanto la evidencia que soporta la hipótesis como la evidencia que la refuta, se puede descomponer en dos partes: una medida de creencia (measure of belief: MB) y una medida de descreencia (measure of disbelief: MD). La MB de una hipótesis  $h$ , dada la evidencia  $e$ , es proporcional a la disminución de la descreencia en  $h$ , y puede pensarse en términos de probabilidades como:

$$\begin{aligned}
 MB[h,e] : \\
 &= 1 \\
 &= \frac{\max[P(h|e), P(h)] - P(h)}{\max[P(1,0) - P(h)]}
 \end{aligned}
 \left. \begin{array}{l} \text{Si } P(h) = 1 \\ \text{en caso contrario} \end{array} \right\}$$

De manera similar, MD es proporcional al decremento de la creencia de la  $h$  como resultado de  $e$

$$\begin{aligned}
 MD[h,e] : \\
 &= 1 \\
 &= \frac{\min[P(h|e), P(h)] - P(h)}{\min[P(1,0) - P(h)]}
 \end{aligned}
 \left. \begin{array}{l} \text{Si } P(h) = 0 \\ \text{en caso contrario} \end{array} \right\}$$

Un elemento concreto de evidencia, bien incrementa la probabilidad de  $h$ , en cuyo caso  $MB(h,e) > 0$  y  $MD(h,e) = 0$  (es decir, no existe ninguna razón para dejar de creer  $h$ ), o bien decrementa la probabilidad de  $h$ , en cuyo caso  $MD(h,e) > 0$  y  $MB(h,e) = 0$ . Estas relaciones pueden deducirse de las fórmulas anteriores para MB y MD.

A partir de esta dos medidas, podemos calcular el CF como:

$$CF[h,e] = MB[h,e] - MD[h,e].$$

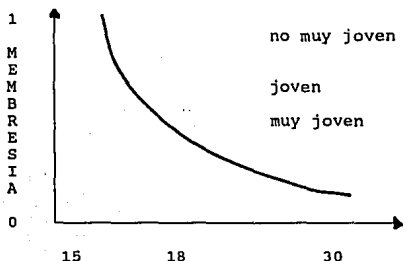
#### 4.2.5 Conjuntos Difusos.

La lógica difusa es un sistema de razonamiento bien

definido que se fundamenta en el empleo de los conjuntos difusos (o borrosos) en lugar de los valores binarios asociados con la lógica bivalente tradicional.

En un sistema de lógica difusa solamente los elementos que se están manipulando son difusos, las reglas de la lógica están bien determinadas.

La tarea de traducir las expresiones del hombre a sistemas de lógica difusa es relativamente simple dado que los hombres tienden a comunicar ideas y cuantificaciones verbalmente antes que descripciones numéricas. Estas expresiones se basan en el empleo de variables lingüísticas tales como grande, viejo y rápido. Para traducir estos valores a sistemas de lógica difusa, describimos simplemente una función de membresía que los representa. Una vez que se hayan establecido las funciones básicas, pueden usarse los modificadores difusos en la función de membresía. A continuación se ilustra el uso de los modificadores "muy" y "no"



JOVEN UNIVERSITARIO DE PRIMER SEMESTRE

Los cuantificadores difusos se emplean para representar una cuantificación aproximada en lógica difusa. Los cuantificadores difusos frecuentemente se emplean para disposiciones, declaraciones que incluyen una cuantificación difusa. Por ejemplo, la declaración, "los caballos tienen cola", realmente implica el uso del cuantificador difuso "la mayoría" (ejemplo, "la mayoría de los perros tienen cola").

Con frecuencia se hace necesario combinar los cuantificadores difusos cuando se construye una inferencia. También

es posible combinar los efectos de conjuntos difusos independientes.

En primer lugar debemos determinar el punto hasta el cual se satisfacen las precondiciones. Lo hacemos hallando el grado de membresía de cada una de las precondiciones mediante el uso de las funciones de membresía dadas.

Hemos presentado el intento por incorporar el sentido común con una base de datos y manejarlo simbólicamente a fin de optimizar la búsqueda de soluciones<sup>55</sup>. En este mismo sentido la lógica formalizada da su contribución, que veremos enseguida.

---

<sup>55</sup> Para mayor información sobre los aspectos humanos del razonamiento difuso, véase [BAND,83], [GUPT,82] y [KAND,86], para información adicional sobre hardware desarrollado para razonamiento difuso, consúltese [JOHN,85].

## CAPITULO V

### LOGICA.

Una forma particular de representar verdades de algún mundo relevante (hechos) es utilizando el lenguaje de la lógica formal, que es una consecuencia de las antiguas consideraciones filosóficas y que sugiere inmediatamente una potente forma de derivar nuevos conocimientos de los antiguos: la deducción matemática. En este marco, podemos concluir que un nuevo enunciado es cierto demostrando que se deduce de los enunciados ya conocidos. Así, se puede manejar la idea de demostración, tal y como se ha desarrollado en matemáticas como forma rigurosa de demostrar la verdad de una proposición que se cree cierta, para incluir la deducción como forma de derivar respuestas a cuestiones y soluciones a problemas.

Newell en teoría de números<sup>56</sup> y Gelernter en geometría<sup>57</sup> fueron pioneros en la exploración de técnicas de IA para la demostración mecánica de teoremas, área que aún esta activa en la investigación de IA<sup>58</sup>. Pero, la utilidad de algunas técnicas matemáticas se extiende mucho más allá del ámbito tradicional de éstas, pues resulta que las matemáticas no son diferentes de otros esfuerzos intelectuales complejos, ya que ambos requieren mecanismos de deducción fiables y un dominio del conocimiento heurístico para controlar lo que de otra forma sería un problema de búsqueda completamente inabordable.

Presentaremos, en primer lugar, el uso de la lógica proposicional como forma de representar la clase de conocimiento acerca del mundo que puede necesitar un sistema de IA; posteriormente exploraremos la lógica de predicados, que es el sistema lógico formal más ampliamente utilizado y terminaré el capítulo con el Principio de Resolución y Unificación, que es una técnica poderosa para realizar la inferencia y evitar la explosión combinatoria.

---

<sup>56</sup> (NEWELL, 63A)

<sup>57</sup> (GELE, 63)

<sup>58</sup> (BLED, 77)

## 5.1 LOGICA PROPOSICIONAL.

La lógica proposicional es atractiva porque es sencilla de manejar y existe para ella un procedimiento de decisión; se pueden representar hechos del mundo real mediante 'proposiciones' lógicas escritas como fórmulas bien formadas (FBF), tal y como se muestra a continuación :

Está lloviendo  
LLOVIENDO

Está soleado  
SOLEADO

Si está lloviendo, no está soleado  
LLOVIENDO ==> SOLEADO

Usando estas proposiciones podríamos deducir, por ejemplo, que si llueve no está soleado. Pero pronto encontraremos limitaciones en la lógica proposicional. Intentemos representar el hecho expuesto en la frase clásica : "Sócrates es un hombre", como:

SOCRATESHOMBRE

pero si queremos representar también que: "Platón es un hombre", se tendría que escribir como

PLATONHOMBRE

que sería una afirmación totalmente separada, y no seríamos capaces de sacar ninguna conclusión sobre las similitudes entre Sócrates y Platón. Sería mucho mejor representar estos hechos como:

HOMBRE(SOCRATES)

HOMBRE(PLATON)

puesto que ahora la estructura de la representación refleja la estructura del conocimiento mismo. Tenemos incluso más dificultades si intentamos representar la igualmente clásica frase:

Todos los hombres son mortales

porque ahora necesitamos realmente la cuantificación, a menos que dejemos escribir frases separadas acerca de la mortalidad de cada ser conocido.

Así, parece que estamos forzados a trasladarnos a la

lógica de predicados como forma de representación de conocimiento, puesto que ésta permite representaciones de entidades que no pueden representarse razonablemente en lógica proposicional. En lógica de predicados podemos representar los hechos del mundo real como declaraciones escritas como FBF. Pero la principal motivación para elegir alguna lógica era que si usáramos declaraciones lógicas como formas de representar conocimiento, dispondríamos entonces de una buena forma de razonar sobre ese conocimiento. La determinación de la validez de una proposición en lógica proposicional es directa, aunque puede ser difícil computacionalmente.

## 5.2 LOGICA DE PREDICADOS.

Antes de adoptar la lógica de predicados como un buen medio de representar conocimiento, deberíamos preguntarnos si proporciona también una buena forma de razonar sobre el conocimiento. A primera vista, la respuesta es sí proporciona una forma de deducir nuevas declaraciones a partir de las anteriores. Desafortunadamente, a diferencia de la lógica proposicional, no posee un procedimiento de decisión, ni siquiera uno exponencial. Si existen procedimientos que encontraran la demostración de un teorema propuesto, si es realmente un teorema; más no existe ninguna garantía de que el proceso se detenga si la declaración propuesta no es un teorema. Un procedimiento simple sería usar las reglas de inferencia para generar teoremas a partir de los axiomas de alguna forma ordenada, comprobando cada una para ver si es aquel para el cual estamos buscando una demostración. Sin embargo, este método no es particularmente eficiente, y quisiéramos encontrar uno mejor.

Exploremos a continuación la forma de representar conocimiento mediante la lógica de predicados; la cual consiste de cuatro componentes principales: un alfabeto, un lenguaje formal, un conjunto de enunciados básicos llamados axiomas (expresados en lenguaje formal) y un conjunto de reglas de inferencia. Cada axioma describe un fragmento de conocimiento y las reglas de inferencia se aplican a los axiomas para deducir nuevos enunciados verdaderos.

El alfabeto para un lenguaje formal consiste en los símbolos a partir de los cuales se construyen los enunciados. Nosotros utilizaremos para nuestro alfabeto :

\* **Constantes.** El componente más sencillo, se utiliza para representar un elemento específico del dominio. Este elemento puede ser cualquier objeto señalado de interés, incluyendo los objetos físicos y las abstracciones tales como ideas, puntos de vista o conjunto de datos. Se representa con una serie de letras mayúsculas.

\* **Variables.** Un símbolo variable se emplea para representar un conjunto de elementos del dominio sin especificar elemento alguno en concreto. Para un símbolo variable se emplea un conjunto de letras minúsculas.

\* **Funciones.** Además de los símbolos constantes y variables, se puede emplear una función para identificar un elemento del dominio. Una función describe un elemento identificándolo como el resultado único de la aplicación de una transformación entre otros elementos del dominio. Una letra minúscula se emplea como el símbolo de una función y los argumentos pueden ser cualquier término válido, donde un término es una forma de identificar un elemento del dominio. Un término se define inductivamente como una variable, una constante o una función.

\* **Predicados.** Los predicados se emplean para representar relaciones dentro del dominio e indican que un elemento se relaciona en alguna otra forma específica. Un predicado tiene el valor de verdadero si los elementos dados están relacionados de modo específico y de falso si no lo están.

\* **Interpretación.** Para que una fórmula sea útil, de alguna forma debemos de estar posibilitados para determinar exactamente lo que representa. Para componer una fórmula, el analista selecciona en primer lugar un conjunto de símbolos específicos y luego establece una correspondencia, relación entre cada uno de los símbolos seleccionados con el elemento del dominio asociado. Este conjunto de correspondencias establece la semántica que nos permite interpretar una fórmula.

\* **Conjunciones.** Para expresar proposiciones compuestas, empleamos conjunciones que combinan fórmulas para construir fórmulas bien definidas (FBD) más complejas. Las fórmulas componentes pueden ser FBD o Fórmulas atómicas. Las conjunciones de uso corriente son :

$\wedge$	and (Y)
$\vee$	or (O) inclusivo
$\implies$	implica
$\equiv$	equivalente a
$\neg$	negación.

\* **Cuantificación.** El cuantificador universal ( $\forall x$ ) se utiliza para aseverar que una fórmula es verdadera para todos los valores de la variable asociada y el cuantificador existencial ( $\exists x$ ) se emplea para aseverar que por lo menos existe alguna asignación para  $x$ , que hará que la fórmula asociada sea veraz. La variable

asociada con un cuantificador se llama variable cuantificada, y el alcance del cuantificador es la fórmula que sigue. Las variables que se referencian en los cuantificadores se llaman ligadas y aquellas que no lo están se llaman libres. La mayoría de las fórmulas de interés para los sistemas expertos se llaman enunciados, que son las fórmulas que contienen solamente variables ligadas.

\* **Lenguaje.** El lenguaje formal asociado a la lógica de predicados es el conjunto de todas las FBD.

\* **Evaluación.** Los conceptos de verdad y prueba formal son importantes en un sistema de lógica formal. El proceso de inferir nuevos conocimientos depende solamente de manipulaciones sintácticas y es un método para la demostración rigurosa de la verdad de un enunciado propuesto, basado en la verdad que ya se ha conocido. Este proceso, obviamente, depende de nuestra habilidad para determinar si una FBD cualquiera es verdadera o falsa.

\* **Delimitadores.** Utilizaremos paréntesis y comas.

Quando se utiliza lógica formal el proceso básico de ingeniería de conocimiento consiste en los siguientes pasos :

- 1) Desarrollar una comprensión del conocimiento.
- 2) Formular el conocimiento.
- 3) Separar los enunciados en sus partes constitutivas.
- 4) Escoger los símbolos para representar los elementos y relaciones en cada componente.
- 5) Construir FBD empleando los símbolos anteriores, que representen los enunciados.

Una vez que se seleccione los símbolos de lógica formal y la interpretación para cada uno de ellos, es importante que posteriormente debamos estar posibilitados para interpretar apropiadamente la lógica formal (para restablecer el significado original sobre la única base de las representaciones simbólicas formales e interpretaciones).

Como mencionamos anteriormente, la inferencia en lógica formal es el proceso de generar nuevas FBD a partir de las FBD existentes, mediante la aplicación de las reglas de inferencia. La regla más común de inferencia, modus ponens, dice " si  $P_1$  es verdadera y  $P_1$  siendo cierta implica que  $P_2$  sea verdadera, entonces



$P_2$  será verdadera", lo cual se expresa simbólicamente como:

$$[P_1 \wedge (P_1 \implies P_2)] \implies P_2.$$

Otra regla común de inferencia<sup>59</sup>, especialización universal, se usa para generar el FBD  $F(\text{INDIVIDUO})$  a partir de la FBD  $(\forall x) [F(x)]$ . Esta regla se puede establecer simbólicamente como:

$$\text{INDIVIDUO} \wedge (\forall x) [F(x)] \implies F(\text{INDIVIDUO})$$

La forma de inferencia por sustitución se basa en el reconocimiento de las leyes de equivalencia que son ciertas para todas las fórmulas. Ilustraré su uso para desarrollar una prueba de un hecho buscado.

Dados los siguientes hechos:

- 1) Cuando un avión se queda sin combustible los motores se detienen.
- 2) Si los motores de un avión se detienen, se requiere un aterrizaje forzoso.
- 3) En un momento dado, el avión del Capitán Trejo se queda sin combustible.

debemos demostrar que los motores del avión se detienen y que se requiere un aterrizaje forzoso cuando se agota el combustible. iniciemos definiendo 3 predicados:

- SC El avión del Capitán Trejo se quedó sin combustible.
- MD Los motores del avión del Capitán Trejo se detienen.
- AF El Capitán Trejo debe efectuar un aterrizaje forzoso.

Los primeros hechos que se dieron se pueden representar como  $SC \implies MD$  y  $MD \implies AF$ . En el momento en que el combustible del avión se agote, podemos agregar el hecho SC y expresar nuestro conocimiento colectivo como la conjunción :

$$(SC \implies MD) \wedge (MD \implies AF) \wedge SC$$

---

<sup>59</sup> KLEN, 83

ahora para poder aplicar la sustitución, utilizaremos las leyes de equivalencia que a continuación se enlistan :

(5.1)	$P_1 \implies P_2$	$\equiv$	$\neg P_1 \vee P_2$
(5.2)	$P_1 \vee P_2$	$\equiv$	$P_2 \vee P_1$
(5.3)	$P_1 \wedge P_2$	$\equiv$	$P_2 \wedge P_1$
(5.4)	$P_1 \vee (P_2 \wedge P_3)$	$\equiv$	$(P_1 \vee P_2) \wedge (P_1 \vee P_3)$
(5.5)	$P_1 \wedge (P_2 \vee P_3)$	$\equiv$	$(P_1 \wedge P_2) \vee (P_1 \wedge P_3)$
(5.6)	$P_1 \vee (P_2 \vee P_3)$	$\equiv$	$(P_1 \vee P_2) \vee P_3$
(5.7)	$P_1 \wedge (P_2 \wedge P_3)$	$\equiv$	$(P_1 \wedge P_2) \wedge P_3$
(5.8)	$P_1 \implies P_2$	$\equiv$	$\neg P_1 \implies \neg P_2$
(5.9)	$\neg(\neg P_1)$	$\equiv$	$P_1$
(5.10)	$\neg(P_1 \vee P_2)$	$\equiv$	$\neg P_1 \wedge \neg P_2$
(5.11)	$\neg(P_1 \wedge P_2)$	$\equiv$	$\neg P_1 \vee \neg P_2$
(5.12)	$P_1 \vee \text{FALSO}$	$\equiv$	$P_1$
(5.13)	$P_1 \vee \text{VERDADERO}$	$\equiv$	$\text{VERDADERO}$
(5.14)	$P_1 \wedge \text{VERDADERO}$	$\equiv$	$P_1$
(5.15)	$P_1 \wedge \text{FALSO}$	$\equiv$	$\text{FALSO}$
(5.16)	$P_1 \vee \neg P_1$	$\equiv$	$\text{VERDADERO}$
(5.17)	$P_1 \wedge \neg P_1$	$\equiv$	$\text{FALSO}$

dando como resultado :

TRANSFORMACION	JUSTIFICACION
$(SC \implies MD) \wedge (MD \implies AF) \wedge SC$	
$(\neg SC \vee MD) \wedge (\neg MD \vee AF) \wedge SC$	5.1
$SC \wedge (\neg SC \vee MD) \wedge (\neg MD \vee AF)$	5.3
$[(SC \wedge \neg SC) \vee (SC \wedge MD)] \wedge (\neg MD \vee AF)$	5.5
$[\text{FALSO} \vee (SC \wedge MD)] \wedge (\neg MD \vee AF)$	5.17
$(SC \wedge MD) \wedge (\neg MD \vee AF)$	5.12
$[(SC \wedge MD) \wedge \neg MD] \vee (SC \wedge MD) \wedge AF$	5.5
$(SC \wedge MD \wedge \neg MD) \vee (SC \wedge MD \wedge AF)$	5.7
$(SC \wedge \text{FALSO}) \vee (SC \wedge MD \wedge AF)$	5.17
$\text{FALSO} \vee (SC \wedge MD \wedge AF)$	5.15
$SC \wedge MD \wedge AF$	5.12

La última fórmula de la tabla,  $SC \wedge MD \wedge AF$ , representa la condición buscada. Alternativamente podríamos resolver este problema aplicando el modus ponens doblemente :

$$[SC \wedge (SC \implies MD)] \implies MD \quad \text{y} \quad [MD \wedge (MD \implies AF)] \implies AF$$

Una de las grandes ventajas de la representación, al emplear la lógica formal es que la inferencia sintáctica es posible y está garantizada en cuanto a validez. (Dado que las fórmulas de partida son válidas). Esta característica hace posible desarrollar sistemas de inferencia automática que son muy poderosos.

Desafortunadamente, aunque se garantice la verdad de todas las inferencias, en realidad no hay garantía que todas sean útiles. De hecho con frecuencia son completamente inútiles. El número de tales inferencias inútiles puede ser finito y aún más, el procesarlas directamente puede ser demorado. Observe, por ejemplo, el número de pasos que utilizamos para resolver un problema sencillo.

Afortunadamente, se han desarrollado varias técnicas para realizar la inferencia mediante el empleo de la lógica formal. A continuación veremos la unificación y la resolución como muestra de estas técnicas.

### 5.3 Principio de Resolución y Unificación de Robinson.

La resolución consiste en la fundamentación de muchos sistemas lógicos y también es la base computacional del Prolog. Para introducir los conceptos básicos de resolución, utilizaremos el siguiente ejemplo:

El detective Holmes y su ayudante Watson han sido llamados para investigar la sospechosa muerte del Sr. Dumas. Fuera de la casa de la víctima los vecinos le dicen a Holmes que la víctima, que era conocida como un solitario de pocos amigos, regresó a casa alrededor de la media noche. Al entrar a la melancólica casa, la cual estaba completamente intacta, hablaron con el forense, quien les comentó que la causa del deceso era ya sea un ataque cardíaco o un envenenamiento mediante una droga rara y oscura.

Después de interrogar a la servidumbre, con lo cual se estableció que uno de los pocos relacionados con el Sr. Dumas era el Sr. Heynes, quien es dueño de una farmacia, Holmes formuló los siguientes hechos :

**CORAZON V VENENO**

(VENENO ^ AMIGO) ==> HEYNES  
 MELANCOLICO  
 (INTACTO V DESPUES-MEDIANOCHÉ) ==> AMIGO  
 INTACTO

donde:

CORAZON ::= El Sr. Dumas muere de una  
 ataque cardiaco.  
 AMIGO ::= El asesino es amigo de la  
 victima.  
 MELANCOLICO ::= La casa es triste.  
 INTACTO ::= La casa está intacta.  
 DESPUES-MEDIANOCHÉ ::= El asesinato ocurrió despues  
 de medianoche.  
 HEYNES ::= Existe fundamento para  
 sospechar que Heynes asesinó  
 a Dumas.

Los hechos adicionales -CORAZON y DESPUES-MEDIANOCHÉ se agregaron después de la llamada del forense que establecía el tiempo del deceso a las 2:00 AM y aún más establecía que el Sr. Dumas no murió de ataque cardiaco. Después de que se agregaron estos hechos, Holmes manda arrestar a Heynes por sospecha de asesinato. La justificación, con base en la resolución, es :

FORMA ORIGINAL	FORMA NORMAL CONJUNTIVA
VENENO V CORAZON	VENENO V CORAZON
(VENENO ^ AMIGO) ==> HEYNES	~VENENO V ~AMIGO V HEYNES
(INTACTO V DESPUES-MEDIANOCHÉ) ==>	~INTACTO V AMIGO
AMIGO	~DESPUES-MEDIANOCHÉ V AMIGO
MELANCOLICO	MELANCOLICO
INTACTO	INTACTO
DESPUES-MEDIANOCHÉ	DESPUES-MEDIANOCHÉ
~CORAZON	~CORAZON

El primer paso en este proceso de resolución es reducir las FBF a la forma normal conjuntiva (FNC), una forma estandar que permite la manipulación mecánica. Este proceso de conversión se basa en las expresiones de equivalencia que vimos anteriormente. Cuando se aplica el proceso de conversión a un conjunto de FBF, produce un conjunto de cláusulas, cada una de las cuales es una disyunción finita de literales (cada cláusula está unida con alguna secuencia de literales "OR" (o) juntos).

Cada una de las cláusulas resultantes se llama un conyunto porque el conjunto total puede verse como una gran conjunción: todas las cláusulas están unidas con secuencias; dado que todas ellas requieren simultáneamente que sean verdaderas.

Davis dice que: " El uso de una FNC es completamente general porque existe un algoritmo para convertir cualquier conjunto de FBF en FNC "<sup>60</sup>. En la tabla anterior se ilustra el conjunto de ejemplo de FBF en original y en FNC correspondiente.<sup>61</sup>

Dado que todas las cláusulas están unidas (implícitamente con secuencias Y), podemos operar sobre cualquier combinación como se quiera. Podríamos, por ejemplo, considerar las siguientes cláusulas :

-VENENO V ~AMIGO V HEYNES

VENENO V CORAZON

En este caso, el literal VENENO ocurre en ambas cláusulas, tanto positivamente como negativamente. Podemos inferir una nueva cláusula, reconociendo que:

- 1) Solamente una forma de VENENO (El positivo o el negativo) puede ser verdadera.
- 2) Ambas cláusulas deben ser simultáneamente verdaderas.

Por tanto, si VENENO es verdadera, entonces ~AMIGO V HEYNES debe ser verdadera para preservar la verdad de la primera cláusula. Por el contrario, si VENENO no es verdadera, entonces CORAZON debe ser verdadera para preservar la verdad de la segunda cláusula. El resultado puede ser reformulado como, "CORAZON es verdadera o (~AMIGO V HEYNES es verdadero)". Esto se puede mostrar más formalmente como CORAZON V ~AMIGO V HEYNES.

---

<sup>60</sup> [DAVI,60]

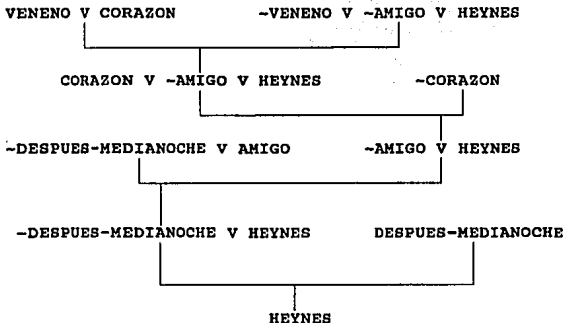
<sup>61</sup> Para una descripción completa del algoritmo de conversión, consultar [RICH,83].

Este proceso, que se denomina resolución de cláusulas, se puede generalizar como sigue:

- 1) Seleccione dos cláusulas principales que contengan la misma literal, una vez en forma positiva y otra en forma negativa.
- 2) Forme una nueva cláusula (el resolvente) mediante la disyunción de secuencias o de todas las literales a partir de las cláusulas principales excepto por la pareja cancelada.

Esta técnica es la base de la resolución, cuando se aplica iterativamente<sup>62</sup>.

Retomando el ejemplo, es posible probar la afirmación de HEYNES mediante la resolución repetida de cláusulas como se muestra a continuación:



Desafortunadamente, aunque esta forma de resolución es directa e intuitivamente atractiva, está limitada por que no existen garantías de que la aplicación de la simple resolución produzca una FBF buscada a partir de las cláusulas dadas aún si lo hace como consecuencia de ellas.

Existe sin embargo, una forma modificada de resolución

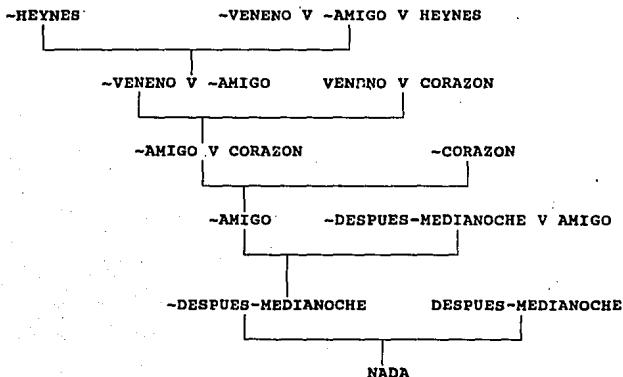
<sup>62</sup> [ROSI, 65]

que supera este problema. Como un ejemplo de esta forma de resolución, supongamos que Holmes no fue capaz de demostrarle a Watson su aseveración directamente.

Como un método alternativo, Holmes podría sugerirle a Watson que intentara demostrar que HEYNES no debería ser sospechoso, entonces debemos de estar en capacidad de agregar  $\sim$ HEYNES al conjunto de nuestras cláusulas sin dañar su consistencia. Esto es, previo a la agregación, había un conjunto de valores verdaderos que satisfacían los requerimientos de que todas las cláusulas sean simultáneamente verdaderas. (La adición de una pieza verdadera no debe invalidar esto).

A continuación se muestra una serie de la resolución que da como resultado una cláusula que consiste solamente en nada. Recordemos que un conjunto de cláusulas no se satisface si no hay una posible interpretación de las cláusulas (conjunto de valores de verdad para los símbolos en las cláusulas que harán verdaderas todas las cláusulas simultáneamente).

Como no hay, por definición, ningún camino para que nada pueda ser verdadero, se ha demostrado que las cláusulas son insatisfacibles. Debemos concluir que la insatisfacibilidad en el conjunto de cláusulas previamente satisfacibles, se ha introducido mediante la inclusión de  $\sim$ HEYNES. Por lo tanto,  $\sim$ HEYNES no es posible que sea verdadero y como  $\sim$ HEYNES es falso,  $\sim(\sim$ HEYNES), o formulado en forma más sencilla HEYNES debe ser verdadera.



El proceso de razonamiento descrito anteriormente es una descripción informal de la resolución por refutación, que es una forma típica de resolución. La resolución por refutación está garantizada para producir una cláusula buscada a partir de un conjunto de cláusulas si la cláusula de hecho resulta de ellas.

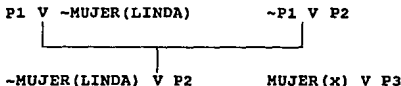
La utilidad de este procedimiento se basa en el hecho de que la resolución es una refutación completa<sup>63</sup>; esto es, que la resolución por refutación está garantizada para demostrar eventualmente la insatisfacibilidad si existiese alguna.

El proceso de resolución para un conjunto dado de FBF (expresada mediante el empleo de lógica proposicional solamente) se puede describir más formalmente como :

- 1) Negar la proposición buscada (llamada el objetivo) y agregarla al conjunto de FBF para conformar un nuevo conjunto.
- 2) Convertir el conjunto FBF a un conjunto FNC.
- 3) Repetir hasta que se produzca nada o hasta que ningún proceso sea posible:
  - a. Seleccione dos cláusulas padres.
  - b. Resuelva las cláusulas padres.
  - c. SI el revolvente es nada,  
ENTONCES concluya que se encontró una contradicción y se demostró el objetivo.
  - SI NO agregue el revolvente al conjunto de cláusulas.

El proceso de resolución, según se ha descrito hasta ahora, se basa en el hallazgo y en la cancelación de dos literales opuestas. No tenemos ningún problema en la determinación si las dos literales candidatas deben realmente cancelarse mientras incluyan solamente predicados sencillos según se mostró anteriormente. Sin embargo, el proceso se torna más complicado cuando consideramos el proceso general (expresiones en lógica de predicados) en que debemos decidir si dos literales con predicados, funciones y variables se pueden cancelar.

Consideremos el siguiente ejemplo :



<sup>63</sup> [CHAN, 73]



donde  $P1$  y  $\neg P1$  se cancelan inmediatamente, ¿pero qué debemos de hacer con  $\neg MUJER(LINDA)$  y  $MUJER(x)$ ? Ciertamente deberíamos darnos por vencidos si uno de los símbolos de predicados fuera por ejemplo,  $NERAK$ ; pero debido a que en este caso los símbolos de predicados son idénticos y opuestos, nos vemos tentados a tratar de cancelarlos.

Ciertamente estas dos literales podrían no verse como idénticas para todos o aún para la mayoría de las interpretaciones, pero en el proceso de resolución estamos buscando un par de literales que tengan al menos una interpretación para la cual no puedan ser simultáneamente verdaderas. Si el conjunto de cláusulas debe permanecer consistente, no puede incluir ambas  $MUJER(x)$  y  $\neg MUJER(LINDA)$  porque existe un caso, ese en que  $x = LINDA$ , que es inconsistente.

La unificación es el proceso formal de determinar si las dos literales se pueden convertir para equipararse idénticamente. El proceso de unificación hace énfasis en la sustitución de los términos - variables, constantes o expresiones de función (un término de función y los términos asociados) - por variables en los literales.

Una instancia de sustitución de una literal resulta de tal sustitución. Un conjunto específico de sustitución es un conjunto de pares ordenados tal que el primer elemento, es el nuevo artículo y el segundo elemento es la variable por la cual es sustituido.

El procedimiento de unificación puede ser visto informalmente como un proceso de equiparamiento de los elementos en dos listas para determinar si hay algún conjunto de sustitución que haría las listas idénticas.

Este proceso es de la siguiente manera<sup>64</sup>:

1) Represente cada predicado como una lista en que el símbolo de predicado es el primer elemento, que está seguido por los argumentos del predicado en orden.

2) Abandone si las dos listas no tienen la misma longitud.

3) Haga una comparación por parejas de los elementos en la lista mediante el empleo de las siguientes reglas:

\* Los símbolos de predicados, los símbolos de funciones y las constantes deben emparejar

---

<sup>64</sup> Para ver este procedimiento más formalmente, se recomienda [MOSI, 84].

exactamente.

- \* En cuanto a las variables establezca una pareja de sustitución. Específicamente, cuando encuentre una variable, reemplácela, y todas las subsiguientes ocurrencias de ella en la lista, con el correspondiente elemento a partir de la otra lista. La única restricción en este emparejamiento es que una variable no se puede reemplazar por un término que contenga la misma variable (prevenir ciclos infinitos).
- \* Los dos predicados se pueden unificar si todos los elementos se emparejan. Note que puede ser necesario invocar la rutina recursivamente para evaluar la lista de elementos que están en listas anidadas.

Al usar la unificación, es posible extender la resolución para obrar sobre expresiones en lógica de predicados (en lugar de una lógica proposicional simple). La resolución en cuanto a lógica de predicados es la misma que aquella que se describió previamente excepto que :

- 1) Dos literales deben ser unificadas antes que se puedan cancelar.
- 2) Las sustituciones hechas para lograr la unificación dentro de una cláusula deben ser elaboradas a través de la cláusula completa, no solamente en la literal que está siendo unificada.

Mediante la formalización de la lógica se ha avanzado en el intento por ampliar las posibilidades de respuestas como procesos no infinitos; es la misma secuencia que encontramos en los algoritmos, y la heurística como posibles respuestas que se acercan a lo aceptable.

## CAPITULO VI

### ALGORITMOS VERSUS HEURISTICA

El concepto de algoritmo es uno de los conceptos básicos de las matemáticas, y por este se entiende una lista de instrucciones donde se especifica una sucesión de operaciones necesarias para resolver cualquier problema de un tipo dado. Por supuesto, ésta no es una definición matemática y precisa del término, pero corresponde al concepto de algoritmo que surgió naturalmente y que se ha usado en matemáticas desde la antigüedad.<sup>65</sup>

En matemáticas, se considera que una serie de problemas está resuelta cuando se encuentra un algoritmo que la resuelve. Descubrir esos algoritmos es una meta natural de las matemáticas. Si no existe un algoritmo para resolver todos los problemas de un tipo dado, siempre es posible que un matemático invente un procedimiento que resuelva ciertos problemas de un tipo, aunque no sea aplicable a los demás casos.

En 1901, en un Congreso Internacional de Matemáticas en París, David Hilbert presentó una lista de veinte problemas no resueltos y señala la importancia de resolverlos ante la sociedad matemática. Entre estos problemas se encontraba el siguiente (Décimo problema de Hilbert): "Hallar un algoritmo para determinar si toda ecuación diofantina tiene una solución entera".<sup>66</sup>

Para el caso particular de las ecuaciones diofantinas con una incógnita se conoce tal algoritmo. Si una ecuación

$$A_n X^n + A_{n-1} X^{n-1} + \dots + A_1 X + A_0 = 0$$

con coeficientes enteros, tiene una solución entera  $X_0$ , entonces  $A_0$  es divisible entre  $X_0$ . Esto sugiere el siguiente algoritmo:

- 1) Hállense todos los divisores del número  $A_0$  (sólo existe un número finito de ellos y se tiene un algoritmo para hallarlos a todos).

---

<sup>65</sup> El término algoritmo proviene del nombre de un matemático medieval de Uzbekistán, Al-Khowárizmí.

<sup>66</sup> Una ecuación diofantina es una ecuación de la forma  $P = 0$ , donde  $P$  es un polinomio con los coeficientes enteros, y para las cuales deben de buscarse soluciones enteras.

- 2) Substitúyanse sucesivamente cada uno de estos divisores en el primer miembro de la ecuación y calcúlese el valor resultante.
- 3) Si cualquiera de los divisores da un valor de cero para el primer miembro, entonces este divisor es una raíz de la ecuación; si ninguno de los divisores da cero, entonces la ecuación no tiene raíces enteras.

Muchos matemáticos han trabajado y siguen trabajando en el problema de Hilbert, pero todavía no se ha encontrado el algoritmo requerido para el caso general con dos o más incógnitas. Además, no parece que exista probabilidad alguna de encontrar ese algoritmo.

Del ejemplo anterior resulta evidente que los algoritmos (en nuestro caso es un algoritmo numérico) poseen las características siguientes :

\* La naturaleza determinativa de los algoritmos. Un algoritmo debe darse en la forma de una lista finita de instrucciones que indiquen el procedimiento a seguir en cada paso del cálculo. Así, el cálculo no depende del que lo hace; es un procedimiento determinativo que puede repetir con éxito cualquier persona en cualquier momento.

\* La generalidad de los algoritmos. Un algoritmo es una lista única de instrucciones que definen un cálculo que puede efectuarse con cualesquiera datos iniciales y que, en cada caso, proporciona el resultado correcto. En otras palabras, un algoritmo dice cómo resolver no precisamente un problema particular, sino toda una clase de problemas similares.

Dado que un algoritmo, especialmente uno práctico<sup>67</sup> normalmente se basa en argumentos sutiles y complicados, su construcción requiere de gran inventiva; pero una vez que se ha encontrado el algoritmo, lo puede aplicar incluso una persona que ignora su objetivo. Sólo se necesita que esta persona sepa efectuar las pocas operaciones elementales que requiere el algoritmo y siga las indicaciones al pie de la letra. Procediendo en forma bastante mecánica; puede resolver cualquier problema del tipo para el cual se diseñó el algoritmo.

Qué tan práctico sea aplicar un algoritmo depende de la complejidad del problema, la rapidez con la que puedan efectuarse

---

<sup>67</sup> Existen algunos problemas, como el ajedrez, con los cuales se puede encontrar una estrategia ganadora, pero dichos algoritmos que son potencialmente factibles, no son prácticamente factibles, debido a la explosión combinatoria.

las operaciones necesarias y el tiempo que uno está dispuesto a dedicarle. Por supuesto, se desea saber si los procedimientos son prácticamente factibles; desafortunadamente no existe un criterio matemático preciso, para distinguir entre los procedimientos prácticos y los que no lo son. Esto depende de los medios disponibles para la computación, y puede cambiar, por ejemplo, con el desarrollo de la tecnología. Así, desde el advenimiento de la computadora, muchos procedimientos que antes no eran factibles, ahora pueden realizarse.

No importa lo laborioso que resulte un algoritmo, el mero hecho de que exista es un hecho notable. ; Por que hasta ahora no se ha encontrado ningún algoritmo para el problema de Hilbert sobre ecuaciones diofantinas !. Mientras tanto, el descubrimiento de un algoritmo, incluso uno que sea difícil de aplicar inspira la confianza de que se podrá simplificarlo, o bien construir otro algoritmo más conveniente.

Sin embargo, para solucionar eficientemente la mayoría de los problemas difíciles, a menudo es necesario comprometer los requerimientos de movilidad y sistematicidad, y construir una estructura de control<sup>68</sup> que aunque no nos garantice que encontremos la mejor respuesta, por lo menos proporcione una respuesta muy buena. Así introducimos la idea de técnica heurística<sup>69</sup>. Una técnica heurística es aquella que mejora la eficiencia del proceso de búsqueda, posiblemente a cambio de sacrificar su completitud. Las técnicas heurísticas son como las guías turísticas. Son buenas cuando apuntan a direcciones interesantes; son malas cuando conducen a callejones sin salida. Algunas técnicas heurísticas ayudan a guiar un proceso de búsqueda sin sacrificar ninguna aspiración de completitud que el proceso pueda haber tenido previamente. Otras (de hecho, muchas de las mejores) pueden llevar a que pase inadvertido un camino excelente. Pero, en promedio, mejoran la calidad de los caminos que se exploran. Usando buenas técnicas heurísticas, podemos esperar lograr buenas (incluso óptimas) soluciones a problemas difíciles en un tiempo menor que el exponencial (algunas heurísticas se ejecutan en un tiempo proporcional al cuadrado de  $N$ , que es una mejora significativa sobre  $N!$ ). Hay algunas buenas técnicas heurísticas de propósito general que son útiles en una gran variedad de dominios de problemas. Además, es posible construir técnicas heurísticas para propósitos especiales que exploten el conocimiento específico del

---

<sup>68</sup> La estructura de control especifica el orden en que se comparan las reglas con la base de datos y una manera de resolver los conflictos que suceden cuando diversas reglas se casan a la vez.

<sup>69</sup> La palabra heurística viene del griego HEURISKEIN, que significa 'descubrir', lo cual es también el origen de EUREKA, que se deriva de la famosa exclamación de Arquímedes HEURIKA ("lo encontré"), que lanzó al descubrir un método para determinar la pureza del oro.

dominio para resolver problemas particulares.

En las técnicas heurísticas de propósito general a menudo es posible probar la existencia de cotas de error, mismas que nos aseguran que no se está pagando un precio mayor en la precisión a cambio de la velocidad. Sin embargo, en muchos problemas de IA, no es posible producir tales cotas tranquilizadas. Esto es cierto por dos razones:

- \* Para problemas del mundo real, a menudo es difícil medir con precisión la bondad de una solución particular.

- \* Para problemas del mundo real a menudo es útil introducir técnicas heurísticas basadas en conocimientos relativamente poco estructurados. Frecuentemente es imposible definir este conocimiento de forma que pueda efectuarse un análisis matemático de su efecto en el proceso de búsqueda que pueda ser llevado a cabo.

Incluso en tales situaciones no estructuradas, puede ser posible decir algo sobre la eficiencia del proceso de búsqueda.

Existen muchas técnicas heurísticas que, aunque no son tan generales, son, sin embargo, útiles, en una amplia variedad de dominios. Por ejemplo, consideremos la tarea de descubrir ideas interesantes en alguna área específica. A menudo es útil la siguiente técnica heurística :

Si existe una función interesante de dos argumentos  $f(x,y)$ , mirar qué sucede si los dos argumentos son idénticos.

En el dominio de las matemáticas, esta técnica conduce al descubrimiento de los cuadrados si  $f$  es la función multiplicación, y conduce al descubrimiento de una función de identidad si  $f$  es la función de unión de conjuntos. En dominios menos formales, esta misma técnica heurística conduce a otros descubrimientos. Para este tipo de heurísticas se ha acuñado la palabra heurética, y el estudio de estas heuréticas ha sido posible gracias a los esfuerzos por construir programas conducidos heurísticamente.<sup>70</sup>

Sin técnicas heurísticas, estaríamos atrapados sin esperanza en la explosión combinatoria. Esto por sí solo sería argumento suficiente en favor de su uso. Pero existen también otros

---

<sup>70</sup> Se recomienda [LENA,82] y [LENA,83] para una discusión de algunos de los puntos involucrados en este estudio.

argumentos<sup>71</sup>:

1) Pocas veces se necesita la solución óptima; usualmente una buena aproximación nos servirá muy bien. De hecho alguna evidencia muestra que la gente, al resolver problemas, no suele optimizar sino más bien satisfacer los requerimientos.<sup>72</sup>

2) Generalmente una aproximación producida por una técnica heurística es lo suficientemente buena como para solucionar el problema.

La explosión combinatoria hace que frente a ciertos problemas basten los expedientes de los algoritmos o de la heurística, ya que frente a problemas difíciles se contentan algunos con la aproximación a la respuesta, una buena respuesta, aunque no la óptima y se avanza en sistemas como veremos en la representación del conocimiento.

---

<sup>71</sup> Para profundizar en los fundamentos de programación heurística y sus aplicaciones, véase [DELG,87].

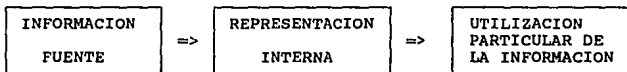
<sup>72</sup> [SINO,81]

## CAPITULO VII

### REPRESENTACION DE CONOCIMIENTO

La representación de conocimiento comprende el estudio de formalismos y de estructuras de datos en el tratamiento de objetos, hechos que establecen relaciones, conceptos, eventos, etc.; los cuales serán usados en la comprensión del lenguaje natural, resolución de problemas, la comprensión de escenas, entre otras.

Desde un punto de vista funcional un sistema opera de la siguiente manera : a partir de una información fuente, ésta es captada y llevada a una representación interna, la cual será posteriormente interpretada, en función de su utilización. Esto se ejemplifica en la siguiente figura :



Las técnicas de representación que voy a mostrar son :

- Mediante cálculo de predicados, que utiliza la sintaxis estándar de la lógica simbólica.
- Mediante redes, las cuales emplean principalmente gráficas dirigidas.
- Mediante redes de producción.

#### 7.1 Representación con Cálculo de Predicados.

El Cálculo de Predicados, como vimos anteriormente, es una extensión del Cálculo Proposicional, el cual trata esencialmente con enunciados o proposiciones, así como las relaciones entre ellos.

Un ejemplo de proposición es el siguiente :

P: "Las acciones son una inversión de renta variable"



este mismo enunciado en el cálculo de predicados se expresaría como

**INVERSION--RENTA--VARIABLE (acciones)**

Es posible formar proposiciones compuestas, utilizando las siguientes conectivas :

Y	denotado como	AND	$\wedge$ , &
O	denotado como	OR	$\vee$ (no exclusivo)
NO	denotado como	NOT	$\neg$ , ~
IMPLICA	denotado como	IF-THEN	$\implies$
EQUIVALENTE	denotado como	IF-AND-ONLY-IF	$\equiv$

La aplicación de las conectivas es como sigue :

Dadas las proposiciones :

- P : Las acciones están a la alza.  
Q : Las acciones son una buena inversión.

- a) La conjunción (AND) aplicada a las dos proposiciones se representa por:  $P \wedge Q$  significa,  
"Las acciones están a la alza y las acciones son una buena inversión".
- b) La disyunción (OR) representada por :  $P \vee Q$  significa,  
"Si las acciones están a la alza, Entonces las acciones son una buena inversión".
- c) La bicondicional (IF-AND-ONLY-IF), representada por :  $P \equiv Q$  significa,  
"Las acciones están a la alza si y sólo si, las acciones son una buena inversión".

El resultado de evaluar una proposición puede tomar dos

valores : FALSO (F) o VERDADERO (V); los cuales se resumen a continuación :

P	Q	$P \wedge Q$	$P \vee Q$	$\neg P$	$P \Rightarrow Q$	$P \equiv Q$
V	V	V	V	F	V	V
V	F	F	V	F	F	F
F	V	F	V	V	V	F
F	F	F	F	V	V	V

Un sistema que utiliza el cálculo de Predicados se puede ilustrar con el siguiente enunciado :

" Los fondos de inversión dan ganancias a sus inversionistas "

el cual puede escribirse bajo la forma:

DAR(donador, o-dado, receptor)

o representar mediante varios predicados binarios :

DONADOR (FONDO-DE-INVERSION)  
 O-DADO (GANANCIAS)  
 RECEPTOR (ACCIONISTAS)

Como puede verse, todo predicado requiere de uno o más argumentos : cada argumento es un término, el cual puede ser una constante, una variable o una función. En nuestro caso, los argumentos son : FONDO DE INVERSION, GANANCIAS, ACCIONISTAS; los cuales a su vez son variables. Mientras que : DAR, DONADOR, O-DADO son predicados.

Una función sería : (VALORES-RENTA-FIJA X), la cual puede tomar uno o más valores, como por ejemplo: Petrobonos, Cetes, etc., los cuales a su vez son constantes.

La utilización del cálculo de predicados es amplia, es decir que un enunciado se puede generalizar tanto como uno quiera:

"Si Petrobonos varia, entonces, este es un valor de renta fija"

hecho que puede generalizarse usando variables y cuantificadores, esto es :

"Si una emisión varia, entonces emisión es un valor de renta fija"

En el enunciado anterior, emisión es la variable; todo objeto instanciado por ésta, en este caso, PETROBONOS, que cumpla con la condición especificada, hace que se efectuen las modificaciones pertinentes.

Sin embargo, la composición de funciones y de predicados no se lleva a cabo en cualquier orden, sino que, es necesario establecer qué valores sustituyen a otros símbolos, lo cual ilustramos a continuación:

- a) Sea el predicado OBJETO-ANIMADO (X), cuyo valor es falso o verdadero, es decir :

Si X instancia ALBIN, el valor es VERDADERO.

en cambio

Si X instancia LIBRO, el valor es FALSO.

- b) Ahora bien, la función CAPITAL (X) da como valor MEXICO.

Si X instancia CIUDAD DE MEXICO.

Si aplicamos primero OBJETO-ANIMADO sobre CAPITAL, en el mayor de los casos dá como resultado FALSO, ya que

**OBJETO-ANIMADO (CAPITAL (X))**

cuando X instancia MEXICO, entonces OBJETO-ANIMADO(CIUDAD DE MEXICO) da falso.

Por el contrario, aplicar la función CAPITAL (X), siendo X, OBJETO-ANIMADO (X), no tiene sentido en virtud de que este último produce por definición falso o verdadero, es decir :

CAPITAL (OBJETO-ANIMADO (X)),

donde

OBJETO-ANIMADO(X) evalúa FALSO O VERDADERO

entonces

CAPITAL (FALSO) o CAPITAL(VERDADERO)

carece de sentido.

Tanto los cuantificadores, los operadores, las variables, los predicados y las reglas para poner en marcha el proceso de inferencia del sistema, constituyen el cálculo de predicados de primer orden.

El Cálculo de Predicados de Primer Orden, se refiere a la cuantificación ejecutada sobre los argumentos de los predicados; cuando se efectúa a nivel superior, como son los predicados y como sería el caso del predicado mismo.

Un ejemplo de orden superior, es el siguiente axioma de inducción :

$$\forall P \quad (( P(0) \wedge \forall n (P(n) \Rightarrow P(n+1)) \Rightarrow \forall n P(n))$$

fórmula que puede leerse como:

"Para todo predicado P, si P es verdad para 0, y para todo n, P(n) implica P(n+1), entonces P(n) es verdad para todo n".

Resumiendo podemos decir, que la resolución de problemas cuya naturaleza es deductiva<sup>73</sup>, su tratamiento es viable mediante la Lógica Simbólica; sin embargo, la mayoría de los problemas reales son de naturaleza inductiva, por ejemplo la interpretación de los conocimientos necesarios para el diagnóstico de un padecimiento que implica conocimientos inciertos; de ahí que se requieran otros tipos de representación de conocimientos, como veremos más adelante.

---

<sup>73</sup> Algunos ejemplos de sistemas que utilizan esta representación se pueden encontrar en [RAPH,68] y [GREE,69].

## 7.2 Redes Semánticas

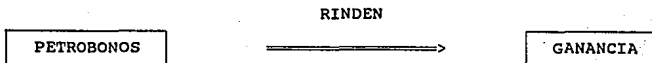
La idea de base en redes semánticas fue introducida por Rose Quillian<sup>74</sup> en su tesis de Ph. D.; esta red representa un modelo de red asociativa conocido como **MEMORIA SEMANTICA**; estableciendo de manera formal un **OBJETO**, proporcionando su significado a semejanza de como lo realiza el ser humano.

Esta representación se compone de un conjunto de nodos relacionados entre sí, por medio de **LIGAS** o **ARCOS**; cada nodo corresponde a un atributo, un estado, una entidad o un evento; las **LIGAS** que van de un nodo a otro establecen relaciones entre los nodos; también es posible que un nodo corresponda a una subred semántica. La organización de la información en este esquema es similar a la presentación que poseen los diccionarios.

A continuación se presenta cómo evolucionan las redes semánticas. Iniciemos con el enunciado siguiente :

**" Los petrobonos rinden ganancias "**,

lo cual puede representarse como:



otra forma, no gráfica, puede ser :

**RINDEN (PETROBONOS, GANANCIA)**

Como cada concepto está ligado a la familia a la cual pertenece, y en virtud de que :

**" Los Petrobonos son emisiones y las ganancias producen un rendimiento "**

la red deviene :

---

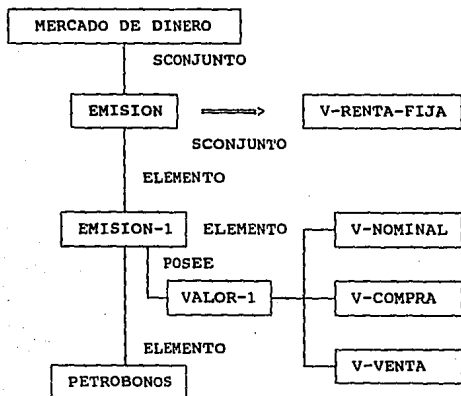
<sup>74</sup> [QUILL, 68]



Consideremos ahora, que nuestra información crece y es significativa, por ejemplo, sabemos que :

"Un Petrobono es un tipo de emisión; las emisiones pertenecen al mercado de dinero, así como a los valores de renta fija; por otra parte este tipo de emisión posee diversos valores como : valor nominal, valor de compra, valor de venta, entre otras características."

Información que puede representarse como sigue :



Los nodos etiquetados son ejemplos de conceptos más generales y se ligan mediante el arco ELEMENTO.

También se tiene la posibilidad de heredar ciertos atributos; como en el ejemplo anterior PETROBONOS hereda los atributos de EMISION y de MERCADO DE DINERO, con el sólo hecho de recorrer la red hacia arriba, por los arcos SCONJUNTO; aunque no siempre este proceso es muy confiable.

Hasta aquí podemos observar que no hay una estandarización del formalismo, como puede darse en la Lógica de primer orden. Sin embargo, existen algunos estudios como el de William Woods<sup>75</sup>, acerca de qué conocimiento puede representarse sobre los arcos.

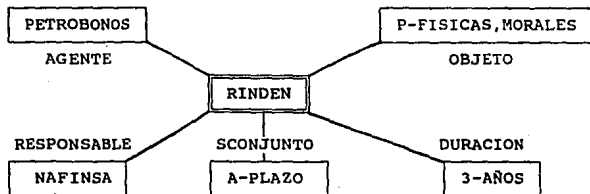
A medida que la cantidad de información crece y que la utilización deseada de la información se diversifica, la complejidad de la red aumenta; como puede observarse en los siguientes enunciados :

a) " Los Petrobonos rinden ganancias "

enunciado esquematizado con antelación, el cual no proporciona mayor información acerca de :

- \* ¿ Qué rinde beneficios ? : Los Petrobonos.
- \* ¿ Quienes son los adquirientes ? : Personas físicas o morales.
- \* ¿Cuál es su duración ? : 3 años.

De ahí que sea más conveniente utilizar RINDEN como un nodo y no como un arco; obteniéndose los siguientes esquemas, que incluyen otros atributos que contestan las preguntas arriba mencionadas, es decir :



<sup>75</sup> [WOOD, 75]

Por otra parte podemos tener más de un enunciado semejante al anterior como sería, por ejemplo :

b) " CERTIFICADOS PLATA rinden ganancia " .

con sus respectivas preguntas

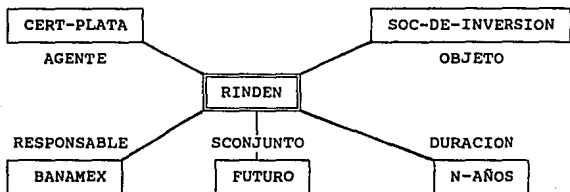
\* ¿ Quién es la emisora ? : Banco Nacional de México .

\* ¿ Qué tipo de emisión es ? : 1 , 2 .

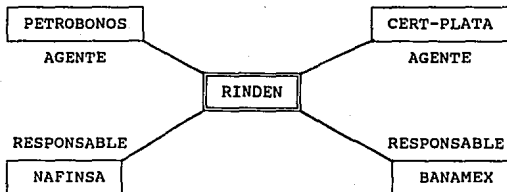
\* ¿ Quienes son los posibles adquirientes ? :

personas físicas, morales,  
sociedades de inversión, -  
instituciones de crédito, -  
de seguros y de finanzas, -  
etc.

En forma similar al esquema del enunciado (a), obtenemos la siguiente red :

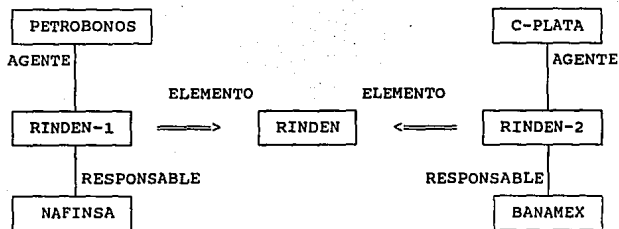


Sin embargo, si conjuntamos las dos redes, habrá información que quedará ambigua, es decir :



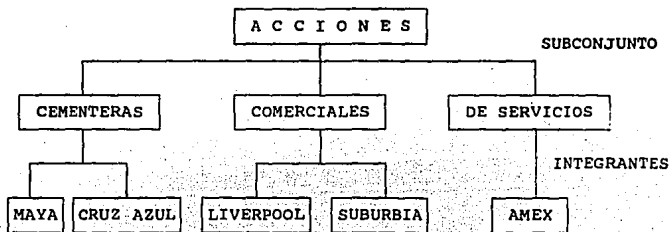


Dado que no se sabe ¿ quién es el responsable de qué ?, ni ¿ quién es el agente colocador ?. En consecuencia, es necesario instanciar a cada ítem de información por separado como parte del concepto RINDEN, es decir :



Este esquema es muy similar a la representación de la Gramática de Casos de Fillmore<sup>76</sup>. Se encuentran también otras aplicaciones en el campo de la comprensión del Lenguaje Natural, como por ejemplo, Simmons<sup>77</sup>.

Otro ejemplo de utilización de las redes semánticas es la clasificación o taxonomía efectuada sobre subconjuntos homogéneos; en donde se realiza la transmisión de propiedades (herencia) de los nodos antecesores sobre los nodos sucesores; como por ejemplo, en el caso de los diferentes tipos de acciones que pueden clasificarse según su ramo, y sus diversos integrantes, como sigue :



<sup>76</sup> (FILL, 68)

<sup>77</sup> (SIMM, 73)

De esta manera una red semántica permite deducir informaciones que no necesariamente se encuentren definidas para cada elemento terminal, pero que están asociadas por el hecho de pertenecer a una clase determinada.

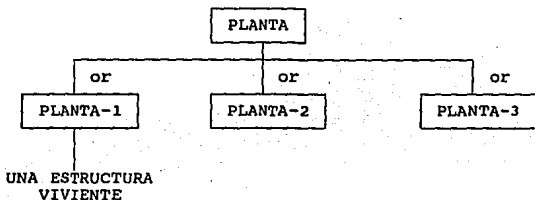
Con respecto a los trabajos de Quillian, veamos el siguiente ejemplo :

Se quiere representar el objeto **PLANTA**, el cual tiene varias acepciones :

- a) Estructura viviente; no es animal; posee frecuentemente hojas; se nutre del aire, agua y tierra.
- b) Instalación industrial.
- c) Inflexión del verbo plantar, ...

Debido a que existen más de una acepción, su representación está constituida por sub-classes. Las tres definiciones están **LIGADAS** por medio de un arco **OR**. Los nodos pueden ser modificados por otras estructuras, por ejemplo : **USO** modifica **APARATO**; las ligas pueden ser conjunciones etiquetadas por **AND**; las relaciones sujeto/objeto se representan por ligas paralelas (ejemplo : **USO** y **GENTE**).

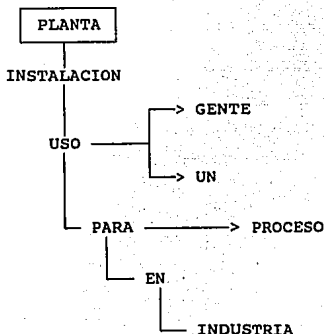
La representación esquemática es la siguiente ;



A) Representación general del concepto **PLANTA**, primera acepción.

Mediante este tipo de red es posible contestar preguntas relacionadas con el conocimiento comprendido ahí. Por ejemplo,

dadas dos palabras se buscan las posibles relaciones que pueden tener mediante una búsqueda en la vecindad de ambas, hasta encontrar al menos una relación.



B) Segunda acepción de PLANTA

El trabajo de Winston (1970-1975)<sup>78</sup> es otro ejemplo típico de redes; entre las características más sobresalientes de este trabajo, está el hecho de poder inferir "CONCEPTOS" de estructuras físicas, tales como un ARCO.

Dada una escena, se establece la descripción de cada objeto físico, comprendidas en un nodo; las ligas entre ellos corresponden a relaciones entre los objetos, como son : A-LA-IZQUIERDA-DE, ENFRENTA-DE Y SOSTENIDO-POR; éstas relaciones pueden modificarse, es decir:

A-LA-IZQUIERDA-DE es equivalente a decir OPUESTO-A-LA-DERECHA-DE

Uno de los objetivos es determinar las diferencias entre los objetos, mediante relaciones simétricas entre escenas, por ejemplo :

<sup>78</sup> [WINS,75]

**A-LA-IZQUIERDA-DE** es simétrica con **A-LA-DERECHA**  
**ENFRENTE-DE** es simétrica con **ATRAS-DE**.

En resumen, la utilización de las redes semánticas versa principalmente sobre la representación de conceptos; asimismo, se han empleado en sistemas expertos, tales como, PROSPECTOR<sup>79</sup>.

Conforme más completa deviene una red semántica, la potencialidad de razonamiento del sistema aumenta; igualmente, hay una relación estrecha entre las estructuras de datos y la complejidad del interprete.

Finalmente, una red establece convenciones arbitrarias, las cuales resultan irrelevantes en el caso de las definiciones matemáticas y en los teoremas. Por otra parte, una red desarrolla una comprensión intuitiva.

### 7.3 Reglas de Producción.

El formalismo de reglas de producción ha sido utilizada desde 1943 por Post<sup>80</sup>, también lo encontramos en los algoritmos de Markov y en la lingüística de Chomsky<sup>81</sup>, que utiliza reglas de reescrituración en el reconocimiento sintáctico de frases de Lenguaje Natural.

Una regla de producción consta de un par ordenado de símbolos identificado como miembro izquierdo y miembro derecho, formando una pareja situación-acción. Por ejemplo :

**" Si una inversión presenta alta liquidez,  
Entonces diversificar el portafolio con esta inversión "**

en este caso la situación es : Inversión presenta alta liquidez y la acción es diversificar.

La sintaxis o escritura de las reglas es diversa, y está en función del proceso de selección y de ejecución de las mismas. Esta sintaxis es reconocida por el intérprete, que accesa las

---

79 [DUDA, 78] y [DUDA, 80]

80 [POST, 43]

81 [CHOM, 57]

reglas, que son en sí una cadena de símbolos; aunque algunos sistemas aceptan el uso de variables<sup>82</sup>; o también, clases sintácticas, un ejemplo son los sistemas "parsing".

Por otra parte, las reglas presentan la posibilidad de emplear disyuntivas, aseveraciones, negaciones y variables cuantificables.

El intérprete tiene como función llevar a cabo el proceso de inferencia. El proceso de inferencia, comprende un ciclo llamado de Control que consiste en :

- \* Seleccionar las reglas pertinentes.
- \* Verificar entre estas reglas, cuales se satisfacen.
- \* Ejecutar las acciones especificadas por estas reglas.

Desde un punto de vista funcional el sistema opera en dos fases, una de reconocimiento y otra de acción. La primera se subdivide a su vez en : seleccionar las reglas pertinentes y en resolver el conflicto de resolución en caso de que exista más de una regla aplicable a una situación dada. Por ejemplo :

Sean las reglas

- 1) " Si una inversión desciende de su valor en libros, Entonces congelar la inversión hasta su recuperación ".
- 2) " Si una inversión desciende de su valor en libros, Entonces liquidarla ".

Como se observa en el caso anterior, la situación en ambas reglas es la misma, en consecuencia el conflicto de resolución, debe determinar cual de las dos acciones ejecutar. Existen diversos criterios, entre otros :

- a) Establecer orden en los datos.
- b) Clasificar las reglas con prioridad de ejecución.
- c) Ejecutar la regla más recientemente instanciada.
- d) Aplicar metareglas.

En la segunda fase se ejecutan las acciones establecidas por las reglas durante el proceso de inferencia. En el caso de EURISKO<sup>83</sup>, las acciones establecidas durante el proceso de

---

82 [WATE,78]

83 [LENA,83]

inferencia se colocan en una AGENDA, ésta contiene a cada paso otra AGENDA de tareas relacionadas con conceptos y/o especializaciones.

La selección de las reglas depende de la situación en curso de tratamiento. Si se consideran los hechos establecidos y se verifican con las reglas, se dice que el razonamiento del sistema es hacia adelante, ya que de cumplirse el miembro izquierdo de la regla, se ejecuta la parte derecha de ésta. Por el contrario, si se consideran las metas posibles a alcanzar, es decir, el miembro derecho de la regla y se verifican sólo las reglas que concluyen esas metas, se trata del razonamiento hacia atrás.

En suma, los Sistemas de Producción permiten representar los conocimientos en forma modular y uniforme; son útiles en los casos donde se detecta y trata con una gran cantidad de estados independientes; en sistemas cuyos objetivos son amplios y cuyas acciones son de corto alcance y de toma de decisión súbita.

Una de las características sobresalientes de los sistemas basados en reglas de producción es su potencialidad de aprendizaje; es decir, que a partir de su base de conocimientos inicial, el sistema sea capaz de generar o simplificar las reglas de inferencia que rigen su comportamiento.

Con la representación del concepto vemos la complejidad de los sistemas y la solución puede buscarse deductivamente con el cálculo de predicados o inductivamente con las redes semánticas o las reglas de producción para representar frases de Lenguaje Natural. Asimismo como avance se presentan los lenguajes de Inteligencia Artificial en la complejidad de los lenguajes típicos LISP (List Programing) y PROLOG (Programing in Logic) en el capítulo final.

## CAPITULO VIII

### LOS LENGUAJES DE LA INTELIGENCIA ARTIFICIAL.

Un ingrediente importante en los lenguajes de programación de la IA es la habilidad para implementar un sistema basado en símbolos; esta capacidad para asimilar y transformar estructuras simbólicas es una necesidad en el desarrollo de un sistema "inteligente". Mientras la mayoría de los lenguajes de programación hacen hincapié en la manipulación de datos numéricos, los lenguajes de IA explotan la representación de conocimiento a través de estructuras simbólicas.

Los lenguajes de IA deben manejar mecanismos para expresar y manipular conocimiento; esto se logra a través de formalismos lógicos que permitan el uso de inferencias. Para lograr un alto grado de eficiencia deben contener un mecanismo estandarizado de control, y al mismo tiempo, permitir el desarrollo de un control óptimo y métodos de inferencia.

Dichos lenguajes deben proveer accesos simples a una serie de estructuras dinámicas y complejas, como son listas, árboles y arreglos. Es de vital importancia que la referencia hacia las estructuras de datos se mantenga lo más simple posible.

Los lenguajes de IA poseen las siguientes propiedades básicas :

- a) Son de tipo intérprete.
- b) No siguen los fundamentos de una máquina de Vonn Neumann.
- c) Tienen una sólida base matemática que les permite un razonamiento formal adecuado a las circunstancias.

A continuación presentamos los dos lenguajes típicos de la IA : LISP y Prolog, buscando poner de manifiesto sus méritos y sus limitaciones.

#### 8.1 LISP.

Lisp fue inventado en 1958 por John McCarthy<sup>84</sup> con el objeto de implementar un lenguaje práctico para la manipulación de listas, ésto lo hace el segundo lenguaje más viejo (después de

---

<sup>84</sup> (MCCA, 60)

Fortran) que aún se usa de una manera generalizada.

McCarthy implementó en Lisp las siguientes ideas :

- 1) Cómputo de expresiones simbólicas en lugar de expresiones numéricas.
- 2) Representación y procesamiento de los datos como listas.
- 3) Estructura de control basada en la composición de funciones que se combinan para formar funciones más complejas.
- 4) Descripción de procesos y problemas por medio de la recursión.
- 5) Representación de los programas en Lisp de la misma forma que los datos (i.e. datos = programas).
- 6) Definición formal del lenguaje en términos de la función EVAL, que le sirve como intérprete.

En Lisp sólo existe un tipo de datos : la lista; la cual se define recursivamente como una secuencia de cero o más elementos contenidos entre paréntesis y en donde cada elemento puede ser un átomo o una lista.

Lisp es un lenguaje de programación con carácter estrictamente funcional : toda la programación consiste en definir y usar funciones. Cada función en Lisp acepta cierto número de argumentos y efectúa ciertas operaciones sobre ellos calculando así el valor resultado.

Los datos procesados tienen la forma general común de las expresiones simbólicas (S-expresiones). Lo que más distingue a Lisp de los demás lenguajes de programación es que tanto los programas en Lisp como los datos tienen la forma de S-expresiones.

Esta característica implica varias consecuencias importantes de las que cabe mencionar la definición precisa de la semántica de programas; en Lisp usando el mismo lenguaje de Lisp, y la posibilidad de crear ciertas partes de programas como resultado de la ejecución de otras partes.

Otro rasgo característico de Lisp es su forma interpretativa e interactiva de ejecución de programas. Cualquier sistema Lisp consiste principalmente en un intérprete que ejecuta en forma cíclica las tres fases básicas de procesamiento de



programas en Lisp :

- se lee la siguiente parte ejecutable del programa,
- se evalúa, o sea se ejecutan las operaciones buscadas,
- se anuncia el valor resultado.

Las partes ejecutables de programa Lisp tienen la forma de E-expresiones (expresiones evaluables) y son básicamente llamados para alguna función. A diferencia de las matemáticas y los lenguajes de programación convencionales en el que el uso de la función fn con argumentos arg1, arg2, ..., argn tiene la forma

fn (arg1, arg2, ... , argn),

la notación usada en Lisp es la siguiente :

(fn arg1 arg2 ... argn).

Cada argumento es también alguna E-expresión a su vez y el intérprete de Lisp lo evalúa antes de pasar a la ejecución de las operaciones que componen la función fn.

Lisp presenta funciones estándar como por ejemplo : PLUS, TIMES, DEFUN, COND, EQ, MEMBER, CAR, CDR y SUB1; su estructura, uso de sus argumentos y valor resultante son parte de la definición del lenguaje.

Las S-expresiones de Lisp son atómicas o compuestas. La s-expresión atómica (el átomo) representa algún valor indivisible y tiene la forma de un valor simbólico (textual) o de un número. La forma más común de la S-expresión compuesta es una lista que se forma encerrando entre paréntesis cualquier secuencia de S-expresiones separadas entre sí por uno o más espacios.

#### Ejemplos de S-expresiones :

ALFA X3 UNATOMOBIENLARGO	átomos simbólicos.
459 -25 46.7 3.5E-2	átomos numéricos.
( P A T O )	lista con elementos P,A,T,O.
(UNAM)	lista con un sólo elemento.
(P (G P) (V (H) G))	lista con elementos P , (G P), (V (H) G).
() o sea NIL	la lista vacía.

Los valores lógicos se expresan usando el átomo NIL para la falsedad y el átomo T (o bien cualquier S-expresión diferente de NIL) para la certidumbre.

Las operaciones elementales con S-expresiones se realizan usando las siguientes funciones de base :

### selectores

- (CAR s)           regresa el primer elemento de la lista s.
- (CDR s)           regresa lo que queda de la lista s si se quitó su primer elemento.

### constructor

- (CONS e s)       regresa la lista con el primer elemento e y el resto s.

### predicados

- (ATOM a)         regresa T si s es un átomo, si no lo que regresa es NIL.
- (EQ x y)         regresa T si x,y son dos átomos idénticos y NIL si son dos átomos diferentes (exige átomos como argumentos).

### Ejemplos de uso de funciones de base :

- |                  |                   |
|------------------|-------------------|
| (CAR '(A B C))   | da valor A.       |
| (CDR '(A B C))   | da valor (B C).   |
| (CONS 'A '(B C)) | da valor (A B C). |
| (ATOM 'A)        | da valor T.       |
| (ATOM '(A B C))  | da valor NIL.     |
| (EQ 'A 'B)       | da valor NIL.     |

El uso incorrecto de alguna función provoca en general algún mensaje de error del sistema. Las funciones se pueden usar de

manera compuesta, por ejemplo :

(CONS (ATOM '(A)) (CDR '(A B))) da valor (NIL B).

(CAR (CDR '(A B C))) da valor B.

Para que podamos aclarar el papel del apóstrofe en los ejemplos presentados anteriormente hay que especificar antes y con más detalle las formas posibles de E-expresiones y el proceso de su evaluación por el intérprete de Lisp. Se distinguen los casos siguientes :

- a) E-expresión es un átomo numérico : su evaluación da el valor numérico correspondiente.
- b) E-expresión es un átomo simbólico : este átomo se interpreta como una variable y su evaluación da el valor que está actualmente asociado con la variable (o bien se reporta un error si no hay valor asociado);
- c) E-expresión tiene la forma compuesta

(fn arg1 arge2 ... argn)

donde el nombre de la función fn indica que se exige un tratamiento especial de los argumentos (por eso se le conoce con el nombre de forma especial). El tratamiento especial puede ser que no se evalúe o que lo realice de algún modo específico.

Un uso muy frecuente tiene, por ejemplo, la forma especial QUOTE que expresa las constantes simbólicas en las E-expresiones. El valor de la expresión

(QUOTE s)

es precisamente la S-expresión s sin que se evalúe, o sea QUOTE no evalúa su argumento y lo regresa como tal. Por razones notacionales se permite usar el apóstrofe para abreviar la forma QUOTE, así que:

(QUOTE (A B C)) equivale a '(A B C)

y se evalúa a (A B C). Con esto se aclara el uso de apóstrofe en

los ejemplos citados.

En el caso b) mencioné los valores de variables, y la pregunta obligada es ¿ cómo se asocia un valor a una variable ?. El caso principal es la asociación del valor de argumento con el parámetro formal de la función durante su ejecución, lo que asegura el mismo intérprete de Lisp.

Lisp tiene también una manera explícita de asignar valor a una variable usando la forma especial SETQ :

(SETQ var expr)

El argumento var tiene que ser un átomo simbólico al que se le asigna el valor del segundo argumento (evaluado) expr. El valor de toda la forma (SETQ...) es aquel de su segundo argumento, pero en la mayoría de los casos se usa solamente por su efecto secundario: la asignación de este valor a variable, que se queda vigente hasta que ocurra algún cambio de valor de la misma variable. Este efecto se ve en el siguiente ejemplo, en el cual se presenta en negrillas las instrucciones y en tono normal la respuesta del sistema, los comentarios de la derecha no forman parte del programa :

--> (SETQ A 1) 1	Se asigna el valor de 1 a A
--> (SETQ B 2) 2	Se asigna el valor de 2 a B
--> (SETQ A (PLUS A B)) 3	Cambio de valor de A
--> (CONS A'(A B)) (3 A B)	Los argumentos de CONS tienen los valores 3 y (A B).

Respecto a la definición del lenguaje Lisp, ya hemos presentado (en forma muy simplificada) la estructura de S-expresiones, de E-expresiones y el modo de evaluar las E-expresiones. El resto del lenguaje corresponde a la definición de sus funciones estándar, especificando para cada una de ellas la estructura y carácter de sus argumentos, el resultado y, en casos particulares, también la forma de evaluación de la función y su efecto secundario.

Tomando en cuenta que una implementación de Lisp contiene hasta 200 o más funciones estándar, se presentará a continuación sólo una breve descripción de las funciones más importantes, por su

uso general dentro de la programación. Las formas que se usan en Lisp para expresar lo que en lenguajes convencionales es llamado "estructura de control" son básicamente:

- \* la composición de funciones que expresan operaciones compuestas y con cierta reserva también secuencias de operaciones.
- \* la definición recursiva que expresa las operaciones cíclicas o interactivas.
- \* el uso de la forma especial COND que corresponde a la estructura IF-THEN-ELSE.
- \* el uso de la forma especial PROG que permite la programación interactiva directa.

La E-expresión formada por COND tiene el patrón general

(COND (c<sub>1</sub> v<sub>1</sub>) (c<sub>2</sub> v<sub>2</sub>) ... (c<sub>n</sub> v<sub>n</sub>))

en el que c<sub>i</sub> y v<sub>i</sub> son E-expresiones. La semántica de COND (o sea, su modo de evaluación) se describe mediante la construcción IF-THEN-ELSE como :

```
IF c1 THEN v1
    ELSE IF c2 THEN v2
        ELSE IF
            ...
            ELSE IF cn THEN vn.
```

Dicho de otra manera : el valor de COND es el de la primera v<sub>i</sub> cuya c<sub>i</sub> correspondiente tiene el valor diferente de NIL; si ninguna c<sub>i</sub> lo tiene, se reporta error.

La E-expresión formada por PROG se puede esquematizar así:

(PROG (var<sub>1</sub> var<sub>2</sub> ... var<sub>k</sub>) e<sub>1</sub> e<sub>2</sub> ... e<sub>n</sub>)

donde var<sub>i</sub> son átomos simbólicos (variables locales de PROG) y e<sub>i</sub> son E-expresiones compuestas. En la evaluación de PROG las variables var<sub>i</sub> se asocian primeramente con el valor NIL, y luego se evalúan las E-expresiones e<sub>1</sub>, e<sub>2</sub>, ... e<sub>n</sub> de manera secuencial. El valor de PROG es el del argumento x de la primera expresión (RETURN x) que se evalúe dentro de las expresiones e<sub>1</sub>, e<sub>2</sub>, ... e<sub>n</sub> y con

ésta se termina la evaluación de PROG. Si se evalúan todas las e, sin encontrar algún RETURN, el resultado es NIL.

PROG permite también el uso de (GO eti<sub>q</sub>) que transfiere la ejecución al lugar determinado por eti<sub>q</sub> (un átomo simbólico que puede declararse como etiqueta poniéndolo al frente de alguna de las e<sub>i</sub>).

Ahora comentaremos la función BORREP: primero se pregunta si la lista L está vacía (las formas equivalentes de esta pregunta son (EQ L NIL) o bien (NULL L)). Si L no está vacía, usando la función estándar MEMBER se verifica si su primer elemento está contenido en el resto. En este caso se puede borrar, o sea el resultado se obtiene llamando otra vez a BORREP dándole solamente el resto de la lista L. Si no ha ocurrido ninguno de los dos casos, se borran las ocurrencias repetidas en el resto de la lista y al resultado se le añade el primer elemento.

A continuación ilustraremos las técnicas de programación en Lisp usando como ejemplo varias implantaciones de la función REVERSE, que invierte una lista (esta función en realidad se ofrece como una de las funciones estándar de Lisp). Primero presentamos la definición recursiva usando las funciones estándar APPEND y LIST :

```
(DEFUN REVERSE (L)
  (COND ((NULL L) NIL)
        (T (APPEND (REVERSE (CDR L))
                   (LIST (CAR L)))))
  ))
```

La lista vacía invertida es también vacía -esto dice la primera variante de COND. Si la lista L no está vacía, primero se le añade al final el primer elemento. Para añadirlo usamos la función APPEND, que exige como argumentos dos listas - por eso hay que hacer la lista que contiene al primer elemento, y esto se realiza llamando a la función LIST.

La implementación que acabamos de explicar es poco eficiente dado que la función APPEND siempre copia su primer argumento. Desarrollamos entonces otra implantación recursiva que no tenga este defecto :

```
(DEFUN REVERSE (L) (REV1 L NIL))
(DEFUN REV1 (L ACC)
  (COND ((NULL L) ACC)
        (T (REV1 (CDR L) (CONS (CAR L) ACC))))
  ))
```

La función auxiliar REV1 en cada paso de recursión, quita el primer elemento de la lista L y lo añade al inicio del "acumulador" ACC. Al vaciarse la lista L, el resultado está

guardado en ACC. La función REVERSE llama a REVI dándole el acumulador vacío.

A manera de conclusión presentamos una implementación iterativa de REVERSE que tiene muchos rasgos comunes con la función REVI :

```
(DEFUN REVERSE (L)
  (PROG (ACC)
    CICLO (COND ((NULL L) (RETURN ACC)))
          (SETQ AC (CONS (CAR L) ACC))
          (SETQ L (CDR L))
          (GO CICLO) ))
```

El valor inicial de ACC es NIL y en cada paso por el ciclo se le añade al inicio un nuevo elemento de la lista L. El uso de L merece una observación: dado que los parámetros en Lisp se llaman "por valor" L sirve al mismo tiempo como una variable local.

Las desventajas principales de Lisp<sup>85</sup> podrían ser :

1. **Sintaxis desagradable.** Los únicos elementos de las sintaxis son los separadores (espacios y paréntesis). Un programa en Lisp puede ser muy difícil de leer si no está bien indentado.
2. **Ausencia de tipos de datos.** No es posible detectar errores relacionados con el tipo de datos durante la ejecución de un programa.
3. **Ineficiencia.** Cualquier programa interpretado es más lento que si se ha compilado. Sin embargo existen compiladores de Lisp que aceleran la ejecución de programas (aunque se pierden las ventajas de flexibilidad, interactividad, etc.)
4. **Falta de un standard.** Existen muchas versiones de Lisp. Sólo en los últimos tres años se ha empezado a adoptar a COMMON Lisp como el standard.

## 8.2 Prolog.

Prolog surge a principios de los setentas como un

---

<sup>85</sup> Para una introducción más detallada de LISP véase los siguientes libros ;  
[SKIL,76], [WILE,84] y [WINS,81].

esfuerzo conjunto entre la Universidad de Marsella<sup>86</sup> y la Universidad de Edimburgo<sup>87</sup>. Este lenguaje, que es una herramienta muy poderosa para la resolución de problemas, es el lenguaje oficial del proyecto japonés de la Quinta generación de computadoras.

Prolog permite en una forma muy conveniente expresar y solucionar problemas que incluyen objetos y relaciones entre ellos. En este lenguaje la programación consiste básicamente en la especificación de ciertas relaciones de manera que permita al sistema "razonar" sobre ellas.

Los datos con los que se trabaja se pueden caracterizar en general como datos simbólicos estructuralizados en forma de árboles y el mecanismo de procesamiento se basa en emparejamiento de patrones y retroceso automático.

Prolog es un lenguaje práctico que se ha caracterizado por la forma declarativa de su programación, esto implica pensar de modo completamente diferente respecto de los problemas a solucionar, lo que parece difícil al principio, pero después de acostumbrarse resulta muy natural.

En Prolog, al igual que en Lisp, los programas se ejecutan de manera interpretativa e interactiva. Un programa en Prolog está formado por una secuencia de cláusulas que son de tres tipos :

- \* hechos, que declaran las cosas que son incondicionalmente ciertas;
- \* reglas, especifican qué cosas se pueden deducir si se cumplen ciertas condiciones;
- \* preguntas, permiten que el usuario pida alguna información derivable de los hechos y reglas del programa.

Sintácticamente<sup>88</sup>, las cláusulas se definen por el patrón común :

---

86 (COLM, 78)  
(ROUS, 75)

87 (KOMA, 74)  
(KOMA, 79)

88 La sintaxis básica del lenguaje está basada en la estructura de las cláusulas de Horn, las cuales son un caso particular de la forma clausal general de la lógica de predicados de 1er orden.



cabeza :- meta<sub>1</sub>, meta<sub>2</sub>, ... , meta<sub>n</sub>.

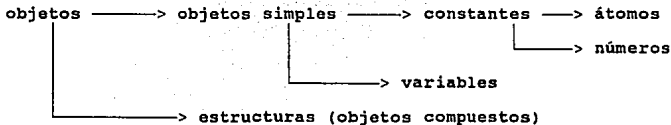
donde cabeza, meta<sub>1</sub>, ... , meta<sub>n</sub> tienen la forma de los llamados términos. Los hechos tienen solamente la cabeza, las preguntas solamente el cuerpo (que es la parte formada por las metas) y las reglas tienen cabeza y cuerpo no vacíos.

A continuación se presenta un ejemplo de una sesión con el sistema Prolog:

```
mas-grande (albin,adria).      % Se anuncia que Albin es más
mas-grande (adria,paty).      % grande que Adria. Adria es
mas-grande (adria,orie).      % más grande que Paty y más
                              % grande que Orie.
                              % Estos son los hechos.
respeta (A, B).              % Regla de respeto - A respeta
mas-grande (B,A).            % a B si B más grande que A.
respeta (A, B) :-            % Otra regla - A respeta a B
mas-grande (C,A)              % si hay C más grande que A
respeta (C, B).              % tal, que respeta a B.

?- respeta (orie, adria).     % Ahora preguntamos y el
  Yes                          % sistema responde.
?- respeta (X, orie)          % ¿ Hay alguien que respeta a
  No                            % Orie ?
?- respeta (X, Y).            % ¿ Quién respeta a quién ?.
  X = adria,                   Y = albin;
  X = paty,                    Y = adria;
  X = orie,                    Y = adria;
  X = paty,                    Y = albin;
  X = orie,                    Y = albin;
```

En Prolog, los datos son de distintos tipos y se distinguen por su apariencia sintáctica. La clasificación es la siguiente :



Los átomos tienen la forma de un identificador que comienza con una letra minúscula, una cadena de caracteres encerrada entre apóstrofes o bien una cadena de caracteres especiales.

Los números en Prolog se expresan en notación decimal. En la mayoría de las implementaciones de Prolog sólo se utilizan los números enteros.

Las variables tienen la forma de un identificador que comienza con una letra mayúscula o con el carácter de subrayado ('\_').

Las estructuras son de la forma general

$$f(c_1, c_2, \dots, c_n)$$

donde  $f$  denota algún functor (formado según la regla de los átomos) y los  $c_i$  son las componentes, que pueden ser cualquier objeto a su vez. El functor de la estructura puede entenderse como nombre de alguna relación que se cumple entre las componentes.

#### Ejemplos de objetos :

alfa x, 'Patricia'  
123 0 -1  
X Result  
fecha (6, feb, 1993)

átomos  
números  
variables  
estructura con el functor

fecha y componentes 6, feb y  
1993

Los objetos en Prolog sin distinguir su tipo se denotan como términos. La operación más importante que hace Prolog con términos es su empatamiento o equiparamiento (matching). Se dice que dos términos son iguales si :

- A) son idénticos o
- B) las variables en los dos términos se pueden asociar con algunos objetos de tal modo que, después de la sustitución de las variables por estos objetos, los dos términos resultan idénticos.

Durante el proceso de equiparamiento Prolog intenta descubrir si dos términos pueden equiparar : en caso negativo, se dice que el equiparamiento falló; en caso contrario, decimos que

tuvo éxito y las variables en los dos términos se asocian con valores que conducen a la identidad. La solicitud explícita de equiparamiento se expresa usando el operador '≡', así que es posible la pregunta :

?- fecha (Día, nov, Año) = fecha (D, Mes, 1993).

El empatamiento de estos términos tiene éxito dado que hay varias sustituciones posibles que lo hacen idénticos, p. e. :

Día = 6, D = 6, Mes = feb, Año = 1993

o bien

Día = seis, D = seis, Mes = feb, Año = 1993.

Lo que busca Prolog es la sustitución más general, que en nuestro ejemplo tiene la forma

Día = D, Mes = feb, Año = 1993.

Las reglas generales que usa Prolog en el proceso de equiparamiento de términos A y B son las siguientes :

1. Si A y B son constantes, equiparan si y sólo si son el mismo objeto.
2. Si A es una variable (sin valor asociado) y B es cualquier cosa, entonces equiparan y A se asocia con B. Algo semejante ocurre si B es una variable sin valor asociado.
3. Si A y B son estructuras, entonces equiparan sólo si :
  - \* las dos tienen el mismo functor y
  - \* sus componentes correspondientes equiparan.

La noción de equiparamiento es indispensable para la descripción del método que usa Prolog para contestar las preguntas. Ya sabemos que una pregunta en Prolog está formada por alguna secuencia de metas separadas entre sí por coma. Para contestar la pregunta, Prolog intenta satisfacer las metas, o sea demostrar que se cumplen (satisfacen) las relaciones expresadas en las metas suponiendo que se cumplen las relaciones representadas en la forma

de hechos y reglas del programa.

Si la pregunta contiene algunas variables, Prolog busca los objetos que se pueden sustituir por ellas para que satisfagan las metas y luego muestra los valores posibles al usuario. Si Prolog no encuentra ninguna asociación de variables que permita satisfacer las metas, su respuesta es 'no'.

Esto queda más claro en el siguiente ejemplo. Teniendo los hechos y reglas :

mas-grande (albin, adria).	(1)
mas-grande (adria, paty).	(2)
mas-grande (adria, orie).	(3)
respeta (A, B) :- mas-grande (B, A).	(4)
respeta (A, B) :- mas-grande (C, A), respeta (C, B).	(5)

quisiéramos saber a quién respeta orie y planteamos la pregunta

?-respeta (orie, Z)

Para Prolog esto significa buscar si se puede satisfacer la meta respeta (orie, Z), entonces busca en el programa la cláusula cuya cabeza equipara con la meta. Encuentra la regla (4), el equiparamiento causa las asociaciones

A = orie, B = Z

y la meta original se reemplaza por el cuerpo de la regla llegando así a la meta

mas-grande (B, orie). (6)

Prolog intenta satisfacer esta meta buscando otra vez en las cláusulas del programa. Encuentra el hecho (3), del equiparamiento resulta

B = adria

y dado que el cuerpo de hecho es vacío, la meta original queda satisfecha. La primera respuesta de Prolog es entonces

Z = adria.

Ahora Prolog busca alguna otra posibilidad en cuanto a cómo satisfacer la meta (6), pero ya no hay, entonces busca otra posibilidad para la meta original. Equipara con la cabeza de la regla (5) recibiendo asociaciones

$A = \text{orie}, B = Z$

y la secuencia de metas que forman el cuerpo de la regla

**mas-grande (C,orie), respeta (C, B). (7)**

Prolog intenta satisfacer la primera de las metas. Busca en las cláusulas, encontrando el hecho (3) con la asociación

$C = \text{adria}$

y dado que el cuerpo es vacío la primera meta queda satisfecha. Ahora hay que satisfacer la segunda meta de (7) que con la asociación pendiente adquirió la forma

**respeta (adria,B). (8)**

Se encuentra la regla (4), el empatamiento da

$A' = \text{adria}, B' = B$

y se intenta satisfacer el cuerpo de la regla (4) que ahora tiene la forma

**mas-grande (B', adria). (9)**

Esta meta se satisface con el hecho (1) causando la asociación

$B' = \text{albin}$

y ya no hay más metas que satisfacer. Entonces la meta original queda satisfecha otra vez y, dadas las asociaciones pendientes

$$B = Z, B' = B, B' = \text{albin},$$

la segunda respuesta de Prolog es

$$Z = \text{albin}.$$

Prolog hace todavía otra prueba de cómo satisfacer la meta original tratando de satisfacer las metas intermedias en el orden inverso. La meta (9) ya no se puede resatisfacer, entonces checa la meta (8). La equipara esta vez con la cabeza de la regla (5), recibe las asociaciones

$$A'' = \text{adria}, B'' = B$$

e intenta satisfacer su cuerpo que adquirió la forma

$$\text{mas-grande (C, adria), respeta (C, B'')}. \quad (10)$$

La primera de las metas se satisface usando el hecho (1) causando la asociación

$$C = \text{albin}$$

y queda por satisfacer la segunda meta que es ahora

$$\text{respeta (albin, B'')}. \quad (11)$$

Se hace el intento de satisfacer la meta (11) usando la regla (4), pero su cuerpo es entonces

$$\text{mas-grande (B'', albin)} \quad (12)$$

que falla. En otro intento usa la regla (5) causando las asociaciones

$$A''' = \text{albin}, B''' = B''$$

y presentando para el próximo paso las metas

mas-grande (C, albin) , respeta (C, B'''). (13)

La primera de estas metas falla, entonces falla otra vez la meta (11) y dado que no hay otras reglas para la relación respeta, falla también la secuencia de metas (10). Pero esta fue tomada por Prolog como la otra (y última) posibilidad de cómo satisfacer la meta original, entonces ya no hay más posibilidades.

En estos ejemplos hemos usado pocas veces las herramientas de Prolog, que consisten básicamente en unas decenas de relaciones estándar. A continuación veremos algunas de ellas :

La lista en Prolog se expresa encerrando en corchetes sus elementos separados entre sí por coma. Así p. e.

[] es la lista vacía  
[a, b, c] es la lista de elementos a, b y c  
[a, f(b,X), [c,d]] es otra lista.

La utilización de corchetes es una extensión sintáctica de Prolog, la representación interna de listas usa un functor estándar que consideramos indispensable introducir.

Cada lista no vacía tiene un primer elemento (la cabeza) y un resto que es también una lista. Prolog ofrece otra extensión notacional para expresar la cabeza y el resto de la lista; se hace de la forma

[ Cabeza : Resto ].

así podemos dar varias apariencias a la misma lista, p.e

[a,b,c] = [a:[b,c]] = [a,b:[c]] = [a,b,c:[]]

Para demostrar el uso de listas presentamos dos ejemplos de relaciones simples. Primero implantamos una relación member de dos argumentos que sirve para la misma cosa que la función member de Lisp (con ciertos detalles diferentes que se aclararán enseguida):

member (X, [X: \_]).  
member (X, [\_;Resto]) :- member (X,Resto).

Si el elemento X esta en la cabeza de la lista, member queda satisfecha -esto dice la primera cláusula como un hecho. El símbolo '\_' representa la llamada variable anónima de Prolog y se usa para equiparar cualquier cosa que no es importante en el resto de la cláusula.

La otra posibilidad respecto de cómo satisfacer member, es quitar el primer elemento de la lista (cualquiera que éste sea) y buscar entonces al elemento X en el resto de la lista.

A diferencia de la función member de Lisp, la implantación en Prolog permite también el uso poco convencional que veremos en seguida :

```
?-member (A, [a, b, c]).  
  A = a;  
  A = b;  
  A = c
```

El retroceso automático recorrió la lista [a,b,c] buscando las asociaciones posibles de la variable A que satisfacen la meta.

La relación reverse, realiza la operación de invertir una lista :

```
reverse (L1, L2) :- REVI (L1, [], L2).  
rev1 ([], L, L).  
rev1 ([X:R1], L1, L2) :- rev1 (R1, [X:L1], L2).
```

Esta definición corresponde al método usado para la segunda variante de la función REVERSE en Lisp y presentamos solamente en forma esquemática el modo de trabajo de Prolog para contestar la pregunta :

```
?- reverse ([a,b,c], X).
```

Se investiga la siguiente secuencia de metas

```
reverse ([a,b,c], X)  
  rev1 ([a,b,c], [], X)  
    rev1 ([b,c], [a], X)
```



```

rev1 ([c], [b,a], X)
rev1 ([], [c,b,a], X)
rev1 ([], [c,b,a], [c,b,a])

```

X = [c,b,a].

Por último comentaremos la función BORREP :

```
borrep ([], []).
```

```

borrep ([E:Resto], Result) :-
    member (E, Resto), !,
    borrep (Resto, Result).
borrep ([E:Resto], Result) :- borrep (Resto,
    Result).

```

```

?- borrep ([a,b,c,a,c,b,d,a,b,e], X).
X = [c, d, a, b, e].

```

```
borrep ([], []).
```

La primera cláusula dice que para la lista vacía el resultado es también vacío. La segunda cláusula se usa si el primer elemento de la lista también está en el resto (lo cual checka member) - entonces se quita y el resto se obtiene procesando el resto. La tercera cláusula se usa en el caso contrario - si el primer elemento no está en el resto; se calcula el resultado del resto y se le añade este elemento al inicio.

El signo de exclamación ( '!', llamado "el corte"<sup>89</sup> ) en la segunda cláusula determina que una vez satisfecha la meta

```
member (E, Resto)
```

el retroceso eventual ya no se usará la tercera cláusula como alternativa para satisfacer borrep - o sea de las dos reglas siempre se usará solamente una.

---

<sup>89</sup> La descripción detallada del efecto de corte rebasa los márgenes de nuestro trabajo; para profundizar en el tema consúltense [CLOC,81].

Quisiera enumerar brevemente, a manera de resumen algunas de las características de este lenguaje :

1. Semántica declarativa heredada de la lógica además de la semántica de procedimientos común.
2. Identidad de forma de programa y datos. Las cláusulas pueden emplearse para representar datos y pueden manipularse como términos por intérpretes escritos en Prolog.
3. Los argumentos de entrada y salida de los procedimientos no tiene que ser definidos de antemano y pueden tener distintos significados de llamada en llamada, esto es, los procedimientos son multi-propósito.
4. Los procedimientos pueden tener múltiples salidas y entradas.
5. Los procedimientos pueden generar, mediante el "backtracking" una secuencia de resultados alternativos, permitiendo con esto hacer iteraciones.
6. Los términos permiten definir una estructura de registros general con un número arbitrario de campos. El número de registros posibles sólo está limitado por la memoria del sistema y no hay restricción sobre los tipos de los campos.
7. El casamiento de patrones que tiene Prolog evita el uso de funciones selectoras y construcciones para operar sobre estructuras de datos.
8. Las estructuras de datos pueden quedar incompletas, es decir, contener variables libres, las cuales pueden ser instanciadas más tarde por otros procedimientos.
9. Prolog por ser un lenguaje declarativo, puede prescindir de instrucciones como "go to", "do for", "while-do-loop", asignaciones y referencias (apuntadores). Tales funciones son indispensables para los lenguajes de procedimientos o algorítmicos.

10. La semántica de procedimientos de un programa sintácticamente bien escrito se encuentra totalmente definida. Es imposible que se presente una condición de error o que una operación no definida se ejecute. Esta semántica definida totalmente asegura que los errores de programación no producirán comportamientos extraños o mensajes de error incomprensibles.
11. Los programas son independientes del hardware de la computadora que se esté utilizando.

## CONCLUSIONES Y PERSPECTIVAS

En la elaboración de este trabajo encontramos algunos detalles interesantes. Comenzando con el problema de no existir una definición de inteligencia, y por lo tanto de Inteligencia Artificial; hecho que motivó una discusión, con la cual se concluyó que la Inteligencia Artificial es la solución de problemas, con el apoyo de la computadora, mediante la aplicación de procesos que son análogos al proceso de razonamiento humano.

El estudio de la inteligencia se ha llevado a cabo desde diferentes puntos de vista, cada uno de ellos ha aportado una cantidad enorme de conocimiento. Aunque me parece que los ingredientes de las teorías que vimos han sido en general demasiado minuciosos, locales e inestructurados como para dar cuenta, práctica y fenomenológicamente, de la efectividad del sentido común. Alguien ha dicho que "el menos común de los sentidos es el sentido común", refiriéndose al hecho de que las decisiones personales no siempre son adecuadas según los criterios de otras personas, o de acuerdo con las consecuencias directas de llevarlas a cabo. Estas son características importantes del hombre: su manera de razonar es individual y siempre existe la posibilidad de que cometa errores de juicio. Para poder construir computadoras que emulen ciertas capacidades mentales del hombre como el sentido común, es de importancia primordial entender de qué manera se logra la organización de la información en el cerebro humano, y conocer la forma en que se lleva a cabo este razonamiento.

El razonamiento cualitativo realiza una abstracción de un fenómeno físico en donde se consideran únicamente conjuntos finitos de posibles valores para las variables involucradas. Constituye, en consecuencia una forma de razonamiento aproximado. A diferencia de los otros enfoques de este tipo de razonamiento, en él no existen propagaciones de incertidumbre, sean de probabilidades o de certezas. La aproximación que hace se refiere a que se considera en él sólo a las propiedades más generales de un fenómeno y a éstas sólo con la posibilidad de tomar un número finito de posibles valores. Los modelos resultantes son de una complejidad computacional relativamente sencilla, pues con ellos la simulación del fenómeno se hace mediante evaluaciones puramente tabulares. Sin embargo, la construcción de tales modelos requiere de un sólido conocimiento del problema a tratar.

Entre las principales ventajas de las técnicas que hemos presentado podemos resaltar la simplicidad de los procedimientos de simulación y explicación por medio de los modelos construidos así

como la no necesidad de realizar cálculos numéricos complejos como los que conllevan los modelos analíticos. Por otro lado, la interpretación de resultados es directa e inherente al proceso de simulación, lo cual se contrapone con las dificultades que se presentan al tratar de distinguir propiedades cualitativas de los modelos analíticos. Los procedimientos de simulación de este enfoque son fácilmente formalizables a partir del modelo construido, mediante un conjunto de reglas sintácticas sencillas, que constituyen en sí una gramática regular y por tanto son fácilmente implementables en un sistema que adquiriría así un conocimiento más intuitivo del fenómeno modelado. Por decirlo en otras palabras, la heurística del fenómeno, se describe como una expresión regular.

Sin embargo, estas técnicas cuentan también con grandes desventajas. Las que resaltan inmediatamente son las siguientes: La formulación de confluencias requiere de una comprensión extensa del fenómeno a modelar. Cada confluencia describe el comportamiento de unas pocas variables en el fenómeno, por lo que una descripción más completa del fenómeno incrementa el número de confluencias y de variables que aparecen en ellas. Esto ocasiona que el autómata o espacio de estados que lo modela crezca sustancialmente cuando se refina el conocimiento del modelo. Por otro lado, en el aspecto formal, la Física Intuitiva no cuenta aún con una caracterización sólida y menos aún con una sólida fundamentación. El uso de los valores representativos y su manipulación algebraica han motivado la aparición de un así llamado **Cálculo Cualitativo**, el cual pretende formalizar el manejo de valores que se hace en el razonamiento cualitativo, pero que, debido esencialmente a que maneja sólo una cantidad finita de posibles valores, se tienen problemas insalvables para generalizarlo. Bien que en él se manejan las principales características variacionales de funciones se pierden las propiedades algebraicas de los conjuntos de valores y la manipulación de valores se restringe a evaluaciones de funciones sin reglas algebraicas de transformación.

Por otra parte, los sistemas tradicionales basados en lógica de predicados son monótonos en el sentido de que el número de enunciados que se saben verdaderas se incrementa estrictamente en el transcurso del tiempo. Pueden añadirse nuevos enunciados al sistema y pueden demostrarse nuevos teoremas, pero ninguno de estos acontecimientos invalidará nunca una sentencia demostrada o conocida previamente. Existen diversas ventajas al trabajar con un sistema como éste, incluyendo:

\* Cuando se añade un nuevo enunciado al sistema no es necesario realizar ninguna comprobación para ver si existen inconsistencias entre el nuevo enunciado y el conocimiento antiguo.

\* No es necesario recordar, para cada enunciado que se ha

demostrado, la lista de los otros enunciados en las que se basa la demostración, puesto que no hay ningún peligro en que dichos enunciados desaparezcan.

Desafortunadamente, tales sistemas monótonos no son muy buenos trabajando con la presencia de incertidumbre. Así, nos vemos forzados a preguntarnos cómo podemos expandir nuestro sistema de lógica monótona para permitir el razonamiento no monótono.<sup>90</sup>

Es más difícil tratar con sistemas no monótonos que con los monótonos, puesto que a menudo es necesario, al eliminar un enunciado de la base de conocimientos, regresar a través de otros enunciados cuyas demostraciones dependen del enunciado eliminado, y, o bien eliminarlas, o bien encontrar nuevas demostraciones que sean válidas respecto a la base de conocimientos actual. La supresión de un único enunciado puede tener un efecto importante sobre una base entera de conocimiento, puesto que todas las demostraciones que dependen de él deben desecharse, y a continuación deben descartarse todas las demostraciones que dependen de los enunciados cuyas demostraciones se han descartado (y para las cuales no pueden hallarse otras demostraciones), y así sucesivamente. Al diseñar sistemas no monótonos es importante intentar asegurar que el sistema no gaste todo su tiempo propagando cambios arriba y abajo.

Para poder propagar cambios en la base de datos y comprobar la validez actual de las demostraciones, es importante almacenar, junto con cada teorema, su demostración, o como mínimo, una lista de las otras sentencias de las que depende la demostración. Esto no es necesario en los sistemas monótonos puesto que una vez hallada la demostración, nunca necesita ser reexaminada. Así, los sistemas no monótonos pueden requerir más espacio de almacenamiento, junto con más tiempo de proceso, que los monótonos.<sup>91</sup>

Por último comentaré que en la gran familia que forma la gente ligada a la computación se oye la pregunta: ¿Qué lenguaje es mejor para esta u otra cosa?". En el campo de la Inteligencia Artificial la pregunta a veces se reduce a la simple (!) selección entre LISP y PROLOG.

En este trabajo no se puso la meta de contestar esta pregunta, que además considero incorrecta si no se aclara que LISP y PROLOG se están tomando en consideración.

---

<sup>90</sup> Hay varios artículos que describen diversos aspectos del razonamiento no monótono en Artificial Intelligence de abril de 1980, que fue un número especial dedicado a este tema.

<sup>91</sup> Para un tratamiento más formal de la lógica no monótona ver (McDE,80).

Basándose en las características generales de los dos lenguajes se puede concluir que LISP permite diseñar las estructuras de datos y los algoritmos que los manejan de una manera muy eficiente. PROLOG a su vez tiene muy buen poder de expresión y entonces permite una formulación bastante natural y directa de la solución de problemas. ¿Cuál de las dos cosas es más importante?. Esto lo tiene que decidir el mismo usuario con respecto al carácter del problema a resolver.

Sin embargo, ambos lenguajes ya han demostrado su gran utilidad y la familiarización con los dos es indispensable para cualquier persona que piense comprometerse en algún trabajo serio en Inteligencia Artificial.

## BIBLIOGRAFIA

### CAPITULO I

- [AUGA,84]  
Augarten, S., "Bit by Bit. An Illustrated History of Computers", Ticknor & Fields, Nueva York, 1984.
- [BINF,87]  
Binford, Thomas O., "La Máquina Ve", en Robótica de Marvin Minsky, Edit. Planeta, México, 1987, pag. 77.
- [COLB,75]  
Colby, K., "Artificial Paranoia", Pergamon Press, Nueva York, 1975.
- [DUDA,79]  
Duda, R.O., P.E. Hart, K. Konolige & R. Reboh, "A Computer-Based Consultant for Mineral Exploration", Technical Report, SRI International, Sep. 1979.
- [ERMA,80]  
Erman, L.D., F. Hayes-Roth, V.R. Lesser & D.R. Reddy, "The HEARSAY-III Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty", Computing Surveys, Vol. 12 No. 2, Junio 1980.
- [FOGE,83]  
Fogel, Jean-Francois, "La Perceé de L'intelligence Artificielle", Le Point, 31-x-83, pags. 74-80.
- [GREE,69]  
Green, C., "Application of Theorem Proving to Problem Solving" en IJCAT 1, 1969.
- [HEIM,80]  
Helms, Steve J., "John Von Neumann and Robert Wiener, from Mathematics to the Technologies of Life and Death", The MIT Press, Cambridge, Massachussetts, 1980.
- [HEPP,87]  
Heppenheimer, T.A., "El Hombre Fabrica un Hombre", en Robótica de Marvin Minsky, Edit. Planeta, México, 1987, pag 25.



- [HYMA,81]  
Hyman, "Charles Babbage, Pioneer of the Computer",  
Princeton University Press, Nueva Jersey, 1981.
- [KULI,80]  
Kulikowki, C.A., "Artificial Intelligence Methods and  
Systems for Medical Consultation", IEEE Transactions on  
Pattern Analysis and Machine Intelligence, Sep. 1980.
- [LENA,83]  
Lenat, D.B., "Eurisko : a program that learns new  
heuristics and domain concepts. The natural of heuristics  
III: program design and results", Artificial  
Intelligence, Marzo, 1983.
- [LIND,80]  
Lindsay, R.K., B.G. Buchanan, E.A. Feigenbaum & L.  
Lederbeg, "Applications of Artificial Intelligence for  
Organic Chemistry : The Dendral Project", McGraw-Hill,  
Nueva York, 1980.
- [LOVE,61]  
Lovelace, A., "Notes upon L.F. Menabrea's Sketch of the  
Analytical Engine Invented by Charles Babbage", en  
Charles Babbage and his Calculations Engines, P. Morrison  
& E. Morrison (Edits), Dover, Nueva York, 1961.
- [MACS,77]  
Macsyma Group, "Macysma Reference Manual", Technical  
Report, MIT, 1977.
- [MARR,82]  
Marr, D., "Vision", Freeman, San Francisco, 1982.
- [MONT,86]  
Montseny Masip, Eduard, "Inteligencia Artificial y  
Computadores de la 5ta. Generación", en Biblioteca de  
Electrónica/informática, Edit. Marcombo, Barcelona, 1986,  
pag 9.
- [MOTO,81]  
Moto-Oka, T., "Fifth Generation Computer Systems",  
Proceedings of the Internacional Conference on Fifth  
Generation Computer Systems, North-Holland Publishing  
company (Edits), Tokio, Japon, Octubre 19-22, 1981.
- [NEWE,63]  
Newell, A., J.C. Shaw & H.A. Simon, "Empirical  
Explorations with the Logic Theory Machine : a case study  
in heuristics", en Computers and Thought, E.A. Feigenbaum  
& J. Feldman (Edits), McGraw-Hill, Nueva York, 1963.

- [NEWE,63A]  
Newell, A. & H. Simon, "GPS, a Program that Simulates Human Thought", en Computers and Thought, E.A. Feigenbaum & J. Feldman (Edit), McGraw-Hill, Nueva York, 1963.
- [POPL,76]  
Pople, H., "Presentation of The INTERNIST System", Proceeding of the AIM Workshop, 1976.
- [ROBE,77]  
Roberts, R.B. & I.P. Goldstein, "The FRL Manual", Technical Report, MIT Artificial Intelligence Laboratory, 1977.
- [SAMN,63]  
Samuel, A.L., "Some Studies in Machins Learning using the Game of Checkers", in Computers and Thought, E.A. Feigenbaum & J. Feldman (Edit), McGraw-Hill, Nueva York, 1963.
- [SHAN,50]  
Shannon, C. E., "Programming a Computer for Playing Chess", Philosophical Magazine, Series 7, Vol. 41, 1950.
- [SHOR,76]  
Shortliffe, E.H., "Computer-based Medical Consultations: MYCIN", Elsevier, Nueva York, 1976.
- [STEF,81]  
Stefik, M., "Planning with Constraints; MOLGEN : part 1", Artificial Intelligence, Vol. 16 No. 2, pp. 111-139, 1981.
- [STEF,81A]  
Stefik, M., "Planning with Constraints; MOLGEN : part 2", Artificial Intelligence, Vol. 16 No. 2, pp. 141-169, 1981.
- [STEF,82]  
Stefik, M., J. Alkins, R. Balzer, J. Benoit, L. Birnbaum, F. Hayes-Roth & E. Sacerdoti, "The Organization of Expert Systems", Artificial Intelligence, Vol. 18, pp 135-173, Marzo 1982.
- [TURI,63]  
Turing, A., "Computing Machinery and Intelligence", en Computers and Thought, E. A Feingenbaum & J. Feldman (Edits), McGraw-Hill, Nueva York, 1963.

- [WHIT,50]  
Whitehead, A.N. & B. Russell, "Principia Mathematica",  
2nd Ed, Cambridge University Press, Cambridge,1950.
- [WEIZ,66]  
Weizenbaum, J., "Eliza - a Computer Program for the Study  
of Natural Language Communication between Man and  
Machine", Communications of the ACM, Vol 9, No. 1.
- [WINO,72]  
Winograd, T., "Understanding Natural Language", Academic  
Press, Nueva York, 1972.
- [WINO,73]  
Winograd, T., "A Procedural Model of Language  
Understanding" en Computer Models of Thought and  
Language, R.C. Schank & K.M. Colby (Edits), Freeman, San  
Francisco, 1973.
- [WINS,75]  
Winston, P.H., "The Psychology of Computer Vision",  
McGraw-Hill, Nueva York, 1975.
- [WHIT,50]  
Whitehead, A.N. & B. Russell, "Principia Mathematica",  
2nd Ed, Cambridge University Press, Cambridge,1950.

## CAPITULO II

- [BUEH,62]  
Buehler, C., "Psychologie im Leben Unserer Zeit",  
Munchen/ Zurich, 1962.
- [FISC,75]  
Fischer, "Lexikon", Fischer Taschenbuch Verlag, Tomo 9,  
1975.
- [HOFS,57]  
Hofstatter, P. R., "Informations Theorie in  
Psychologie", (HG: P. R. Hofstatter), Frankfurt, 1957.
- [POPP,82]  
Popper, K. R./ Eccles, J. C., "Das Ich Und Sein Gehirn",  
2. Avflage, Munchen, 1982.
- [RITC,84]  
Ritchie, D., "Gehirn Und Computer", Stuttgart, 1984.

- [SIMO,84]  
 Simons, G., "Sind Computer Lebendig ?", Stand und Zukunft der Computer Entwicklung, Munchen, 1984.
- [STEI,84]  
 Steinnacker, I., "Intelligente Maschine ?", in Artificial Intelligente Eine Einführung (HG: L. Ritchier/W. Stucky), Stuttgart, Seite 7 FF, 1984.
- [WEIZ,78]  
 Weizenbaum, Joseph, "La Frontera entre el Ordenador y la Mente", Ediciones Piramide (Madrid), 1978.
- [WEIZ,84]  
 Weizenbaum, Joseph, "Kurs auf den Eisberg", Zurich, 1984.

### CAPITULO III

- [ARBI,78]  
 Arbib, M.A., "Segmentation Schemas and Cooperative Computation", Studies in Mathematical Biology, Levin, S.A. (Edit.), The Mathematical Association of America, pp. 118-155.
- [ASHB,52]  
 Ashby, W.R., "Design for a Brain", Wiley, Nueva York, 1952.
- [MINS,63]  
 Minsky, M., "Steps toward Artificial Intelligence", in Computers and Thought, E.A. Feigenbaum & J. Feldman (Edits), McGraw-Hill, 1963.
- [MINS,75]  
 Minsky, M., "A Framework for Representing Knowledge", in Psychology of Computer Vision, P. Winston (Edit.), McGraw-Hill, 1975.
- [PIAG,71]  
 Piaget, J., "Biology and Knowledge", The University of Chicago Press, Chicago, 1971.
- [PIAG,76]  
 Piaget, J., "The Psychology of Intelligence", Littlefield, Adams & Co., Nueva Jersey, 1976.

#### CAPITULO IV

[ADAM,84]

Adams, J., "Probabilistic Reasoning and Certainty Factors", in B. Buchanan and E. Shortliffe (Eds.), Ruled-Based Expert Systems", Adison-Wesley, Menlo Park, Calif., 1984.

[BAND,83]

Bandler, W., "Representation and Manipulation of Knowledge in Fuzzy Expert Systems", Proceeding of the Workshop on Fuzzy Sets and Knowledge-based Systems, Queen Mary College, University of London, 1981.

[BUCH,83]

Buchanan, B. and R. Duda, "Principles of Rule-based Expert Systems", in M. Yovits (Eds.), Advanced in Computers, Vol. 22, Academic Press, New York, 1983.

[BUCH,84]

Buchanan, B. and E. Shortliffe (Eds.), "Rule-based Expert Systems", Adison-Wesley, Menlo Park, Calif., 1984.

[CHAP,84]

Chapa Vergara Sergio, "Arquitectura de Sistemas Expertos", Reporte Técnico, Serie Amarilla, CINVESTAV, Depto. de Ingeniería Eléctrica, México, 1984.

[CHOI,85]

Choi, T. Y., "Formal techniques for the specification, verification and construction of communication protocols", IEEE Communications Magazine, vol 23, no 10, Octubre, 1985.

[DAVI,77]

Davis, R. & J. J. King, "An Overview of Production Systems", in Machine Intelligence 8, E. Elcock & D. Michie (Eds.), Horwood, Chinchester, England, 1977.

[DENN,78]

Denning, Dennis and Qualitz, "Machines, Languages and Computation", Prentice-Hall, 1978.

[DOYL,79]

Doyle, J., "A Truth Maintenance Systems", Artificial Intelligence, Vol.12, No. 3, 1979.

[DUDA,76]

Duda, R., "Subjective Bayesian Methods for Rule-based Inference Systems", Stanford Research Institute, Menlo

Park, Calif., 1976.

[DUDA,79]

Duda, R., P. E. Hart, K. Konolige & R. Reboh, "A Computer-based Consultant for Mineral Exploration", Technical Report, SRI International, Sept. 1979.

[DUDA,80]

Duda, R., J. Gaschnig and P. Hart, "Model Design in the PROSPECTOR Consultant System for Mineral Exploration", in D. Michie (Ed.), "Expert Systems in the Microelectronic Age", Edinburgh University Press, Edinburgh, 1980.

[FORB,84]

Forbus, "Qualitative Process Theory", Artificial Intelligence 24, 1984, pp. 85-168.

[GINZ,69]

Ginzburg, "Algebraic Theory of Automata", John Wiley and Sons, Nueva York, 1969.

[GUPT,82]

Gupta, M. and E. Sanchez (Eds.), "Approximate Reasoning in Decision Analysis", North-Holland, Amsterdam, 1982.

[JOHN,85]

Johnson R., "Bell Mints First Artificial Intelligence Chip; Engine Uses Fuzzy Logic", Electronic Engineering Times, Dec. 16, 1985.

[KAND,86]

Kandel, A., "Fuzzy Mathematical Techniques with Applications", Addison-Wesley, Reading, Mass., 1986.

[KAUF,75]

Kaufmann, A., "Introduction to the Theory of Fuzzy Subsets", Vol. 1, Academic Press, New York, 1975.

[KLEE,84]

Kleer and Brown, "A qualitative physics based on confluences", Artificial Intelligence 29, 1986, pp. 33-61.

[MCDE,80]

Mc Dermott, D. & J. Doyle, "Non-Monotonic Logic I", Artificial Intelligence, Vol. 13, Apr. 1980.

[POIN,81]

Poincaré, Henri, "Filosofía de la Ciencia", CONACYT, México, 1981. (Original en francés editado a finales del siglo pasado. Poincaré murió en 1912).

- [QUIN,78]  
Quine, W. V. & J. S. Ullian, "The Web of Belief", Raudom House, New York, 1978.
- [SHOR,76]  
Shortliffe, E.H., "Computer-based Medical Consultations: MYCIN", Elsevier, New York, 1976.
- [SPIE,75]  
Spiegel, M., "Probability and Statistics", Mc-Graw Hill, New York, 1975.
- [STRA,85]  
Strandberg, C., D. Abramovich, D. Mitchell and K. Prill, "PAGE-1 : A Troubleshooting Aid for Nonimpact Page Printing Systems", Proceedings of the Second Conference on Artificial Intelligence Applications, IEEE pRes, 1985.
- [ZADE,65]  
Zadeh, L. A., "Fuzzy Sets", Information and Control, 8:338-359 (1965).
- [ZADE,75]  
Zadeh, L. A., "Fuzzy Sets", Information and Control, Vol. 8, 1975.
- [ZADE,78]  
Zadeh, L. A., "PRUF: A Meaning Representation Language for Natural Languages", International Journal of Man-Machine Studies, Vol. 10, 1978.

#### CAPITULO V

- [BLED,77]  
Bledsoe, W. W., "Non-resolution Theorem Proving", Artificial Intelligence, Vol. 9, 1977.
- [CHAN,73]  
Chang, C. and R. Lee, "Symbolic Logic and Mechanical Theorem Proving", Academic Press, New York, 1973.
- [DAVI,60]  
Davis, M. and H. Putnam "A Computing Procedure for Quantification Theory", JACM, 7 (1960).
- [GELE,63]  
Gelernter, H., J. R. Hansen, & D. W. Loveland, "Empirical

Explorations of the Geometry Theorem Proving Machine", in Computers and Thought, E. A. Feigenbaum & J. Feldman (Eds.), Mc-Graw Hill, New York, 1963.

[KLEN,83]

Klenk, V, "Understanding Symbolic Logic", Prentice Hall, Englewood Cliffs, N. J., 1983.

[NEWE,63A]

Newell, A. & H. Simon, "GPS, a Program that Simulates Human Thought", en Computers and Thought, E.A. Feigenbaum & J. Feldman (Edit), McGraw-Hill, Nueva York, 1963.

[RICH,83]

Rich, E., "Artificial Intelligence", Mc-Graw Hill, New York, 1983.

[ROBI,65]

Robinson, J., "A Machine-oriented Logic Based on the Resolution Principle", JACM, 12(1), 1965.

[WOSL,84]

Wos, L., R. Overbeef, E. Luck and J. Boyle, "Automated Reasoning", Prentice Hall, Englewood-Cliffs, N.J., 1984.

## CAPITULO VI

[DELG,87]

Delgado Ortega, Fernando, "Fundamentos de Programación Heurística y Aplicaciones", Tesis para obtener el grado de Licenciado en Matemáticas Aplicadas y Computación, ENEP Acatlán, 1987.

[LENA,82]

Lenat, D. B., "Heuretics: The Nature of Heuristics", Artificial Intelligence, Vol 19, No. 2, Oct. 1982.

[LENA,83]

Lenat, D.B., "Eurisko : a program that learns new heuristics and domain concepts. The natural of heuristics III: program design and results", Artificial Intelligence, Marzo, 1983.

[SIMO,81]

Simon, H. A., "The Sciences of the Artificial", 2nd Ed., MIT Press, Cambridge, Mass, 1981.



- [TRAK,85]  
Trakhtenbrot, B. A., "Algoritmos y Computadoras", Edit. Limusa, México, D.F., 1985.
- CAPITULO VII
- [CHOM,57]  
Chomsky, N., "Syntactic Structures", La Haye, Mouton, 1957.
- [DUDA,78]  
Duda, Richard O.; Hart, Peter E. and Satherman, G.L., "Semantic Network Representations in Rulle Based Inference Systems", pp. 203-221 in Pattern Directed Inference Systems, D.A. Waterman and F. Hayes-Roth (Edits), Academic Press, Nueva York, 1978.
- [DUDA,80]  
Duda, Richard O.; Gaschining, John G. and Hart Peter E., "Model Design in the PROSPECTOR Consultant System for Mineral Exploration", pp 153-167 in Expert System in the Microelectronics Age, D. Michie (Edit), Edinburgh University Press, Edinburgh, 1980.
- [FILL,68]  
Fillmore, C., "The Case for Case ", in Banch and Harms (Edit), Universals in Linguistic Theory, Chicago, Halt, Rinehart and Winston, 1968.
- [GREE,69]  
Green, C., "The Application of the Orem-Proving to Question-Answering Systems", these Ph D., Dept. of Electrical Engineering Standford University, 1969.
- [LENA,83]  
Lenat, D.B., "EURISKO: A program that learns new heuristics and domain concepts", Artificial Intelligence 21, 1983.
- [POST,43]  
Post E., "Formal Reductions of the General Combinatorial Decicion Problem", American Journal of Mathematics 65, pp. 197-268, 1943.
- [QUIL,68]  
Quilliam, M.R., "Semantic Memory" in Semantic Information Processing, M. Minsky (Edit), Cambridge, Massachuttes, MIT Press, pp. 22-270, 1968.

- [RAPH, 68]  
Raphael, B., "SIR: A Computer Program for Semantic Information Retrieval", in Semantic Information processing, Cambridge, Mass., MIT Press, 1968.
- [SIMM, 73]  
Simmons, R.F., "Semantic Network: their computation and use for understanding english sentences" in Computer Models of Thought and Language, Shank et Colby (Edits), San Francisco, Freeman, 1973.
- [WATE, 78]  
Waterman, D. A. & F. Hayes-Roth, "Pattern-Directed Inference Systems, Academic Press, New York, 1978.
- [WINS, 75]  
Winston, P.H., "Learning Structural Descriptions from Examples in Psychology of Computer Vision", P.H. Winston (Edit), Mc Graw-Hill, Nueva York, 1975.
- [WOOD, 75]  
Woods, W., "What's in a Link", Foundations for Semantic Network Representation and Understanding, Bobrow et Collins (Edits), Nueva York, Academic Press, pp 35-82, 1975.

#### CAPITULO VIII

- [BORL, 86]  
Borland International, Inc., "Turbo-Prolog Owner's Handbook", Borland Inc, 1986.
- [BRAT, 86]  
Bratko, I., "Prolog Programming For Artificial Intelligence", Addison-Wesley, Reading, Massachusetts, 1986.
- [CLOC, 81]  
Clocksin, F. W. & Mellish, C. S., "Programming in Prolog", Springer-Verlag, Berlin, 1981.
- [COLM, 78]  
Colmerauer, A., "Metamorphosis Grammars, Natural Language Communications With Computers 1", 1978.
- [KOWA, 74]  
Kowalsky, R. A., "Predicate Logic as a Programming Language", Proc. IFIP 74, pp. 569-574, 1974.

- [KOWA,79]  
Kowalsky, R. A., "Algorithm = Logic + Control", CACM, 1979.
- [MCCA,60]  
McCarthy, John, "Recursive Functions of Symbolic Expressions and their Computation by Machine", Communications ACM 11, 1960. pp 184-195.
- [ROUS,75]  
Roussel, P., "Prolog - Manual de Reference et D'Utilisation : Groupe de IA", Ver Luminy, Univ. D'Aix, Marsella, Francia, 1975.
- [SKIL,76]  
Skilossy, L., "Let's Talk Lisp", Prentice-Hall, Englewood Cliffs, Nueva Jersey, 1976.
- [WILE,84]  
Wilensky, R., "Lispcraft", Norton & Co. Inc, Nueva York, 1984.
- [WINS,81]  
Winston, P.H. & Horn, B.H., "Lisp", Addison-Wesley, Reading, Massachuttes, 1981.